



# 4th Board of Studies Meeting 2021

*at the*

**Department of Computer and  
Software Technology**



Web: [www.uswat.edu.pk](http://www.uswat.edu.pk)

Contact: 0946-613082

Address: PTCL Campus, Sector-C Kanju Township, Kanju Swat



Office of the Registrar  
(Academics Section)

**University of Swat**

Main Administration Block, Charbagh, Swat  
Cell# 0316-4115599  
[khurshid@uswat.edu.pk](mailto:khurshid@uswat.edu.pk)

No.UoS/Acad/2021-95

Dated: 01/03/2021

## NOTIFICATION

In supersession of Notification No. UoS/Acad/2020-424 dated; 06/10/2020 followed by UoS/Acad/2020-500 dated; 18/11/2020, the Vice Chancellor under provision of Section-3 (2) of the Constitution, Functions and Powers of Authorities of the University of Peshawar Statutes, 2016 has been pleased to constitute Board of Studies for Department of Computer and Software Technology, University of Swat comprising of the following:

- |   |          |
|---|----------|
| 1. <b>Dr. Sanaullah,</b><br>Chairman, Department of Computer and Software Technology, University of Swat.   | Convener |
| 2. <b>Dr. Kifayat Ullah,</b><br>Department of Computer and Software Technology, University of Swat          | Member   |
| 3. <b>Dr. Muzammil Khan,</b><br>Department of Computer and Software Technology, University of Swat          | Member   |
| 4. <b>Dr. Amjad Ali,</b><br>Department of Computer and Software Technology, University of Swat              | Member   |
| 5. <b>Mr. Umar Ali,</b><br>Department of Computer and Software Technology, University of Swat               | Member   |
| 6. <b>Prof. Dr. Jamil Ahmad,</b><br>Chairman, Department of Computer Science and IT, University of Malakand | Member   |
| 7. <b>Dr. Arif ur Rahman,</b><br>Department of Computer Science, Bahria University, Islamabad               | Member   |
| 8. <b>Dr. Sehat Ullah,</b><br>Department of Computer Science, University of Malakand                        | Expert   |
| 9. <b>Prof. Dr. Shah Khusro,</b><br>Chairman, Department of Computer Science, University of Peshawar        | Expert   |
| 10. <b>Dr. Pervez Khan,</b><br>Department of Computer Science, University of Malakand                       |          |

### Terms of Reference:

- The term of office of members of the Board of Studies, other than ex-officio members, shall be three years.
- The quorum for meetings of the Board of Studies shall be one-half of members, a fraction being counted as one.
- Functions of the Board of Studies shall be:
  - To advise the Authorities on all academic matters connected with instruction, research and examination in the subject or subjects concerned.
  - To propose curricula and syllabi for all degrees, diplomas and certificates courses in the subject or subjects concerned
  - To suggest a panel of names of paper-Setters and Examiners in the subjects or subjects concerned; and
  - To perform such other functions as may be prescribed by Regulations.

(Khurshid Alam)  
Deputy Registrar (Academics)

### Copy for Information to the:

- PS to Vice Chancellor
- PS to Registrar
- Members Concerned through Convener
- File

(Khurshid Alam)  
Deputy Registrar (Academics)



## Department of Computer and Software Technology

### **Vision**

To impart quality education at all levels to contribute to the national pool of computer scientists who can meet the demands of the industry and the academia, and to excel in research by developing linkages with national and international organizations.

### **Mission**

To train students towards critical thinking and enable them to contribute to IT industries to national and international levels, and to engage students in workshops, seminars, and other research activities to prepare them for job market.



## Annexure -A

### **BS Computer Science Scheme of Study**

#### **Aims**

The program should also provide an excellent foundation for further formal learning and training. The Computer Science curriculum is expected to provide environments to put into practice, the principles and techniques learnt during the course of implementation of academic program. As a result, the graduate should be able to assume responsible positions in business, government, and education at the research, development, and planning levels.

#### **Objectives**

- The program should provide a broad understanding of the field via introducing concepts, theory, and techniques.
- Intensive education/training in focused areas of Computer Science is desirable.
- The program may encourage students to develop and use abstract models in addition to apply respective technology in practical situations.
- Computer Science graduates require special communication skills both orally and in writing. They must be able to produce well-organized reports, which clearly delineate objectives, methods of solution, results, and conclusions for a complex task.
- Analyze the local and global impact of computing on individuals, organizations, and society.
- Recognize the need for and an ability to engage in continuing professional development.
- Use the current techniques, skills, and tools necessary for computing practice.
- Use and apply the latest technical concepts and practices in the core information technologies.
- Identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems.

Understand the best practices and standards and their application. Assist in the creation of an effective project plan.

#### **Programme Model:**

The programme is designed to achieve systematically the objectives set out above. It has been structured to suit the needs of the students, the demands of the market and trends. During the first two years of the programme the students will be given core understanding of the programme. The students will be exposed to the discipline in a systematic, gradual and definite way. Students will also be trained in the skills and techniques which are rooted in the basic sciences like mathematics and physics. These areas will be taken care of in the supporting courses which have been allocated reasonably sufficient space. Students' personal traits and personality polishing will be cared for by the general education courses including communication and writing skills. A host of slots for elective courses have also been proposed to give to the students an opportunity to move towards their areas of interest. During the senior years the students will be given exposure to the more specialized aspects



of the discipline. They will also be given training in at least one application domain which will help institutions to prepare human resource well suited to the needs of different segments of the job market. In order to inculcate among them a scientific attitude they will go through a substantial lab work, which will prepare them for the industry and for further research oriented studies. The final year project will mark the crystallization and culmination of the students' four-year learning experience.

The programme structure is given as under:

Program Duration	8 semesters spread over 4 years
No of Semesters per year	2 semesters (Fall semester & Spring semester)
Minimum Credit Hours required	130

### Eligibility:

The eligibility criterion for admission to BS Computer Science is given as under:

FSc (Pre-Medical/Pre-Engineering/Intermediate in Computer Science (ICS)) or FA with mathematics or equivalent with 50% marks.

### Program Structure:

As per HEC guidelines of BS Computer Science 2017, the following table gives the credit hour distribution of the core and elective courses:

#	Category	Credit Hours		
1	Computing Courses	51		
	Core Courses		39	
	Mathematics and Science Foundation		12	
2	Computer Science Courses		51	
	Computer Science Core Courses	24		
	Computer Science Supporting Courses	09		
	Computer Science Electives Courses	18		
3	General Education Courses	19	19	
4	University Elective Courses	12	12	
5	Non-Credit Courses	00	00	
	Total Credit Hours		133	
Computing Core Courses				
S.#	Course Code	Courses Title	Credit Hours	
01	CS-102	Programming Fundamental	4 (3+1)	
02	CS -151	Object Oriented Programming	4 (3+1)	
03	CS -201	Data Structure and Algorithms	4 (3+1)	
04	CS -152	Discrete Structures	3 (3+0)	



06	CS -301	Operating Systems	4 (3+1)	
07	CS -251	Database Systems	4 (3+1)	
08	SE -302	Software Engineering	3 (3+0)	
09	CS -352	Computer Networks	4 (3+1)	
10	CS -452	Human Computer Interaction	3 (3+0)	
11	CS -404	Final year Project-I	3 (0+3)	
12	CS -404	Final Year Project-II	3(0+3)	
<b>Computer Science Supporting Courses</b>				
13	MATH-201	Differential Equations	3 (3+0)	
14	MATH-	Multi-variate Calculus	3 (3+0)	
15	CS-304	Graph Theory	3 (3+0)	
16	CS-305	Theory of Programming Languages	3 (3+0)	
17	MATH-302	Numerical Computing	3 (3+0)	
<b>General Education Courses</b>				
18	ENG-101	English Composition and Comprehension	3 (3+0)	
19	ENG-151	Communication and Presentation Skills	3 (3+0)	
20	ENG-351	Technical and Business Writing	3 (3+0)	
21	ISL-101	Islamic Studies	2 (2+0)	
22	PS-151	Pakistan Studies	2 (2+0)	
23	MGT-201	Professional Practices	3 (3+0)	
24	CS-101	Introduction to Information and Communication Technologies	3 (2+1)	



Computer Science Core courses				
#	Course Code	Course Title	Credit Hours	
25	CS-303	Compiler Construction	3 (3+0)	
26	CS-203	Computer Organization & Assembly Language	4 (3+1)	
27	CS-202	Digital Logic Design	4 (3+1)	
28	CS-252	Design & Analysis of Algorithms	3 (3+0)	
29	CS-403	Parallel & Distributed Computing	3 (3+0)	
30	CS-351	Artificial Intelligence	4 (3+1)	
31	CS-253	Theory of Automata	3 (3+0)	
Mathematics and Science Foundation Courses				
32	MATH-101	Calculus and Analytical Geometry	3 (3+0)	
33	STAT-151	Probability and Statistics	3 (3+0)	
34	MTH-251	Linear Algebra	3 (3+0)	
35	PHY-101	Applied Physics	3 (3+0)	
Computer Science Elective Courses				
#	Course Code	Course Title	Credit hours	
36	CS-353	Modern Programming Languages	3 (2+1)	
37	IT-451	Web Technologies	3 (2+1)	
38	CS-401	Visual Programming	3(2+1)	
39	CS-402	Network Strategies	3 (3+0)	
40	SE-352	Advance Software Engineering	3 (3+0)	
41	CS-453	Web Engineering	3 (2+1)	
42	IT-454	Information System Audit	3 (3+0)	
43	IT-455	Cloud Computing	3(3+0)	
44	CS-354	Introduction to Soft Computing	3 (3+0)	
45	CS-454	Real Time Systems	3 (3+0)	
46	CS-355	Data Warehousing	3 (3+0)	
47	CS-455	Data Mining	3 (3+0)	
49	CS-456	Data Encryption and Security	3 (3-0)	
50	CS-356	Advance Database Management Systems	3 (2+1)	
51	CS-357	Introduction to Bioinformatics	3 (3+0)	
52	SE-353	System Analysis and Design	3 (3+0)	
53	SE-354	Event Driven Programming	3 (2+1)	
54	IT-456	Social Networks	3 (3+0)	



55	CS-358	Wireless Sensor Networks	3 (3+0)	
56	SE-355	Functional Programming	3 (2+1)	
57	CS-359	Mobile Computing	3 (3+0)	
58	IT-457	Cyber Law	3 (3+0)	
59	CS-457	Computer Graphics	3 (2+1)	
60	CS-458	Big Data Analytics	3 (3+0)	
61	IT-458	E-Commerce	3 (3+0)	
62	CS-360	Game Application Development	3 (2+1)	
63	SE-356	Global Software Development	3 (3+0)	
64	IT-459	Management Information System	3 (3+0)	
65	CS-361	Mobile Application Development	3 (2+1)	
66	IT-460	Multimedia Communication	3 (3+0)	
67	CS-459	Natural Language Processing	3 (3+0)	
68	CS-460	Semantic Web	3 (3+0)	
69	CS-362	System Programming	3 (2+1)	
70	SE-357	Topics in Software Engineering	3 (3+0)	

## University Elective Courses

71	MGT-151	Principles of Accounting	3(3+0)	
72	MGT-251	Organization Behavior	3(3+0)	
73	PSY-401	Principles Psychology	3(3+0)	
74	MGT-451	Entrepreneurship	3(3+0)	

## Non-Credit Courses

75	MATH-103	Mathematics-I	3 (3+0)	
76	MATH-154	Mathematics-II	3 (3+0)	

A Student majoring in **Bachelor of Science in Computer Science (BS-CS)** must complete minimum of **130 Credit Hours** courses. The courses semester wise structure is as follows:

Semester I			
Pre-requisite	Course Code	Title	Credit Hours
-	CS-101	Introduction to Information and Communication Technologies	3 (2+1)



-	CS-102	Programming Fundamentals	4 (3+1)
-	MATH-101	Calculus and Analytical Geometry	3 (3+0)
-	ISL-151	Islamic Studies	2 (2+0)
-	ENG-101	English Composition and Comprehension	3 (3+0)
-	PHY-101	Applied Physics	3 (3+0)
	MATH-103	Mathematics-I	3 (3+0)*
<b>Total Semester Credit Hours</b>			<b>18 (16+2)</b>
<i>* Non-Credit Course for Pre-Medical Students</i>			
<b>Semester II</b>			
Pre-requisite	Course Code	Title	Credit Hours
CS-102	CS-151	Object Oriented Programming	4 (3+1)
-	CS-152	Discrete Structures	3 (3+0)
ENG-101	ENG-151	Communication and Presentation Skills	3 (3+0)
-	STAT-151	Probability and Statistics	3 (3+0)
-	PS-151	Pakistan Studies	2 (2+0)
-	MGT-151	Principles of Accounting	3 (3+0)
	MATH-154	Mathematics-II	3 (3+0) *
<b>Total Semester Credit Hours</b>			<b>18 (17+1)</b>
<i>* Non-Credit Course for Pre-Medical Students</i>			
<b>Semester III</b>			
Pre-requisite	Course Code	Title	Credit Hours
CS-151	CS-201	Data Structure and Algorithms	4 (3+1)
-	CS-203	Computer Organization and Assembly Language	4 (3+1)
PHY-101	CS-202	Digital Logic Design	4 (3+1)
-	MGT-201	Professional Practices	3 (3+0)
-	MTH-201	Differential Equations	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>18(15+3)</b>
<b>Semester IV</b>			
Pre-requisite	Course Code	Title	Credit Hours
CS-201	CS-252	Design and Analysis of Algorithms	3(2+1)
	CS-253	Theory of Automata	3(3+0)
	CS-251	Database Systems	4(3+1)
	MATH-251	Linear Algebra	3 (3+0)
	MGT-251	Organization Behavior	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>16(14+2)</b>



<b>Semester V</b>			
Pre-requisite	Course Code	Title	Credit Hours
CS-253	CS-303	Compiler Construction	3 (3+0)
	CS-304	Graph Theory	3 (3+0)
CS-201	CS-301	Operating Systems	4 (3+1)
	CS-302	Software Engineering	3 (3+0)
	MATH-302	Numerical Computing	3 (2+1)
<b>Total Semester Credit Hours</b>			<b>16 (14+2)</b>
<b>Semester VI</b>			
Pre-requisite	Course Code	Title	Credit Hours
CS-202	CS-351	Artificial Intelligence	4 (3+1)
	CS-352	Computer Networks	4 (3+1)
		CS Elective-1	3 (2+1)
		CS Elective-2	3 (3+0)
ENG-151	ENG-351	Technical and Business Writing	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>17(14+3)</b>
<b>Semester VII</b>			
Pre-requisite	Course Code	Title	Credit Hours
		CS Elective-3	3 (2+1)
		CS Elective-4	3 (3+0)
	PSY-401	Principles of Psychology	3 (3+0)
CS-301	CS-403	Parallel and Distributed Computing	3 (3+0)
	CS-404	Final Year Project – I	3 (0+3)
<b>Total Semester Credit Hours</b>			<b>15 (11+4)</b>
<b>Semester VIII</b>			
Pre-requisite	Course Code	Title	Credit Hours
		CS Elective-5	3 (3+0)
	MGT-451	Entrepreneurship	3 (3+0)
		CS Elective-6	3 (3+0)
	CS-452	Human Computer Interaction	3 (3+0)
	CS-404	Final Year Project-II	3 (0+3)
<b>Total Semester Credit Hours</b>			<b>15(12+3)</b>

*\*The selection of the CS elective courses from the list will be based on the expertise of faculty member and demand of the market*



## Annexure -B

### **BS Software Engineering Scheme of Study**

#### **Programme Objective:**

The objective of the programme is to prepare students for professional careers and for graduate studies in software engineering. With a balance between computing theory and practical application of software engineering concepts including software project management, methodologies, tools and technologies in the modern software development environments. Graduates of such a programme will be able to function as proficient software developers and effective team members. They will have grounding in communication, mathematics and science, and the cultural, historical, and social issues that influence and effect or relate to the development of high quality software systems. They will have knowledge of and experience with software product engineering and engineering management and an understanding of professional issues and practices. Graduates will be able to understand and assess their own software engineering capabilities and performance.

The curriculum is designed to ensure breadth across allied disciplines and supporting subjects; and depth in most areas of the software engineering body of knowledge. Various components have been included in the curriculum to ensure that the graduates will:

1. Understand and be able to apply mathematics, physical science, computer science and related disciplines.
2. Understand and be able to apply the principles of software engineering practice and process, subject to realistic constraints.
3. Be able to model, analyze, document and track system requirements, both functional and non-functional.
4. Be able to design, implement, deploy and maintain software systems.
5. Be able to verify and validate the software systems.
6. Have an awareness of current industry standards and practices.
7. Be able to work in one or more application domains.
8. Understand and apply the principles of the team process.
9. Be able to understand and apply software project management skills, measurement, estimation, costing, planning, deployment and tracking of resources.
10. Have strong communication and interpersonal skills.
11. Be capable of independent learning.
12. Understand professional responsibility and application of ethical principles.
13. Have knowledge of economics, humanities and social sciences.

#### **Programme Model:**

The programme is designed to achieve systematically the objectives set out above. It has been structured to suit the needs of the students, the demands of the market and trends. During the first two years of the programme the students will be given core understanding of the programme may be similar to other focusing areas in computing—computer science, information technology. The



students will be exposed to the discipline in a systematic, gradual and definite way. Students will also be trained in the skills and techniques which are rooted in the basic sciences like mathematics and physics. These areas will be taken care of in the supporting courses which have been allocated reasonably sufficient space. Students' personal traits and personality polishing will be cared for by the general education courses including communication and writing skills. A host of slots for elective courses have also been proposed to give to the students an opportunity to move towards their areas of interest. During the senior years the students will be given exposure to the more specialized aspects of the discipline. They will also be given training in at least one application domain which will help institutions to prepare human resource well suited to the needs of different segments of the job market. In order to inculcate among them a scientific attitude they will go through a substantial lab work, which will prepare them for the industry and for further research oriented studies. The final year project will mark the crystallization and culmination of the students' four-year learning experience.

The programme structure is given as under:

Program Duration	8 semesters spread over 4 years
No of Semesters per year	2 semesters (Fall semester & Spring semester)
Minimum Credit Hours required	130

### Eligibility:

The eligibility criterion for admission to BS Software Engineering is given as under:

FSc (Pre-Medical/Pre-Engineering Intermediate in Computer Science (ICS)) or FA with mathematics or equivalent.

### Program Structure:

As per HEC guidelines of BS Software Engineering 2017, the following table gives the credit hour distribution of the core and elective courses:

#	Category			Credit Hours		
1	Computing Courses			51		
	Core Courses					39
	Mathematics and Science Foundation					12
2	Software Engineering Courses				51	
	Software Engineering Core Courses			24		
	Software Engineering Supporting Courses			09		
	Software Engineering Electives Courses			18		
3	General Education Courses			19	19	
4	University Elective Courses			12	12	
5	Non-Credit Courses			00	00	
	Total Credit Hours				133	
Computing Core Courses						
S.No	Pre-Requisite	Course Codes	Courses Title		Credit Hours	



01	-	CS-102	Programming Fundamental	4 (3+1)
02	CS -102	CS-151	Object Oriented Programming	4 (3+1)
03	CS -151	CS-201	Data Structure and Algorithms	4 (3+1)
04	-	CS-152	Discrete Structures	3 (3+0)
06	CS -201	CS-301	Operating Systems	4 (3+1)
07	CS -201	CS-251	Database Systems	4 (3+1)
08	-	CS-302	Software Engineering	3 (3+0)
09	-	CS-352	Computer Networks	4 (3+1)
10	-	CS-363	Information Security	3 (3+0)
11	-	CS-404	Final year Project-I	3 (0+3)
12		CS-404	Final year Project-II	3 (0+3)
Software Engineering Supporting Courses				
13	-	SE-451	Business Process Engineering	3 (3+0)
14	SE-304	SE-358	Formal Methods in Software Engineering	3 (3+0)
15	-	SE-303	Operations Research	3 (3+0)
16	-	SE-304	Simulation and Modeling	3 (3+0)
Total Credit Hours			12 (12+0)	
General Education Courses				
17	-	ENG-101	English Composition and Comprehension	3 (3+0)
18	ENG-101	ENG-151	Communication and Presentation Skills	3 (3+0)
19	ENG-151	ENG-351	Technical and Business Writing	3 (3+0)
20	-	ISL-101	Islamic Studies	2 (2+0)
21	-	PS-151	Pakistan Studies	2 (2+0)
22	-	MGT-201	Professional Practices	3 (3+0)
23	-	CS-101	Introduction to Information and Communication Technologies	3 (2+1)
Software Engineering Core courses				
#	Pre-Req	Course Code	Course Title	Credit Hours
24		SE-303	Software Construction and Development	3 (2+1)
25		CS-452	Human Computer Interaction	3 (3+0)
26		SE-201	Software Requirements Engineering	3 (3+0)
27		SE-251	Software Design and Architecture	3 (2+1)
28		SE-359	Software Re-Engineering	3 (3+0)
29		SE-401	Software Project Management	3 (3+0)
30		CS-453	Web Engineering	3 (2+1)
31		SE-358	Software Quality Engineering	3 (3+0)



Mathematics and Science Foundation Courses				
32	-	MATH-101	Calculus and Analytical Geometry	3 (3+0)
33	-	STAT-151	Probability and Statistics	3 (3+0)
34	-	MATH-251	Linear Algebra	3 (3+0)
35	-	PHY-101	Applied Physics	3 (3+0)
Software Engineering Elective Courses				
SNo.	Prereq	Course Code	Course Title	Credit hours
36	-	CS-353	Modern Programming Languages	3 (2+1)
37	-	IT-451	Web Technologies	3 (2+1)
38	-	CS-401	Visual Programming	3(2+1)
39	-	CS-253	Theory of Automata	3 (3+0)
40	-	CS-402	Network Strategies	3 (3+0)
41	-	SE-352	Advance Software Engineering	3 (3+0)
42	-	IT-454	Information System Audit	3 (3+0)
43	-	IT-455	Cloud Computing	3(3+0)
44	-	CS-354	Introduction to Soft Computing	3 (3+0)
45	-	CS-454	Real Time Systems	3 (3+0)
46	-	CS-355	Data Warehousing	3 (3+0)
47	-	CS-455	Data Mining	3 (3+0)
48	-	CS-456	Data Encryption and Security	3 (3-0)
49	-	CS-356	Advance Database Management Systems	3 (2+1)
50	-	CS-357	Introduction to Bioinformatics	3 (3+0)
51	-	SE-353	System Analysis and Design	3 (3+0)
52	-	SE-354	Event Driven Programming	3 (2+1)
53	-	IT-456	Social Networks	3 (3+0)
54	-	CS-358	Wireless Sensor Networks	3 (3+0)
55	-	SE-355	Functional Programming	3 (2+1)
56	-	CS-359	Mobile Computing	3 (3+0)
57	-	IT-457	Cyber Law	3 (3+0)
58	-	CS-457	Computer Graphics	3 (2+1)
59	-	CS-458	Big Data Analytics	3 (3+0)
60	-	IT-458	E-Commerce	3 (3+0)
61	-	CS-360	Game Application Development	3 (2+1)
62	-	SE-356	Global Software Development	3 (3+0)
63	-	IT-459	Management Information System	3 (3+0)



64	-	CS-361	Mobile Application Development	3 (2+1)
65	-	IT-460	Multimedia Communication	3 (3+0)
66	-	CS-459	Natural Language Processing	3 (3+0)
67	-	CS-460	Semantic Web	3 (3+0)
68	-	CS-362	System Programming	3 (2+1)
69	-	SE-357	Topics in Software Engineering	3 (3+0)
70	-	SE-463	Software Metrics	3 (3+0)

## University Elective Courses

73	-	MGT-151	Principles of Accounting	3 (3+0)
74	-	MGT-251	Organization Behavior	3 (3+0)
75	-	PSY-401	Principles of Psychology	3 (3+0)
76	-	MGT-451	Entrepreneurship	3 (3+0)

## Non-Credit Courses

77		MATH-103	Mathematics-I	3 (3+0)
78		MATH-154	Mathematics-II	3 (3+0)

A Student majoring in **Bachelor of Science in Software Engineering (BSSE)** must complete minimum of **130 Credit Hours** courses. The courses semester wise structure is as follows:

Semester I			
Pre-requisite	Course Code	Title	Credit Hours
-	CS-101	Introduction to Information and Communication Technology	3 (2+1)
-	CS-102	Programming Fundamentals	4 (3+1)
-	MATH-101	Calculus and Analytical Geometry	3 (3+0)
-	ISL-101	Islamic Studies	2 (2+0)
-	ENG-101	English Composition and Comprehension	3 (3+0)
-	PHY-101	Applied Physics	3 (3+0)
	MATH-103	Mathematics-I	3 (3+0) *
Total Semester Credit Hours			18(16+2)
* Non-Credit Course for Pre-Medical Students			
Semester II			
Pre-requisite	Course Code	Title	Credit Hours
CCC-102	CS-151	Object Oriented Programming	4 (3+1)
-	SE-302	Software Engineering	3 (3+0)



ENG-101	ENG-151	Communication and Presentation Skills	3 (3+0)
-	CS-152	Discrete Structures	3 (3+0)
-	PS-151	Pakistan Studies	2 (2+0)
-	MGT-151	Principles of Accounting	3 (3+0)
MATH-103	MATH-154	Mathematics-II	3 (3+0) *
<b>Total Semester Credit Hours</b>			<b>18(17+1)</b>
<i>* Non-Credit Course for Pre-Medical Students</i>			
<b>Semester III</b>			
Pre-requisite	Course Code	Title	Credit Hours
CS-151	CS-201	Data Structure and Algorithms	4 (3+1)
SE-302	SE-201	Software Requirements Engineering	3 (3+0)
SE-304	CS -452	Human Computer Interaction	3 (3+0)
-	MATH-251	Linear Algebra	3 (3+0)
-	MGT-251	Organization Behavior	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>16 (15+1)</b>
<b>Semester IV</b>			
Pre-requisite	Course Code	Title	Credit Hours
CS-201	CS-301	Operating Systems	4 (3+1)
-	CS-251	Database Systems	4 (3+1)
SE-302	SE-251	Software Design and Architecture	3 (2+1)
-	STAT-151	Probability and Statistics	3 (3+0)
-	PSY-401	Principles of Psychology	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>17 (14+3)</b>
<b>Semester V</b>			
Pre-requisite	Course Code	Title	Credit Hours
SE-302	SE-303	Software Construction and Development	3 (2+1)
-	CS-352	Computer Networks	4 (3+1)
ENG-151	ENG-351	Technical and Business Writing	3 (3+0)
		SE Elective-1	3 (2+1)
-	SE-303	Operations Research	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>16 (13+3)</b>
<b>Semester VI</b>			
Pre-requisite	Course Code	Title	Credit Hours
SE-302	SE-358	Software Quality Engineering	3 (3+0)
-	CS-363	Information Security	3 (3+0)
-	MGT-201	Professional Practices	3 (3+0)
	CS-453	Web Engineering	3 (2+1)
-		SE Elective-2	3(2+1)
SE-302	SE-358	Formal Methods in Software Engineering	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>18 (16+2)</b>
<b>Semester VII</b>			
Pre-requisite	Course Code	Title	Credit Hours
SE-302	SE-401	Software Project Management	3 (3+0)



	SE-359	Software Re-Engineering	3 (3+0)
CS-151		SE Elective-3	3 (2+1)
-		SE Elective-4	3 (3+0)
-	CS-404	Final Year Project – I	3 (0+3)
<b>Total Semester Credit Hours</b>			<b>15 (11+4)</b>
<b>Semester VIII</b>			
Pre-requisite	Course Code	Title	Credit Hours
CS-352		SE Elective-5	3 (3+0)
		SE Elective-6	3 (3+0)
-	SE-451	Business Process Engineering	3 (3+0)
-	MGT-451	Entrepreneurship	3 (3+0)
-	CS-404	Final Year Project-II	3 (0+3)
<b>Total Semester Credit Hours</b>			<b>15 (12+3)</b>

*\*The selection of the SE elective courses from the list will be based on the expertise of faculty member and demand of the market*



## Annexure – C

### **BS Information Technology Scheme of Study**

#### **Aims and Objectives**

1. Bachelor of Information Technology program gives the students a strong knowledge base of programming, systems analysis and design, business telecommunication, and database management with concentration in a variety of areas.
2. Areas such as Web media, internet technology, security and information assurance area and few examples of the direction they can take within our program.
3. The aim of the Degree course in Information Technology is to produce graduates who have good grounding and a wide range of knowledge of Technology.
4. The information technology graduate's preparation will train them for work in the field of the companies that produce information system and computer network, and for companies, administrations, service and laboratories that use them.
5. Apply knowledge of computing and mathematics appropriate to the discipline.
6. Analyze a problem, and identify and define the computing requirements appropriate to its solution.
7. Design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs.
8. Function effectively on teams to accomplish a common goal.
9. Understand the professional, ethical, legal, security and social issues and responsibilities.
10. Communicate effectively with a range of audiences.
11. Analyze the local and global impact of computing on individuals, organizations, and society.
12. Recognize the need for and an ability to engage in continuing professional development.
13. Use the current techniques, skills, and tools necessary for computing practice.
14. Use and apply the latest technical concepts and practices in the core information technologies.
15. Identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems.
16. Integrate IT-based solutions into the user environment.
17. Understand the best practices and standards and their application.
18. Assist in the creation of an effective project plan.

#### **Program Outcomes**

1. Demonstrate ability to understand and contribute to the scientific, mathematical, and theoretical foundations on which computer science and information technologies are built.
2. Explain and apply appropriate information technologies and employ appropriate methodologies to help an individual or organization achieve its goals and objectives.
3. Use and apply current and emerging technical concepts and practices in information technologies.



4. Demonstrate independent, critical thinking and problem-solving competencies by being able to analyze, identify and define the requirements that must be satisfied to address problems or opportunities faced by organizations or individuals.
5. Anticipate the importance of research by being aware of basic research artifacts such as the structure of a research paper, brainstorming.
6. Demonstrate practical hands-on expertise in selection, installation, customizing and maintenance of the state-of-the-art computing infrastructure.
7. Demonstrate understanding of the social and ethical concerns of the practice of Information Technology.
8. Demonstrate the ability to work cooperatively in teams.
9. Demonstrate effective communication skills.

### Programme Model:

The programme is designed to achieve systematically the objectives set out above. It has been structured to suit the needs of the students, the demands of the market and trends. During the first two years of the programme the students will be given core understanding of the programme. The students will be exposed to the discipline in a systematic, gradual and definite way. Students will also be trained in the skills and techniques which are rooted in the basic sciences like mathematics and physics. These areas will be taken care of in the supporting courses which have been allocated reasonably sufficient space. Students' personal traits and personality polishing will be cared for by the general education courses including communication and writing skills. A host of slots for elective courses have also been proposed to give to the students an opportunity to move towards their areas of interest. During the senior years the students will be given exposure to the more specialized aspects of the discipline. They will also be given training in at least one application domain which will help institutions to prepare human resource well suited to the needs of different segments of the job market. In order to inculcate among them a scientific attitude they will go through a substantial lab work, which will prepare them for the industry and for further research oriented studies. The final year project will mark the crystallization and culmination of the students' four-year learning experience.

The programme structure is given as under:

Program Duration	8 semesters spread over 4 years
No of Semesters per year	2 semesters (Fall semester & Spring semester)
Minimum Credit Hours required	130

### Eligibility:

The eligibility criterion for admission to BS Information Technology is given as under:

FSc (Pre-Medical/Pre-Engineering Intermediate in Computer Science (ICS)) or FA with mathematics or equivalent.

### Program Structure:

As per HEC guidelines of BS Information Technology 2017, the following table gives the credit hour distribution of the core and elective courses:



#	Category	Credit Hours		
1	Computing Courses			51
	Core Courses		39	
	Mathematics and Science Foundation		12	
2	Information Technology Domain Courses			52
	Information Technology Core Courses		24	
	Information Technology Supporting Courses		09	
	Information Technology Electives Courses		19	
3	General Education Courses		19	19
4	University Elective Courses		12	12
5	Non-Credit Courses		00	00
	Total			134
Credit Hours				
Computing Core Courses ( CCC)				
S.N o	Course Code	Courses Title	Credit Hours	
01	CS-102	Programming Fundamental	4 (3+1)	
02	CS-151	Object Oriented Programming	4 (3+1)	
03	CS-201	Data Structure and Algorithms	4 (3+1)	
04	CS-152	Discrete Structures	3 (3+0)	
06	CS-301	Operating Systems	4 (3+1)	
07	CS-251	Database Systems	4 (3+1)	
08	CS-302	Software Engineering	3 (3+0)	
09	CS-352	Computer Networks	4 (3+1)	
10	CS-363	Information Security	3 (3+0)	
11	CS-404	Final year Project-I	3 (0+3)	
12	CS-404	Final year Project-II	3 (0+3)	
IT Supporting Courses (Any three from the list)				
12	IT-351	Enterprise System	3 (3+0)	
13	IT-304	Modeling and Simulation	3 (3+0)	
14	IT-353	Formal Methods	3 (3+0)	
15	SE-303	Operational Research	3 (3+0)	
16	SE-201	Software Requirement Engineering	3 (3+0)	
General Education Courses				
16	ENG-101	English Composition and Comprehension	3 (3+0)	
17	ENG-151	Communication and Presentation Skills	3 (3+0)	



18	ENG-351	Technical and Business Writing	3 (3+0)	
19	ISL-101	Islamic Studies/ Ethics	2 (2+0)	
20	PS-151	Pakistan Studies	2 (2+0)	
21	MGT-201	Professional Practices	3 (3+0)	
22	CS-101	Introduction to Information and Communication Technologies	3 (2+1)	
<b>IT Core courses</b>				
#	Course Code	Course Title	Credit Hours	
23	IT-461	Cyber Security	3 (3+0)	
	CS-463	Database Administration and Management	4 (3+1)	
24	IT-251	IT Project Management	3 (3+0)	
25	IT-352	IT Infrastructure	3 (3+0)	
26	CS-304	System and Network Administration	4 (3+1)	
27	IT-401	Virtual Systems and Services	4 (3+1)	
28	IT-451	Web technologies	3 (3+0)	
<b>Mathematics and Science Foundation Courses</b>				
29	MATH-101	Calculus and Analytical Geometry	3 (3+0)	
30	STAT-151	Probability and Statistics	3 (3+0)	
31	MATH-251	Linear Algebra	3 (3+0)	
32	PHY-101	Applied Physics	3 (3+0)	
<b>Information Technology Elective Courses</b>				
#	Course Code	Course Title	Credit hours	
33	CS-202	Digital Logic Design	4 (3+1)	
34	CS-353	Modern Programming Languages	3 (2+1)	
35	SE-352	Advance Software Engineering	3 (2+1)	
36	CS-402	Network Strategies	3 (3+0)	
37	CS-401	Visual Programming	3(2+1)	
38	CS-253	Theory of Automata	3 (3+0)	
39	IT-454	Information System Audit	3 (3+0)	
40	IT-455	Cloud Computing	3(3+0)	
41	CS-354	Introduction to Soft Computing	3 (3+0)	
42	CS-454	Real Time Systems	3 (3+0)	
43	CS-355	Data Warehousing	3 (3+0)	



44	CS-455	Data Mining	3 (3+0)	
45	CS-456	Data Encryption and Security	3 (3-0)	
46	CS-356	Advance Database Management Systems	3 (2+1)	
47	CS-357	Introduction to Bioinformatics	3 (3+0)	
48	SE-353	System Analysis and Design	3 (3+0)	
49	SE-354	Event Driven Programming	3 (2+1)	
50	IT-456	Social Networks	3 (3+0)	
51	CS-358	Wireless Sensor Networks	3 (3+0)	
52	SE-355	Functional Programming	3 (2+1)	
53	CS-359	Mobile Computing	3 (3+0)	
54	IT-457	Cyber Law	3 (3+0)	
55	CS-457	Computer Graphics	3 (2+1)	
56	CS-458	Big Data Analytics	3 (3+0)	
57	IT-458	E-Commerce	3 (3+0)	
58	CS-360	Game Application Development	3 (2+1)	
59	SE-356	Global Software Development	3 (3+0)	
60	IT-459	Management Information System	3 (3+0)	
61	CS-361	Mobile Application Development	3 (2+1)	
62	IT-460	Multimedia Communication	3 (3+0)	
63	CS-459	Natural Language Processing	3 (3+0)	
64	CS-460	Semantic Web	3 (3+0)	
65	CS-362	System Programming	3 (2+1)	
66	SE-357	Topics in Software Engineering	3 (3+0)	

## University Elective Courses

67	MGT-151	Principles of Accounting	3(3+0)	
68	MGT-251	Organization Behavior	3(3+0)	
69	PSY-401	Principles Psychology	3(3+0)	
70	MGT-451	Entrepreneurship	3(3+0)	

## Non-Credit Courses (NCC)

71	MTH-103	Mathematics-I	3 (3+0)	1
72	MTH-154	Mathematics-II	3 (3+0)	2
<b>Total Credit Hours</b>			<b>0 (0+0)</b>	

A Student majoring in **Bachelor of Science in Information Technology (BS-IT)** must complete minimum of **130 Credit Hours** courses. The courses semester wise structure is as follows:



Semester I			
Pre-requisite	Course Code	Title	Credit Hours
-	CS-101	Introduction to Information and Communication Technology	3 (2+1)
-	CS-102	Programming Fundamentals	4 (3+1)
-	ENG-101	English Composition and Comprehension	3 (3+0)
-	MTH-101	Calculus and Analytical Geometry	3 (3+0)
-	PHY-101	Applied Physics	3 (3+0)
-	ISL-101	Islamic Studies	2 (2+0)
	MTH-103	Mathematics-I	3 (3+0) *
<b>Total Semester Credit Hours</b>			<b>18(16+2)</b>
<i>* Non-Credit Course for Pre-Medical Students</i>			
Semester II			
Pre-requisite	Course Code	Title	Credit Hours
CS-102	CS-151	Object Oriented Programming	4 (3+1)
ENG-101	ENG-151	Communication and Presentation Skills	3 (3+0)
-	STAT-151	Probability and Statistics	3 (3+0)
-	MGT-151	Principles of Accounting	3 (3+0)
-	MGT-251	Organizational Behavior	3 (3+0)
-	PS-151	Pak-Studies	2 (2+0)
-	MTH-154	Mathematics-II	3 (3+0) *
<b>Total Semester Credit Hours</b>			<b>16 (15+1)</b>
<i>* Non-Credit Course for Pre-Medical Students</i>			
Semester III			
Pre-requisite	Course Code	Title	Credit Hours
CS-151	CS-201	Data Structure and Algorithms	4 (3+1)
-	CS-152	Discrete Structures	3 (3+0)
-	-	IT Elective-1	4 (3+1)
-	MATH-251	Linear Algebra	3 (3+0)
-	PSY-401	Principal of Psychology	3 (3+0)
<b>Total -Semester Credit Hours</b>			<b>17(16+2)</b>
Semester IV			
Pre-requisite	Course Code	Title	Credit Hours
CS-201	CS-301	Operating Systems	4(3+1)
-	CS-363	Information Security	3(3+0)
-	CS-352	Computer Networks	4(3+1)



-	IT-251	IT Project Management	3 (3+0)
-	SE-303	Operational Research	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>19(17+2)</b>
<b>Semester V</b>			
Pre-requisite	Course Code	Title	Credit Hours
-	CS-251	Database Systems	4 (3+1)
-	CS-302	Software Engineering	3 (3+0)
-	-	IT Elective-2	3 (2+1)
-	CS-304	System and Network Administration	4 (3+1)
-	MGT-451	Entrepreneurship	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>17(14+3)</b>
<b>Semester VI</b>			
Pre-requisite	Course Code	Title	Credit Hours
-	IT-451	Web Technologies	3 (2+1)
CS-302	SE-201	Software Requirements Engineering	3 (3+0)
-	-	IT Elective-3	3 (3+0)
-	IT-352	IT Infrastructure	3 (3+0)
ENG-151	ENG-351	Technical and Business Writing	3 (3+0)
-	IT-351	Enterprise System	3 (3+0)
<b>Total Semester Credit Hours</b>			<b>18 (17+1)</b>
<b>Semester VII</b>			
Pre-requisite	Course Code	Title	Credit Hours
-	IT-401	Virtual Systems and Services	4 (3+1)
		IT Elective-4	3 (3+0)
		IT Elective-5	3 (2+1)
-	MGT-201	Professional Practices	3 (3+0)
-	CS-404	Final Year Project – I	3 (0+3)
<b>Total Semester Credit Hours</b>			<b>16 (11+5)</b>
<b>Semester VIII</b>			
Pre-requisite	Course Code	Title	Credit Hours
-	IT-461	Cyber Security	3 (3+0)
-		IT Elective-6	3 (3+0)
CS-251	CS-463	Database Administration and Management	4 (3+1)
	CS-404	Final Year Project-II	3 (0+3)
<b>Total Semester Credit Hours</b>			<b>13(9+4)</b>

*\*The selection of the IT elective courses from the list will be based on the expertise of faculty member and demand of the market*



## **Scheme of Studies MS Computer Science**



**Department of Computer and Software  
Technology, University of Swat**



## 1. Introduction

The Department of Computer and Software Technology currently offers four programs, i.e., BS in Computer Science, BS in Software Engineering, BS in Information Technology, and Master in Computer Science (16 years). The department has currently three full-time PhD faculty members, and has the capacity to initiate MS program in Computer Science.

## 2. Scope/Vision

The basic intention of the MS program in Computer Science is to develop the student's critical professional thinking and intuition. The curriculum must be structured to provide a balanced mixture of learning experiences to make the graduate capable of sound professional decisions.

## 3. Mission

Our aim is to develop a high-profile research environment by establishing dynamic research labs, and maintaining national and international collaborations.

## 4. Objectives

The main objectives of MS program are to provide the students a high-level of understanding and knowledge in the field of Computer Science. The curriculum will enable the students to attain advance skills capable of solving future challenges. The objectives are highlighted as under.

- The program should provide a broad understanding of the field via introducing concepts, theory, and techniques.
- Intensive education/training in focused areas of Computer Science is desirable.
- The program may encourage students to develop and use abstract models in addition to apply respective technology in practical situations.
- Computer Science graduates require special communication skills both orally and in writing. They must be able to produce well-organized reports, which clearly delineate objectives, methods of solution, results, and conclusions for a complex task.
- Use the current techniques, skills, and tools necessary for computing practice.
- Use and apply the latest technical concepts and practices in the core information technologies.
- Identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems.

## 5. Outcomes



The graduates will assume numerous leadership responsibilities in universities, industries and R & D sectors.

## 6. Minimum eligibility criteria for admission in MS program

1. MSc in Computer Science with 16 years of education or equivalent, minimum of 72 credit hours with 2.5 CGPA (semester system) or 45% (annual system).  
or  
BS CS/SE/IT (4 years) with 16 years of education or equivalent, minimum of 130 credit hours with 2.5 CGPA (semester system) or 66% (Term system).
2. Test requirement for Entry to MS program for students is as may be prescribed by Higher Education of Commission of Pakistan from time to time.
3. The following core courses are recommended to be completed before entering the MS program.

1. Analysis of Algorithms
2. Computer Networks
3. Computer Programming
4. Data Structures
5. Operating Systems
6. Theory of Automata or Compiler Construction

A student selected for admission having deficiency in the above stated courses may be required to study a maximum of FOUR courses, which must be passed in the first two semesters. Deficiency courses shall be determined by the Graduate Studies Committee, before admitting the student.

## Program Structure for MS program

The complete details of MS (CS) are given below:

- 1- Minimum credit hours shall be 30 for MS Computer Science program.
- 2- The program shall comprise of minimum 4 semesters spread over 2 years with two semesters a year. The maximum duration for MS program is as per HEC policy.
- 3- The list of elective courses/specialization for MS Computer Science may be updated by the university.

## Category wise Credit Hours Distribution



Category or Area	Credit Hours
Core	6
Electives	18
Thesis	6
<b>Total Credit Hours</b>	<b>30</b>

## MS Computer Science Semester-wise Breakdown Structure

The Department of Computer & Software Technology has the option to change the sequence and number of courses offered in each semester as per its need.

Semester 1			
S.No.	Code	Subjects	Credit Hrs.
1	CS 701	Advanced Theory of Computation	3
2	CS 702	Advanced Algorithm Analysis	3
3	CS 703	Advanced Computer Architecture	3
4	CS 801	Research Methods	3
Total			12
Semester 2			
S.No.	Code	Subjects	Credit Hrs.
1	CS 704	Advanced Operating System	3
2	CS	Elective II	3
3	CS	Elective III	3
4	CS	Elective IV	3
Total			12
Semester 3			



Research Thesis	3
<b>Semester 4</b>	
Research Thesis	3



## List of Electives Courses for MS Program

The students taking an elective course with a prerequisite, the prerequisite course will be offered as a non-credit course. The students may not be allowed to participate in any elective course until the prerequisite courses are passed. However, such courses shall be graded as 'Pass' or 'Fail' and shall not be counted towards the CGPA. Deficiency courses shall be determined by the Graduate Studies Committee. The Courses are not limited to the list given below (Students will select 6 Courses of 18 credit hours.);

Code	Specialization Areas	Cr. Hrs
CS 801	Research Methods	3
CS 802	Advanced Networking	3
CS 803	Ubiquitous Computing	3
CS 804	Advanced Wireless Networks	3
CS 805	Advanced Wireless Sensor Networks	3
CS 806	Topics in Computing Networking	3
CS 807	Network Security	3
CS 808	Network Performance Evaluation	3
CS 809	Embedded Systems	3
CS 810	Advanced Computer Architecture	3
CS 811	Advanced Operating Systems	3
CS 812	Advanced Topic in Parallel Programming	3
CS 813	Fault Tolerant Systems	3
CS 814	Power Aware Computing	3
CS 815	Real Time Systems Scheduling	3



CS 816	Distributed Real Time Java Systems	3
CS 817	Advanced Topics in Real Time Systems	3
CS 818	Mixed Criticality Systems Scheduling	3
CS 819	Advanced Mixed Criticality Systems Scheduling	3
CS 820	Distributed Computing	3
CS 821	Data Mining	3
CS 822	Information Retrieval	3
CS 823	Operation Research	3
CS 824	Text Mining	3
CS 825	Advanced Human Computer Interactions	3
CS 826	Digital Libraries	3
CS 827	Machine Learning	3
CS 828	Natural Language Processing	3
CS 829	Software Quality Assurance	3
CS 830	Web Application Engineering	3
CS 831	System Modelling and Simulation	3
CS 832	System Architecture and Design	3
CS 833	Software Engineering and Formal Specifications	3
CS 834	Component Based Computing	3
CS 835	Data Science	3
CS 836	Digital Image Processing	3
CS 837	Web Mining	3



CS 838	Cloud Computing	3
CS 839	Collaborative Data Mining	3
CS 840	Advanced Theory of Computation	3
CS 841	Advanced Algorithm Analysis	3



## **Scheme of Studies PhD in Computer Science (New Policy, January 2021)**



**Department of Computer and Software Technology,  
University of Swat**



## 1. Introduction

PhD programs are research-oriented programs which enable students to understand the ethics and philosophy of research. These programs enhance the technical and research abilities of students and provide them in-depth knowledge in the selected areas.

The Department of Computer and Software Technology currently offers five programs, i.e., BS in Computer Science, BS in Software Engineering, BS in Information Technology, MS and PhD Computer Science. The department has currently four full-time PhD faculty members.

## 2. Scope/Vision

The basic intention of the PhD program in Computer Science is to develop the student's critical professional thinking and intuition. The curriculum is structured to provide a balanced mixture of learning experiences to make the graduate capable of sound professional decisions. These programs will produce high-skilled professional graduates and researchers capable of facing future challenges.

## 3. Mission

Our aim is to develop a high-profile research environment by establishing dynamic research labs, and maintaining national and international collaborations.

## 4. Objectives

The main objectives of the PhD program are to provide the students a high-level of understanding and knowledge in the field of Computer Science. The curriculum will enable the students to attain advance skills capable of solving future challenges. The objectives are highlighted as under.

- The program should provide a broad understanding of the field via introducing concepts, theory, and techniques.
- The program may encourage students to develop and use abstract models in addition to apply respective technology in practical situations.
- Analyze the local and global impact of computing on individuals, organizations, and society.
- Intensive education/training in focused areas of Computer Science is desirable.
- Recognize the need for and an ability to engage in continuing professional and research development.
- Use the current techniques, skills, and tools necessary for computing practice.
- Use and apply the latest technical concepts and practices in the core information technologies.



- Computer Science graduates require special communication skills both orally and in writing. They must be able to produce well-organized reports, which clearly delineate objectives, methods of solution, results, and conclusions for a complex task.
- Identify and analyze user needs and take them into account in the selection, creation, evaluation and administration of computer-based systems.
- Understand the best practices and standards and their application.
- Assist in the creation of an effective project plan.

## 5. Outcomes

The graduates will assume numerous leadership responsibilities in universities, industries and research & development sectors.

## 6. Minimum eligibility criteria for admission in PhD program

1. As per HEC policy (Section 3), the minimum requirement for admission to a PhD program shall be completion of the BS (or equivalent) degree, i.e., 16 Years of Education.
2. **Completion of Prior Degree.** Prior to entry into a PhD program, the student shall have been awarded his or her BS/MS/MPhil or equivalent degrees.
3. **Prior Degree in Different Discipline.** Consistent with best practices internationally (including leading global universities), it shall generally not be a requirement for admission to a PhD program that the applicant shall have completed a prior degree in the same discipline. If the applicant has a strong interest in pursuing a PhD degree in a different discipline and, in accordance with the department's policy, the admissions committee is satisfied that the applicant's prior education has sufficiently prepared him or her to undertake the course of studies of the doctoral program (or, in the opinion of the admissions committee, the preparation can be deemed satisfactory by taking a few additional courses after starting the program), the applicant shall be considered for admissions notwithstanding their prior qualification in a different discipline.
4. **Minimum GPA Requirement.** For admission in PhD programs, a minimum CGPA of 3.0 (out of 4.0 in the semester system) or First Division (in the annual system) in the BS (or equivalent) or MS degree obtained is required, whether such was degree obtained from Pakistani or foreign universities.
5. **Testing Requirement.** Applicants to PhD programs shall be required to fulfill the following testing requirements:
  - a. All applicants to PhD programs shall be required to take one of the following non-subject specific admissions tests:
    - i. The Graduate Record Examination (GRE) test administered by the Education Testing Service;
    - ii. A graduate admission test administered by the Education Testing Council; or



- iii. With the permission of the HEC, an equivalent test developed by the university, or by another university, for admissions to graduate programs.
  - b. The applicants having valid GAT-Subject will be eligible for admission in PhD Computer Science
6. **Statement of Purpose.** As part of the application for admission to PhD programs, applicants shall be required to submit a statement of purpose, which shall form an integral part of the application. The admissions committee shall use the information provided to ascertain the preparedness and interest of the candidate in pursuing doctoral studies, and whether the department has the requisite resources to train and supervise the doctoral candidate in the subspecialty he or she is interested in.



## MINIMUM REQUIREMENTS FOR AWARD OF PHD DEGREES

1. **Coursework and Residency Requirement.** The following requirements shall be fulfilled before the award of a PhD Degree:
  - a. The student shall complete coursework of at least 48 credit hours of which the majority shall be fulfilled through regular classes. There shall be a residency requirement for a period of at least two years
  - b. In case of those students who have obtained admission in the PhD program based on MS/MPhil or equivalent degree in the same discipline, shall complete coursework of at least 24 credit hours. There shall be a residency requirement for a period of at least one year.
2. **Comprehensive Examination.** Following the completion of coursework, every PhD student shall be required to pass a comprehensive examination in order to be granted candidacy as PhD researcher; provided that if the student fails to pass the comprehensive test, he or she shall be allowed one more attempt.
3. **Doctoral Dissertation.** Each PhD researcher shall be required to write a doctoral dissertation that meets the following minimum criteria:
  - a. The PhD dissertation shall be supervised by a faculty member who holds a PhD (or equivalent) degree and is an HEC approved supervisor. At the time of appointment as supervisor, the faculty member shall be a full-time faculty member of the university in which the student is enrolled.
  - b. In addition to the PhD committee members, the PhD dissertation must be evaluated by at least two external experts who may be either (i) Pakistan-based Distinguished National Professors, Meritorious Professors, or Tenure Track Professors, or (ii) PhD experts from academically advanced countries, however, that if the PhD candidate publishes his or her dissertation research in a peer reviewed journal that is classified by the HEC as category X or above, the PhD dissertation will only require evaluation by one external expert.
  - c. A plagiarism test in accordance with the HEC's Plagiarism Policy must be conducted on the dissertation before its submission to the external experts.
  - d. An open defense of the dissertation is required after positive evaluation of the dissertation by the committee members.
4. **Research Publication.** Each PhD researcher shall be required to publish at least one research paper as its first author during his or her doctoral studies in an HEC approved Y category (or above) journal for the award of PhD degree.
5. **Reporting Requirements.** The following documents shall be submitted to the HEC following the completion of studies:



- a. A duly filled completion form shall be sent to the HEC from the Office of the Controller of Examinations of the university notifying the HEC that the PhD scholar has completed all the requirements for the award of the PhD degree.
  - b. A copy of PhD Dissertation for including in PhD Country Directory and for attestation of the PhD degree by the HEC.
  - c. A duly filled Proforma for the PhD Country Directory, signed by the Principal Supervisor, Controller of Examination and the Vice Chancellor (or Rector).
6. **Degree Completion Timeline.** The PhD degree shall be awarded by universities after a minimum of three (3) years and not more than eight (8) years after the enrolment of the student; provided that for students who are unable to complete the program within eight (8) years, the university may designate a competent authority to determine whether the delay was caused by circumstances beyond the student's control, and if so, grant an extension in such exceptional circumstances; provided further that in no event shall the PhD degree be awarded more than ten (10) years after the enrolment of the student in the program. The date of notification of the award of the PhD degree subsequent to the PhD defense shall be considered to be the date of the completion of PhD studies.
7. **Award of MS/MPhil Degrees on the Basis of PhD Studies.** If a student successfully completes the full set of requirements for the award of MS degree during the course of their PhD studies, the university may award the MS degree on completion of the relevant requirements, i.e., 30 credit hours course work.

## PhD in Computer Science

### Semester-wise Breakdown Structure

Semester 1			
S.No.	Code	Subjects	Credit Hrs.
1	CS	Elective I	3
2	CS	Elective II	3
3	CS	Elective III	3
4	CS	Elective IV	3
		Total:	12
Semester 2			
1	CS	Elective I	3
2	CS	Elective II	3
3	CS	Elective III	3
4	CS	Elective IV	3
		Total:	12



Semester 3			
1	CS	Elective I	3
2	CS	Elective II	3
3	CS	Elective III	3
4	CS	Elective IV	3
		Total:	12
Semester 4			
1	CS	Elective I	3
2	CS	Elective II	3
3	CS	Elective III	3
4	CS	Elective IV	3
		Total:	12

The students have to write the thesis in the remaining semesters.

### List of Elective Courses for Ph.D.

The students taking an elective course with a prerequisite, the prerequisite course will be offered as a non-credit course. The students may not be allowed to participate in any elective course until the prerequisite courses are passed. However, such courses shall be graded as 'Pass' or 'Fail' and shall not be counted towards the CGPA. The Courses are not limited to the list given below;

Code	Specialization Areas	Cr. Hrs
CS 801	Research Methods	3
CS 802	Advanced Networking	3
CS 803	Ubiquitous Computing	3
CS 804	Wireless Networks	3
CS 805	Wireless Sensor Networks	3
CS 806	Topics in Computing Networking	3
CS 807	Network Security	3



CS 808	Network Performance Evaluation	3
CS 809	Embedded Systems	3
CS 810	Advanced Computer Architecture	3
CS 811	Advanced Operating Systems	3
CS 812	Advanced Topic in Parallel Programming	3
CS 813	Fault Tolerant Systems	3
CS 814	Power Aware Computing	3
CS 815	Real Time Systems Scheduling	3
CS 816	Distributed Real Time Java Systems	3
CS 817	Advanced Topics in Real Time Systems	3
CS 818	Mixed Criticality Systems Scheduling	3
CS 819	Advanced Mixed Criticality Systems Scheduling	3
CS 820	Distributed Computing	3
CS 821	Data Mining	3
CS 822	Information Retrieval	3
CS 823	Operation Research	3
CS 824	Text Mining	3
CS 825	Advanced Human Computer Interactions	3
CS 826	Digital Libraries	3
CS 827	Machine Learning	3
CS 828	Natural Language Processing	3
CS 829	Software Quality Assurance	3



CS 830	Web Application Engineering	3
CS 831	System Modelling and Simulation	3
CS 832	System Architecture and Design	3
CS 833	Software Engineering and Formal Specifications	3
CS 834	Component Based Computing	3
CS 835	Data Science	3
CS 836	Digital Image Processing	3
CS 837	Web Mining	3
CS 838	Cloud Computing	3
CS 839	Collaborative Data Mining	3
CS 840	Advanced Theory of Computation	3
CS 841	Advanced Algorithm Analysis	3



## BS Programs Course Contents

**Course Name:** Programming Fundamentals

**Course Code:** CS-102

**Credit Hours:** 4 (3+1)

**Prerequisites:** None

**Objectives:** To enable the students to program and learn the basic programming concepts. Analyze and explain the behavior of simple programs involving the fundamental programming constructs.

**Learning Outcome:** Explain Computer Programming concepts, like;

- Ability to design algorithmic solution to problems
- Ability to design programs with Interactive Input and Output
- Ability to design programs utilizing arithmetic expressions
- Ability to design programs utilizing repetition
- Ability to design programs utilizing decision making
- Ability to design programs utilizing arrays
- Ability to develop recursive solutions
- Ability to test and verifying programs
- Ability to develop simple search and sort algorithms

### Course Outline:

This course covers overview of Computer Programming, Principles of Structured and Modular Programming, Overview of Structured Programming Languages, Algorithms and Problem Solving, Program Development: Analyzing Problem, Designing Algorithm/Solution, Testing Designed Solution, Translating Algorithms into Programs, Fundamental Programming Constructs, Data Types. Basics of Input and Output, Selection and Decision (If, If - Else, Nested If- Else, Switch Statement and Condition Operator), Repetition (While and For Loop, Do -While Loops), Break Statement, Continue Statement, Control Structures, Functions, Arrays, Pointers, Records, Files (Input-Output), Testing & Debugging.

### Reference Materials:

1. C How to Program by Paul Deitel and Harvey Deitel, Prentice Hall; 7 edition (March 4, 2012). ISBN-10: 013299044X



2. Programming in C by Stephen G. Kochan, Addison-Wesley Professional; 4 edition (September 25, 2013). ISBN-10: 0321776410
3. Java How to Program by Paul Deitel and Harvey Deitel, Prentice Hall; 9th edition (March, 2011)
4. C++ How to Program by Paul Deitel and Harvey Deitel, Prentice Hall; 9th edition (February, 2013)

Course Name: Programming Fundamentals	
Week #	Week wise distribution
1.	overview of Computer Programming, Principles of Structured and Modular Programming, Overview of Structured Programming Languages, Algorithms and Problem Solving, Program Development: Analyzing Problem,
2.	Designing Algorithm/Solution, Testing Designed Solution, Translating
3.	Algorithms into Programs,
4.	Fundamental Programming Constructs, Data Types. Basics of Input and Output,
5.	Selection and Decision If, If -Else, Nested If- Else, Switch Statement and Condition Operator),
6.	Repetition While and For Loop,
7.	Do -While Loops),
8.	Break Statement, Continue Statement,
9.	Control Structures
10.	Functions,
11.	Arrays,
12.	Pointers,
13.	Records,
14.	Files Input-Output),
15.	Testing & Debugging.
16.	Wrap up work and presentations



**Course Name:** Object Oriented Programming

**Course Code:** CS-151

**Credit Hours:** 4 (3+1)

**Prerequisites:** Programming Fundamentals

**Objectives:** To prepare object-oriented design for small/medium scale problems and to demonstrate the differences between traditional imperative design and object oriented design.

**Learning Outcome:**

- Describe the principles of object-oriented programming
- Apply the concepts of data encapsulation, inheritance, and polymorphism to large-scale software
- Acquire the concepts of Graphical User Interfaces Professional Skill
- Design and develop object-oriented computer programs, and develop programs with Graphical User Interfaces capabilities
- Formulate problems as steps so as to be solved systematically
- Develop software with team-work in mind

**Course Outline:**

Evolution of Object Oriented Programming (OOP), Object Oriented concepts and principles, problem solving in Object Oriented paradigm, OOP design process, classes, functions/methods, objects and encapsulation; constructors and destructors, operator and function/method overloading, association, aggregation, composition, generalization, inheritance and its types, derived classes, function/method overriding, abstract and concrete classes, virtual functions, polymorphism, exception handling.

**Reference Materials:**

1. Java: How to Programme, Harvey M. Deitel and Paul J. Deitel, Prentice Hall; 8 edition (March 27, 2009). ISBN-10: 0136053068
2. C++: How to Programme, Prentice Hall; 8 edition March 25, 2011). ISBN - 10: 0132662361
3. Object Oriented Programming in C++ by Robert Lafore, Sams Publishing; 4 edition (December 29, 2001). ISBN-10: 0672323087
4. Java Programming: From the Ground Up by Ralph Bravaco and Shai Simonson, McGrawHill Higher Education New York, 2010, ISBN 978-0-07-352335-4



5. Beginning Java by Ivor Horton, John Wiley & Sons, Inc, 7th Edition, 2011, ISBN: 978-0-470-40414-0

Course Name: Object Oriented Programming	
Week #	Week wise distribution
1.	Evolution of Object Oriented Programming OOP), Object Oriented concepts and principles,
2.	problem solving in Object Oriented paradigm,
3.	OOP design process,
4.	classes, functions/methods,
5.	objects and encapsulation;
6.	constructors and destructors,
7.	operator and function/method overloading,
8.	association, aggregation, composition, generalization,
9.	inheritance and its types,
10.	derived classes,
11.	function/method overriding,
12.	abstract and concrete classes,
13.	virtual functions,
14.	polymorphism,
15.	exception handling.
16.	Wrap up work and presentations



**Course Name:** Data Structure and Algorithms

**Course Code:** CS-201

**Credit Hours:** 3 (2+1)

**Prerequisites:** Programming Fundamentals

**Course Objectives:** To Teach the students how to design algorithm and analyze the complexity

**Learning Outcome:**

- Able to understand the concepts of data structure, data type and array data structure.
- Able to analyze algorithms and determine their time complexity.
- Able to implement linked list data structure to solve various problems.
- Able to understand and apply various data structure such as stacks, queues, trees and graphs to solve various computing problems using C-programming language.
- Able to implement and know when to apply standard algorithms for searching and sorting.
- Able to effectively choose the data structure that efficiently model the information in a problem

□

**Course Outline:**

Introduction to Data Structures and Algorithms. Complexity Analysis. Arrays. Sorting Algorithms: Insertion Sort, Selection Sort, Bubble Sort, Shell Sort, Heap Sort, Quick Sort, Merge Sort, Radix Sort, Bucket Sort. Linked Lists: Singly Linked Lists, Doubly Linked Lists, Circular List. Stacks, Queues, and Priority Queue. Recursion: Function call and Recursion Implementation, Tail Recursion, Non-tail Recursion, Indirect Recursion, Nested Recursion, Backtracking. Trees: Binary Trees, Binary Heap, Binary Search. Tree Traversal, Insertion, Deletion, and Balancing a Tree. Heap. B-Tree, B+Tree, Spanning Tree, Splay  
Trees. Graphs: Representation, Traversal, Shortest Path, and Cycle Detection; Isomorphic Graphs. Graph Traversal Algorithms. Hashing. Memory Management and Garbage Collection.

**Reference Materials:**

1. Data Structures & Problem Solving Using Java by Mark Allen Weiss, Addison-Wesley, 4th Edition (October 7, 2009). ISBN -10: 0321541405 or Latest Edition)
2. Algorithms, Robert Sedgewick, Princeton University Publisher: Addison-Wesley Professional latest Edition)



3. Data Structures: Abstraction and Design Using Java by Koffman and Wolfgang, Wiley; 2nd Edition (January 26, 2010). ISBN-10: 0470128704
4. Data Structures and Algorithms in C++ by Adam Drozdek, Course Technology; 4th Edition (August 27, 2012). ISBN-10: 1133608426

Course Name: Data Structure and Algorithms	
Week #	Week wise distribution
1.	Introduction to Data Structures and Algorithms. Complexity Analysis. Arrays. Sorting Algorithms:
2.	Insertion Sort, Selection Sort, Bubble Sort, Shell Sort, Heap Sort, Quick Sort,
3.	Merge Sort, Radix Sort, Bucket Sort. Linked Lists: Singly Linked Lists, Doubly Linked Lists,
4.	Circular List. Stacks, Queues, and Priority Queue. Recursion: Function call and Recursion Implementation, Tail Recursion,
5.	Non-tail Recursion, Indirect Recursion, Nested Recursion,
6.	Backtracking. Trees: Binary Trees,
7.	Binary Heap,
8.	Binary Search.
9.	Tree Traversal, Insertion, Deletion, and Balancing a Tree.
10.	Heap. B-Tree, B+Tree,
11.	Spanning Tree,
12.	Splay Trees.
13.	Graphs: Representation, Traversal, Shortest Path, and Cycle Detection;
14.	Isomorphic Graphs. Graph Traversal Algorithms. Hashing.
15.	Memory Management and Garbage Collection.



16.	Wrap up work and presentations
-----	--------------------------------

**Course Name:** Discrete Structures

**Course Code:** CS-152

**Credit Hours:** 3

**Prerequisites:** None

**Objectives:** To learn the application of formal logic proofs and/or informal, but rigorous, logical reasoning to real problems. Comprehend discrete structures and their relevance within the context of computer science.

**Learning Outcome:**

- Use logical notation to define and reason about fundamental mathematical concepts such as sets, relations, functions, and integers.
- Evaluate elementary mathematical arguments and identify fallacious reasoning (not just fallacious conclusions).
- Synthesize induction hypotheses and simple induction proofs.
- Prove elementary properties of modular arithmetic and explain their applications in Computer Science, for example, in cryptography and hashing algorithms.

**Course Outline:**

Mathematical reasoning: introduction to logic, propositional and predicate calculus; negation disjunction and conjunction; implication and equivalence; truth tables; predicates; quantifiers; natural deduction; rules of Inference; methods of proofs; use in program proving; resolution principle; Set theory: Paradoxes in set theory; inductive definition of sets and proof by induction; Relations, representation of relations by graphs; properties of relations, equivalence relations and partitions; Partial function theory; Elementary combinatorics; counting techniques; recurrence relation; generating functions. Graph Theory: elements of graph theory, Planar Graphs, Graph Colouring, Euler graph, Hamiltonian path, trees and their applications.

**Reference Materials:**

1. Discrete Mathematical Structure with Application to Computer Science, J. P. Temblay and B Manohar, McGraw-Hill, 2nd Edition.

<b>Course Name:</b> Discrete Structures
---



Week #	Week wise distribution
1.	Mathematical reasoning:
2.	Mathematical reasoning:
3.	introduction to logic,
4.	propositional and
5.	predicate calculus; negation disjunction and
6.	conjunction; implication and equivalence;
7.	truth tables;
8.	predicates; quantifiers;
9.	natural deduction; rules of Inference; methods of proofs;
10.	use in program proving; resolution prin-ciple;
11.	Set theory: Paradoxes in set theory; inductive definition of sets and proof by induction; Relations,
12.	representation of relations by graphs; properties of relations, equivalence relations and partitions; Partial function theory;
13.	Elementary combinatorics; counting techniques;
14.	recurrence relation; generating functions.
15.	Graph Theory: elements of graph theory, Planar Graphs, Graph Colouring,
16.	Wrap up Euler graph, Hamiltonian path, trees and their applications.work and presentations



**Course Name:** Digital Logic Design

**Course Code:** CS-202

**Credit Hours:** 4 (3+1)

**Prerequisites:** None

**Course Objectives:** To introduce the basic tools for design with combinational and sequential digital logic and state machines. To learn simple digital circuits in preparation for computer science.

**Learning Outcome:**

- Upon successful completion, students will be able to:
- Realize complex logic functions utilizing programmable logic.
- Design machines for the purpose of manipulating data streams.
- Design complex digital systems.

**Course Outline:**

Number Systems, Logic Gates, Boolean Algebra, Combination logic circuits and designs, Simplification Methods K-Maps, Quinne, Mc-Cluskey,, Flip Flops and Latches, Asynchronous and Synchronous circuits, Counters, Shift Registers, Shift Registers Counters, Triggered devices & its types. Binary Arithmetic and Arithmetic Circuits, Memory Elements, State Machines. Introduction Programmable Logic Devices (CPLD, FPGA); Lab Assignments using tools such as Verilog HDL/VHDL, MultiSim, etc.

**Reference Materials:**

1. Digital Fundamentals by Thomas L. Floyd, Prentice Hall; 11th edition.
2. Fundamentals of Digital Logic with Verilog Design by Stephen Brown and Zvonko Vranesic, McGraw-Hill; 3rd Edition February 12, 2013). ISBN -10: 0073380547
3. Digital Fundamentals: A Systems Approach by Thomas L. Floyd, Prentice Hall; (July 13, 2012). ISBN-10: 0132933950
4. Digital Design, by M. Morris Mano, Michael D. Ciletti, 4th Edition, Prentice Hall (2007). ISBN-10: 0131989243
5. Fundamentals of Logic Design by Jr. Charles H. Roth and Larry L Kinney, CL Engineering; 6th Edition (March 13, 2009). ISBN-10: 0495471690



Course Name:	
Week #	Week wise distribution
1.	Number Systems, Logic Gates, Boolean Algebra,
2.	Combination logic circuits and designs,
3.	Simplification Methods K-Maps,
4.	Quinne,
5.	Mc-Cluskey,,
6.	Flip Flops and Latches,
7.	Asynchronous and Synchronous circuits,
8.	Counters,
9.	Shift Registers,
10.	Shift Registers Counters,
11.	Triggered devices & its types.
12.	Binary Arithmetic and Arithmetic Circuits, Memory Elements,
13.	State Machines.
14.	Introduction Programmable Logic Devices CPLD, FPGA);
15.	Lab Assignments using tools such as Verilog HDL/VHDL, MultiSim, etc.
16.	Wrap up work and presentations



**Course Name: Operating Systems**

**Course Code: CS-301**

**Credit Hours: 4 (3+1)**

**Pre-requisites: Data Structure and Algorithms**

**Objectives:** To introduce the basic tools for design with combinational and sequential digital logic and state machines. To learn simple digital circuits in preparation for computer science. High-level understand what is an operating system and the role it plays. A highlevel understanding of the structure of operating systems, applications, and the relationship between them.

**Learning Outcome:**

- Able to understand the difference between different types of modern operating systems, virtual machines and their structure of implementation and applications.
- Able to understand the difference between process & thread, issues of scheduling of user level processes / threads and their issues & use of locks, semaphores, monitors for synchronizing multiprogramming with multithreaded systems and implement them in multithreaded programs.
- Able to understand the concepts of deadlock in operating systems and how they can be managed / avoided and implement them in multiprogramming system.
- Able to understand the design and management concepts along with issues and challenges of main memory, virtual memory and file system.
- Able to Able to understand the types of I/O management, disk scheduling, protection and security problems faced by operating systems and how to minimize these problems.

**Course Outline:**

History and Goals, Evolution of operating systems. Operating System: Services, Structure, User Interface. Virtual Machines concept, System Boot, System Calls, Types of System Calls. Processes: Concept, Scheduling, Operations on Processes, Inter-process Communication. Threading: Multithreading Models, Thread Libraries, Threading Issues, processor scheduling; deadlock prevention, avoidance, and recovery; main-memory management; virtual memory management swapping, paging, segmentation and pagereplacement algorithms); Disks management and other input/output devices; file-system structure and implementation; protection and security. Case studies: Linux/Windows Operating Systems.

\*Lab assignments involving different single and multithreaded OS algorithms.



## Reference Materials:

1. Operating System Concepts by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, Wiley; 9th edition December 17, 2012). ISBN -10: 1118063333
2. Operating Systems: Internals and Design Principles by William Stallings, Prentice Hall; 7 edition (March 10, 2011). ISBN-10: 013230998X
3. Applied Operating Systems Concepts by Silberschatz A., Peterson, J.L., & Galvin P.C. Wiley; 8th Edition 2011). ISBN -10: 1118112733

Course Name: Operating System	
Week #	Week wise distribution
1.	History and Goals, Evolution of operating systems.
2.	Operating System: Services, Structure, User Interface.
3.	Virtual Machines concept, System Boot, System Calls,
4.	Types of System Calls. Processes: Concept, Scheduling,
5.	Operations on Processes, Inter-process Communication. Threading: Multithreading Models,
6.	Thread Libraries, Threading Issues, processor scheduling;
7.	deadlock prevention, avoidance, and recovery;
8.	main-memory management; virtual memory management swapping,
9.	paging, segmentation and page replacement algorithms);
10.	Disks management and other input/output devices;
11.	file-system structure and implementation;
12.	file-system structure and implementation;
13.	protection and security.
14.	Case studies: Linux/Windows Operating Systems.
15.	Case studies: Linux/Windows Operating Systems.
16.	Wrap up work and presentations



**Course Name:** Database Systems

**Course Code:** CS-251

**Credit Hours:** 4 (3+1)

**Prerequisites:** None

**Objectives:** To understand several requirement and operations that the analyst needed to analyze, design, and implement the database systems thru DBMS.

### Learning Outcome

- Able to master the basic concepts and understand the applications of database systems.
- Able to construct an Entity-Relationship (E-R) model from specifications and to perform the transformation of the conceptual model into corresponding logical data structures.
- Able to understand the basic database storage structures and access techniques.
- Able to distinguish between good and bad database design, apply data normalization principles, and be aware of the impact of data redundancy on database integrity and maintainability.
- Able to construct queries and maintain a simple database using SQL.
- Able to apply database transaction management and database recovery.

### Course Outline:

Basic Database Concepts, Database Architecture, DB Design Life Cycle, Schema Architecture, Conceptual, Logical and Physical Database Modelling and Design, , Entity Relationship Diagram ERD, Enhanced ERD, Relational Data Model, Mapping ERD to Relational Model, Functional Dependencies and Normalization, Relational Algebra, Structured Query Language SQL), Transaction Processing, Concurrency Control And Recovery Techniques, Query Optimization Concepts.

### Reference Materials:

1. Database Systems A Practical Approach to Design, Implementation, and Management, Thomas Connolly and Carolyn Begg, Prentice Hall; 7th edition (March 10, 2011)
2. Modern Database Management by Fred McFadden, Jeffrey Hoofer, Mary Prescott, Prentice Hall; 11th Edition (July 26, 2012). ISBN-10: 0132662256



3. Fundamentals of Database Systems by R. Elmasri and S. Navathe. 6th Edition, AddisonWesley (2010). ISBN-10: 0136086209.
4. Database Design and Relational Theory: Normal Forms and All That Jazz by C. J. Date, O'Reilly Media; 1st Edition (April 24, 2012). ISBN-10: 1449328016.
5. Modern Database Management by Fred McFadden, Jeffrey Hoofer, Mary Prescott, Prentice Hall; 11th Edition (July 26, 2012). ISBN-10: 0132662256

Course Name: Database System	
Week #	Week wise distribution
1.	Basic Database Concepts, Database Architecture, DB Design Life Cycle,
2.	Schema Architecture, Conceptual,
3.	Logical and Physical Database Modelling and Design, ,
4.	Entity Relationship Diagram ERD,
5.	Enhanced ERD,
6.	Relational Data Model,
7.	Mapping ERD to
8.	Relational Model,
9.	Functional Dependencies and
10.	Normalization,
11.	Relational Algebra,
12.	Structured Query Language SQL),
13.	Transaction Processing,
14.	Concurrency Control And Recovery Techniques,
15.	Query Optimization Concepts.
16.	Wrap up work and presentations



**Course Name: Software Engineering (Introduction to Software Engineering)**

**Course Code: SE-302**

**Credit Hours: 3**

**Prerequisites:** Object Oriented Programming

**Course Objectives:** To learn the basic concepts, enable students to know about the fundamental construct of software and the activities evolve in the development of modular and effective software. To explain the major issues in contemporary software development and maintenance, as related to complex and critical software systems. Become more aware about the practices which typically apply in software development projects from cradle to grave. See software as a part of a larger system, and be aware of the principles of systems engineering as they are relevant to the engineering of software.

### **Learning Outcome**

- Ability to gather and specify requirements of the software projects.
- Ability to analyze software requirements with existing tools
- Able to differentiate different testing methodologies
- Able to understand and apply the basic project management practices in real life projects
- Ability to work in a team as well as independently on software projects

### **Course Outline:**

Overview of SE, Practice & Myths; the Software Processes, Generic Process Models: Framework Activity, Task Set, Process Patterns, Process Improvement, CMM. Prescriptive Process Models: Waterfall Model, Incremental Process Model, Evolutionary Process Model. Specialized Process Models: Component Based Development; The Formal Methods Models, Agile Development. Business Information Systems: Components; Types; and Evaluating methods. SDLC: Phases; System Planning; Preliminary Investigation, SWOT Analysis; the Importance of Strategic Planning; Evaluation of Systems Requests; Requirements Engineering. Difference between Structured Analysis and Object Oriented Analysis; Difference between FDD Diagrams & UML Diagrams; Data & Process Modelling. Diagrams: Data Flow, Context, Conventions, Detailed Level DFD's; the Design Process; Architecture Design Elements, Interface Design Elements, Component-Level Design Elements, Deployments Design Elements; System Architecture, Architectural Styles; User Interface Design; WebApps Interface Design; Software Quality Assurance. Validation Testing, System Testing. Internal and External View of Testing. Project Management. Risk Management; Maintenance and Reengineering.



## Reference Materials:

1. Software Engineering 8E by Ian Sommerville, Addison Wesley; 8th Edition 2006). ISBN -10: 0321313798
2. Software Engineering: A Practitioner's Approach by Roger S. Pressman, McGraw-Hill Science/Engineering/Math; 7th Edition 2009. ISBN -10: 0073375977

Week #	Week wise distribution
1.	Overview of SE, Practice & Myths; the Software Processes, Generic Process Models:
2.	Framework Activity, Task Set, Process Patterns, Process Improvement, CMM. Prescriptive Process Models: Waterfall Model,
3.	Incremental Process Model, Evolutionary Process Model. Specialized Process Models: Component Based Development;
4.	The Formal Methods Models,
5.	Agile Development. Business Information Systems: Components; Types; and
6.	Evaluating methods. SDLC: Phases; System Planning; Preliminary Investigation,
7.	SWOT Analysis; the Importance of Strategic Planning; Evaluation of Systems Requests; Requirements Engineering.
8.	Difference between Structured Analysis and Object Oriented Analysis; Difference between FDD Diagrams & UML Diagrams;
9.	Data & Process Modelling. Diagrams: Data Flow,
10.	Context, Conventions, Detailed Level DFD's; the Design Process;
11.	Architecture Design Elements, Interface Design Elements, Component-Level Design Elements,
12.	Deployments Design Elements; System Architecture, Architectural Styles; User Interface Design;
13.	WebApps Interface Design; Software Quality Assurance.



---

14.	Validation Testing, System Testing. Internal and External View of Testing.
15.	Project Management. Risk Management; Maintenance and Reengineering.
16.	Wrap up work and presentations



**Course Name:** Computer Networks

**Course Code:** CS -352

**Credit Hours:** 3

**Prerequisites:** None

**Objectives:** This Course introduces students to computer networks and concentrates on building a firm foundation for understanding Data Communications and Computer Networks. It is based around the OSI Reference Model that deals with the major issues in the bottom three (Physical, Data Link and Network) layers of the model. Students are also introduced to the areas of Network Security and Mobile Communications. This module provides the student with fundamental knowledge of the various aspects of computer networking and enables students to appreciate recent developments in the area.

**Learning Outcome:** Upon completion of this module, students will be able to:

- Have a good understanding of the OSI Reference Model and in particular have a good knowledge of Layers 1-3.
- Analyze the requirements for a given organizational structure and select the most appropriate networking architecture and technologies;
- Have a basic knowledge of the use of cryptography and network security;
- Specify and identify deficiencies in existing protocols, and then go onto formulate new and better protocols;
- Have an understanding of the issues surrounding Mobile and Wireless Networks.
- Have a working knowledge of datagram and internet socket programming

**Course Outline:**

Data Communication concepts, Analogue and digital Transmission, Noise, Media, Encoding, Asynchronous and Synchronous transmission. Network system architectures OSI, TCP/I P), Error Control, Flow Control, Data Link Protocols, Bridging. Local Area Networks and MAC Layer protocols, Multiplexing, Switched and IP Networks, Inter-networking, Routing. Transport layer protocols TCP, UDP and SCTP. Application Layer Protocols, Wireless LANs, Lab exercises using tools such as Wireshark, OpNet, Packet tracer etc.

**Reference Materials:**

1. Data Communications and Networking, by Behrouz A. Forouzan, McGraw-Hill Science; 5th edition (February 17, 2012). ISBN-10: 0073376221



2. Data and Computer Communications by William Stallings, Prentice Hall; 9th Edition (August 13, 2010. ISBN -10: 0131392050
3. 3.Computer Networks by Andrew S. Tanenbaum and David J. Wetherall, Prentice Hall; 5th Edition (October 7, 2010. ISBN -10: 0132126958
4. 4.Computer Networks and Internets by Douglas E. Comer, Prentice Hall; 5th Edition (April 28, 2008). ISBN-10: 0136066984

Course Name:	
Week #	Week wise distribution
1.	Data Communication concepts,
2.	Analogue and digital Transmission, Noise,
3.	Media,
4.	Encoding,
5.	Asynchronous and Synchronous transmission.
6.	Wireless LANs,
7.	Lab exercises using tools such as Wireshark, OpNet, Packet tracer etc.
8.	Network system architectures OSI, TCP/IP),
9.	Error Control,
10.	Flow Control,
11.	Data Link Protocols,
12.	Bridging. Local Area Networks and
13.	MAC Layer protocols, Multiplexing, Switched and
14.	IP Networks, Inter-networking, Routing. Transport layer protocols TCP,
15.	UDP and SCTP. Application Layer Protocols,
16.	Wrap up work and presentations



**Course Name:** Human Computer Interaction

**Course Code:** CS -452

**Credit Hours:** 3

**Prerequisites:** None

**Objectives:** To Design, implement and evaluate effective and usable graphical computer interfaces. Describe and apply core theories, models and methodologies from the field of HCI. Describe and discuss current research in the field of HCI. Implement simple graphical user interfaces using the Java Swing toolkit. Describe special considerations in designing user interfaces for older adults.

**Learning Outcome:** On completion of this course according to course goals, the student should be able to:

- Understand the basics of human and computational abilities and limitations.
- Understand basic theories, tools and techniques in HCI.
- Understand the fundamental aspects of designing and evaluating interfaces.
- Practice a variety of simple methods for evaluating the quality of a user interface. □Apply appropriate HCI techniques to design systems that are usable by people.

### Course Outline:

The human and the computer and their interaction, Human psychology and ergonomics, Interaction Paradigms, Interaction design basics, HCI in the software process, Design rules, Implementation support, Evaluation techniques, Universal design, User support, ,Cognitive models, Socio-organizational issues and stakeholder requirements, Communication and collaboration models, Task analysis, Dialog notations and design , Models of the system, Modeling rich interaction, Groupware, Ubiquitous computing and augmented realities

### Reference Materials:

1. Human-Computer Interaction by Alan Dix, Janet E. Finlay, Gregory D. Abowd, Russell Beale, Prentice Hall; 3rd Edition December 20, 2003. ISBN-10: 0130461091
2. Human-Computer Interaction: Concepts And Design by J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, T. Carey, Addison Wesley; 1st Edition (April 30, 1994). ISBN10: 0201627698.



3. Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications by Julie A. Jacko, CRC Press; 3 Edition (May 4, 2012. ISBN 10: 1439829438
4. Interaction Design: Beyond Human - Computer Interaction by Yvonne Rogers, Helen Sharp, and Jenny Preece, Wiley; 3rd Edition June 15, 2011). ISBN-10: 0470665769
5. Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules by Jeff Johnson, Morgan Kaufmann; 1st Edition June 3, 2010). ISBN -10: 012375030X.

Course Name: Human computer interaction	
Week #	Week wise distribution
1.	The human and the computer and their interaction, Human psychology and ergonomics,
2.	Interaction Paradigms, Interaction design basics,
3.	HCI in the software process, Design rules,
4.	Implementation support, Evaluation techniques,
5.	Universal design, User support,
6.	Cognitive models,
7.	Socio-organizational issues and
8.	stakeholder requirements,
9.	Communication and collaboration models,
10.	Task analysis,
11.	Dialog notations and design ,
12.	Models of the system,
13.	Modeling rich interaction,
14.	Groupware,
15.	Ubiquitous computing and augmented realities
16.	Wrap up work and presentations



## Computing - Supporting Courses

**Course Name:** Calculus and Analytical Geometry

**Course Code:** MATH-101

**Credit Hours:** 3

**Prerequisites:** None

**Objectives:** To enable the students to think and use the applications of calculus and analytical geometry. Map other concepts with computer science perspectives.

### Learning Outcome:

- Have knowledge related to the fundamentals of calculus and analytical geometry.
- Understand the differentiation integration and their applications.
- Apply the acquired knowledge to solve problems of practical nature.

### Course Outline:

Real Numbers and the Real Line, Coordinates, Lines, and Increments, Functions, Shifting Graphs, Trigonometric Functions. Limits and Continuity: Rates of Change and Limits, Rules for Finding Limits, Target Values and Formal Definitions of Limits, Extensions of the Limit Concept, Continuity, Tangent Lines. Derivatives: The Derivative of a Function, Differentiation Rules, Rates of Change, Derivatives of Trigonometric Functions, The Chain Rule, Implicit Differentiation and Rational Exponents. Applications of Derivatives. Integration: Indefinite Integrals, Integration by Substitution, Definite Integrals, Substitution in Definite Integrals. Numerical Integration. Applications of Integrals. Transcendental Functions: Inverse Functions and Their Derivatives, Natural Logarithms, The Exponential Function,  $a^x$  and  $\log_a x$ , Growth and Decay, L'Hôpital's Rule, Relative Rates of Growth, Inverse Trigonometric Functions, Derivatives of Inverse Trigonometric Functions; Hyperbolic Functions. Conic Sections, Parametrized Curves, and Polar Coordinates. Graphing in Polar Coordinates. Polar Equations for Conic Sections. Integration in Polar Coordinates. Vectors and Analytic Geometry in Space; Vectors in the Plane Dot Products; Vector-Valued Function Cartesian (Rectangular) Coordinates and Vectors in Space. Dot Products; Cross Products. Lines and Planes in Space; Cylinders and Quadric Surfaces; Cylindrical and Spherical Coordinates.



## Reference Materials:

1. Calculus and Analytic Geometry by George B. Thomas and Ross L. Finney, Addison Wesley; 10th Edition (1995) ISBN-10: 0201531747
2. Calculus and Analytical Geometry by Swokowski, Olinick and Pence, 6th Edition, 1994, Brooks/Cole Publishers.
3. Calculus by Howard Anton, Irl C. Bivens, Stephen Davis, Wiley; 10th Edition (2012, ISBN -10: 0470647728
4. Calculus with Analytic Geometry: Student Solution Manual by Howard Anton, Wiley; 5th Edition (1995). ISBN-10: 0471105899

Course Name: Calculus and Analytical Geometry	
Week #	Week wise distribution
1.	Real Numbers and the Real Line, Coordinates, Lines, and Increments, Functions, Shifting Graphs, Trigonometric Functions.
2.	Limits and Continuity: Rates of Change and Limits, Rules for Finding Limits,
3.	Target Values and Formal Definitions of Limits, Extensions of the Limit Concept,
4.	Continuity, Tangent Lines. Derivatives:
5.	The Derivative of a Function,
6.	Differentiation Rules, Rates of Change,
7.	Derivatives of Trigonometric Functions,
8.	The Chain Rule, Implicit Differentiation and Rational Exponents.
9.	Applications of Derivatives. Integration: Indefinite Integrals, Integration by Substitution,
10.	Natural Logarithms, The Exponential Function, $a^x$ and $\log a^x$ , Growth and Decay, L'Hôpital's Rule, Relative Rates of Growth,
11.	Inverse Trigonometric Functions, Derivatives of Inverse Trigonometric Functions; Hyperbolic Functions.



12.	Conic Sections, Parametrized Curves, and Polar Coordinates. Graphing in Polar Coordinates. Polar Equations for Conic Sections.
13.	Integration in Polar Coordinates. Vectors and Analytic Geometry in Space; Vectors in the Plane Dot Products; Vector-Valued Function Cartesian Rectangular) Coordinates and Vectors in Space.
14.	Definite Integrals, Substitution in Definite Integrals. Numerical Integration. Applications of Integrals. Transcendental Functions: Inverse Functions and Their Derivatives,
15.	Dot Products; Cross Products. Lines and Planes in Space; Cylinders and Quadric Surfaces; Cylindrical and Spherical Coordinates.
16.	Wrap up work and presentations



**Course Name: Probability and Statistics**

**Course Code: STAT-151**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives:**

To enable the students to think and use the applications of calculus and analytical geometry. Map other concepts with computer science perspectives. To understand the fundamental concepts of statistics and probability. The role of probability in evaluation in computer science in related domain research.

**Learning Outcome:**

- Use statistical vocabulary, Construct various frequency distributions of grouped and ungrouped data.
- Calculate and interpret descriptive statistics of samples and populations. (Measures of central tendency, measures of dispersion.)
- Calculate simple probabilities, Find the mean and variance of a probability distribution including the binomial distribution.
- Understand and calculate expected values, Calculate the probabilities or scores of normal distributions and the normal approximation of the binomial distribution.
- Use the Central Limit Theorem to calculate the probabilities of the mean for any distribution.
- Formulate, calculate and interpret hypotheses test for one parameter and to compare two parameters, for both large and Small samples, Z and T for one two samples.

**Course Outline:**

Introduction to Statistics and Data Analysis, Statistical Inference, Samples, Populations, and the Role of Probability. Sampling Procedures. Discrete and Continuous Data. Statistical Modeling. Types of Statistical Studies.

Probability: Sample Space, Events, Counting Sample Points, Probability of an Event, Additive Rules, Conditional Probability, Independence, and the Product Rule, Bayes' Rule. Random Variables and Probability Distributions. Mathematical Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables, Chebyshev's Theorem. Discrete Probability Distributions. Continuous Probability Distributions. Fundamental Sampling Distributions and Data Descriptions: Random Sampling,



Sampling Distributions, Sampling Distribution of Means and the Central Limit Theorem. Sampling Distribution of  $S^2$ , t-Distribution, F-Quantile and Probability Plots. Single Sample & One- and Two-Sample Estimation Problems. Single Sample & One- and Two-Sample Tests of Hypotheses. The Use of P-Values for Decision Making in Testing Hypotheses Single Sample & One - and Two-Sample Tests), Linear Regression and Correlation. Least Squares and the Fitted Model, Multiple Linear Regression and Certain, Nonlinear Regression Models, Linear Regression Model Using Matrices, Properties of the Least Squares Estimators.

### Reference Materials:

1. Probability and Statistics for Engineers and Scientists by Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers and Keying E. Ye, Pearson; 9th Edition January 6, 2011). ISBN -10: 0321629116
2. Probability and Statistics for Engineers and Scientists by Anthony J. Hayter, Duxbury Press; 3rd Edition (February 3, 2006), ISBN-10: 0495107573
3. Schaum's Outline of Probability and Statistics, by John Schiller, R. Alu Srinivasan and Murray Spiegel, McGraw-Hill; 3rd Edition 2008). ISBN -10: 0071544259

Course Name:	
Week #	Week wise distribution
1.	Introduction to Statistics and Data Analysis, Statistical Inference, Samples,
2.	Populations, and the Role of Probability. Sampling Procedures. Discrete and Continuous Data. Statistical Modeling. Types of Statistical Studies.
3.	Probability: Sample Space, Events, Counting Sample Points, Probability of an Event,
4.	Additive Rules, Conditional Probability, Independence, and the Product Rule, Bayes' Rule.
5.	Random Variables and Probability Distributions. Estimation Problems.
6.	Single Sample & One- and Two-Sample Tests of Hypotheses. The Use of P-Values for Decision Making in



7.	Mathematical Expectation: Mean of a Random Variable, Variance and Covariance of Random Variables, Means and Variances of Linear Combinations of Random Variables
8.	Chebyshev's Theorem. Discrete Probability Distributions. Continuous Probability Distributions. Fundamental Sampling Distributions and Data Descriptions: Random Sampling, Sampling Distributions,
9.	Sampling Distribution of Means and the Central Limit Theorem.
10.	Sampling Distribution of $S^2$ , t-Distribution,
11.	F Quantile and Probability Plots.
12.	, Single Sample & One- and Two-Sample
13.	Testing Hypotheses Single Sample & One - and Two Sample Tests), Linear Regression and Correlation.
14.	Least Squares and the Fitted Model, Multiple Linear Regression and Certain,
15.	Nonlinear Regression Models, Linear Regression Model Using Matrices, Properties of the Least Squares Estimators.
16.	Wrap up work and presentations



**Course Name:** Linear Algebra

**Course Code:** MTH-251

**Credit Hours:** 3

**Prerequisites:** None

**Objective:** The objective of the course is to provide a rigorous approach towards the solutions of linear models which involves more than one variable. The techniques discussed in this course can be implemented on a wide range of applications from physical world. The matrix algebra will be helpful in performing and understanding of matrix computations on a machine. The eigenvalues, eigenvectors, inner product spaces, orthogonality are useful concepts for the analysis of dynamical systems

**Learning Outcome:** On successful completion of this course unit students will be able to

- solve systems of linear equations by using Gaussian elimination to reduce the augmented matrix to row echelon form or to reduced row echelon form; understand the basic ideas of vector algebra: linear dependence and independence and spanning;
- be able to apply the basic techniques of matrix algebra, including finding the inverse of an invertible matrix using Gauss-Jordan elimination;
- know how to find the row space, column space and null space of a matrix, and be familiar with the concepts of dimension of a subspace and the rank and nullity of a matrix, and to understand the relationship of these concepts to associated systems of linear equations;
- be able to find the eigenvalues and eigenvectors of a square matrix using the characteristic polynomial and will know how to diagonalize a matrix when this is possible, recognize and invert orthogonal matrices, orthogonally diagonalize symmetric matrices, find the change-of-basis matrix with respect to two bases of a vector space, notations of a linear transformation and its matrix.

### **Course Outline:**

Introduction to Vectors. Solving Linear Equations. Elimination Factorization. Vector Spaces and Subspaces. Orthogonality. Determinants. Eigenvalues and Eigenvectors. Linear Transformations. Linear Transformation, Applications of Matrices in Engineering. Graphs and Networks, Markov Matrices, Population, and Economics. Linear Programming. Fourier Series. Linear Algebra for Functions, Linear Algebra for Statistics and Probability, Computer Graphics. Numerical Linear Algebra. Complex Vectors and Matrices. Discrete Transforms and Simple Applications. Cosine



Transform, The Discrete Fourier Transform. Simplification and Factorization of the DFT Matrix. Fast Fourier Transforms. The Discrete Time Fourier Transform. The Z-Transform.

## Reference Materials:

1. Introduction to Linear Algebra by Gilbert Strang, Wellesley Cambridge Press; 4th Edition (February 10, 2009). ISBN-10: 0980232716
2. Linear Algebra: A Modern Introduction by David Poole by Brooks Cole; 3rd Edition (May 25, 2010). ISBN-10: 0538735457
3. Elementary Linear Algebra with Applications by Bernard Kolman, David Hill, 9th Edition, Prentice Hall PTR, 2007. ISBN-10: 0132296543

Course Name:	
Week #	Week wise distribution
1.	Introduction to Vectors. Solving Linear Equations. Elimination Factorization.
2.	Vector Spaces and Subspaces. Orthogonality. DeTerminants. Eigenvalues and Eigenvectors.
3.	Linear Transformations. Linear Transformation, Applications of Matrices in Engineering. Graphs and Networks,
4.	Markov Matrices, Population, and Economics. Linear Programming. Fourier Series.
5.	Linear Algebra for Functions, ,
6.	Linear Algebra for Statistics and Probability
7.	Computer Graphics.
8.	Numerical Linear Algebra.
9.	Complex Vectors and Matrices.
10.	Discrete Transforms and Simple Applications.
11.	Cosine Transform,
12.	The Discrete Fourier Transform.
13.	Simplification and
14.	Factorization of the DFT Matrix.



---

15.	Fast Fourier Transforms. The Discrete Time Fourier Transform. The Z-Transform.
16.	Wrap up work and presentations



**Course Name: Basic Electronics (Physics - I)**

**Credit Hours: 3**

**Prerequisites: None**

**Course Objectives:** The subject aims to provide the student with: An understanding of basic EE abstractions on which analysis and design of electrical and electronic circuits and systems are based, including lumped circuit, digital and operational amplifier abstractions. The capability to use abstractions to analyze and design simple electronic circuits. The ability to formulate and solve the differential equations describing time behavior of circuits containing energy storage elements. The capability to design and construct circuits, take measurements of circuit behavior and performance, compare with predicted circuit models and explain discrepancies.

**Learning Outcomes:** Students will:

- Learn how to develop and employ circuit models for elementary electronic components, e.g., resistors, sources, inductors, capacitors, diodes and transistors;
- Become adept at using various methods of circuit analysis, including simplified methods such as series-parallel reductions, voltage and current dividers, and the node method;
- Gain an intuitive understanding of the role of power flow and energy storage in electronic circuits;
- Develop the capability to analyze and design simple circuits containing non-linear elements such as transistors using the concepts of load lines, operating points and incremental analysis;
- Learn how the primitives of Boolean algebra are used to describe the processing of binary signals and to use electronic components such as MOSFET's as building blocks in electronically implementing binary functions;
- Learn how the concept of noise margin is used to provide noise immunity in digital circuits;
- Be introduced to the concept of state in a dynamical physical system and learn how to analyze simple first and second order linear circuits containing memory elements;
- Be introduced to the concept of singularity functions and learn how to analyze simple circuits containing step and impulse sources;
- Be introduced to the concept of sinusoidal-steady-state (SSS) and to use impedance methods to analyze the SSS response of first and second-order systems;
- Learn how to calculate frequency response curves and to interpret the salient features in terms of poles and zeros of the system function;



- Gain insight into the behavior of a physical system driven near resonance, in particular the relationship to the transient response and the significance of the quality factor Q;
- Learn how operational amplifiers are modeled and analyzed, and to design Op-Amp circuits to perform operations such as integration, differentiation and filtering on electronic signals;
- Be introduced to the concepts of both positive and negative feedback in electronic circuits;
- Learn how negative feedback is used to stabilize the gain of an Op-Amp-based amplifier and how positive feedback can be used to design an oscillator;
- Acquire experience in building and trouble-shooting simple electronic analog and digital circuits

### Course Outline:

Zero Reference Level, Ohm's Law, Linear & Non-Linear Resistors, Cells in Series and Parallel. Resistive Circuits. Resistors, Inductors, Capacitors, Energy Sources. Magnetism and Electromagnetism; Theory of Solid State; P-N Junction; Forward Biased P-N Junction; Forward V/I Characteristics; Reverse Biased P-N Junction; Reverse Saturation Current; Reverse V/I Characteristics, Junction Breakdown, Junction Capacitance. Opto-electronics Devices; Spectral Response of Human Eye; Light Emitting Diode LED; Photoemission Devices, Photomultiplier Tube, Photovoltaic Devices, Bulk Type Photoconductive Cells, Photodiodes, P- N Junction Photodiode, PIN Photodiode, and Avalanche Photodiode; DC Power Supplies; Rectifiers. Filters, Voltage Multipliers, Silicon Controlled Rectifier SCR; The Basic Transistor; Transistor Biasing, Transistor Circuit Configuration; Modulation and Demodulation; Carrier Waves; Integrated Circuits.

### Reference Materials:

1. Basic Electronics Solid State by B. L. Theraja, S Chand & Co Ltd, 5th Edition, 2007, ISBN-13: 978-8121925563
2. Electronic Principles by Albert Paul Malvino, 6th Edition, 1999, ISBN 0-07-115604-6

Course Name:	
Week #	Week wise distribution
1.	Zero Reference Level, Ohm's Law, Linear & Non-Linear Resistors,
2.	Cells in Series and Parallel.
3.	Resistive Circuits. Resistors,
4.	Inductors, Capacitors,



5.	Energy Sources. Magnetism and Electromagnetism;
6.	Theory of Solid State; P-N Junction;
7.	Forward Biased P-N Junction; Forward V/I Characteristics;.
8.	Reverse Biased P-N Junction; Reverse Saturation Current; Reverse V/I Characteristics, Junction Breakdown, Junction Capacitance
9.	Opto-electronics Devices; Spectral Response of Human Eye; Light Emitting Diode LED; Photoemission Devices,
10.	Photomultiplier Tube, Photovoltaic Devices, Bulk Type Photoconductive Cells,
11.	Photodiodes,
12.	P- N Junction Photodiode, PIN Photodiode,
13.	and Avalanche Photodiode; DC Power Supplies; Rectifiers. Filters, Voltage
14.	Multipliers, Silicon Controlled Rectifier SCR; The Basic Transistor; Transistor Biasing,
15.	Transistor Circuit Configuration; Modulation and Demodulation; Carrier Waves; Integrated Circuits.
16.	Wrap up work and presentations



## **Computing – General Education Courses**

**Course Name:** English Composition and Comprehension

**Course Code:** ENG-101

**Credit Hours:** 3

**Prerequisites:** None

**Objectives:** The student should be able to know about the composition process.

### **Learning Outcome:**

- Practices correct English in speaking and writing.
- Comply even complex English language texts.
- Exhibit sound vocabulary and skills to use English in professional life.
- Avoid common errors usually made by the learners of English as second language.

### **Course Outline:**

Punctuation Principles. Spelling Rules. Writing Mechanics. Frequently Confused Words.

Frequently Misused Words, Phrases, Synonyms, Antonyms, Idioms. General Vocabulary. Use of Articles and One, A Little/ A Few, This, That, Care, Like, Love, Hate, Prefer, Wish,

All, Each, Every, Both, Neither, Either, Some, Any, No, None, etc. Interrogatives. Kinds of Nouns.

Prepositions. Possessive, Personal, Reflexive, and Relative Pronouns and Clauses. Classes of Verbs.

Usage of May, Can, Ought, Should, Must, Have To, Need for Obligation, Must, Have, Will and

Should. The Auxiliaries Dare and Used. The Gerund & The Participles. Commands, Requests,

Invitations, Advice, Suggestions. The Subjunctive. The Passive Voice. Indirect Speech.

Conjunctions. Purpose. Clauses: Noun Clauses; Clauses of Reason, Result, Concession, Comparison,

Time. Numerals, Dates, Weights and Measures. Phrasal Verbs. Irregular Verbs. Overview of Present,

Past, Future and Perfect Tenses.

### **Reference Materials:**

1. A Practical English Grammar by A. J. Thomson and A. V. Martinet, 4th Edition Oxford University Press (1986).
2. Basic English Usage by Michael Swan, Oxford Univ Pr (Sd) (January 1986). ISBN-10: 0194311872



3. Functional English In Aglobal Society: Vocabulary Building and Communicative Grammar by Nicanor L. Guintomary Ann R. Sibal Brian D. Villaverde Dept. of Languages, Literature and Humanities College of Arts and Sciences Southern Luzon State University (2012)
4. English Composition and Grammar: Complete Course by John E. Warriner, Harcourt Brace Jovanovich; Complete Course Benchmark Edition (January 1988). ISBN-10: 0153117362
5. Companion to English: Vocabulary (Learners Companion) by George Davidson, PrimEd Publishing (March 1, 2003). ISBN-10: 9814070904

Course Name:	
Week #	Week wise distribution
1.	Punctuation Principles. Spelling Rules. Writing Mechanics. Frequently Confused Words. Frequently Misused Words,
2.	Phrases, Synonyms, Antonyms, Idioms. General Vocabulary.
3.	Use of Articles and One,
4.	A Little/ A Few, This, That, Care, Like, Love, Hate, Prefer, Wish, All, Each, Every, Both, Neither,
5.	Either, Some, Any, No, None, etc. Interrogatives. Kins of Nouns. Prepositions. Possessive, Personal, Reflexive, and Relative Pronouns and Clauses. Classes of Verbs.
6.	Usage of May, Can, Ought, Should, Must, Have To,
7.	Need for Obligation, Must, Have, Will and Should.
8.	The Auxiliaries Dare and Used.
9.	The Gerund & The Participles. Commands, Requests,
10.	Invitations, Advice, Suggestions. The Subjunctive
11.	. The Passive Voice. Indirect Speech. Conjunctions.
12.	Purpose. Clauses: Noun Clauses;
13.	Clauses of Reason, Result, Concession, Comparison,
14.	Time. Numerals, Dates, Weights and Measures.



---

15.	Phrasal Verbs. Irregular Verbs. Overview of Present, Past, Future and Perfect Tenses.
16.	Wrap up work and presentations



**Course Name: Communication and Presentation Skills**

**Course Code: ENG-151**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives:** At the end of the course the student should be able to know about the composition process, grammar and punctuation use, and the way how to communicate.

**Learning Outcome:**

- Display intermediates to advanced level English language skills extending from the Freshman English I course.
- Express an enhanced ability in the general verbal and non-verbal English language Communication Skills which can support real life Electronic Engineering settings requiring team work and leadership skills.
- Practice basic research and writing skills associated to research work, to help them in writing research papers for the contemporary Engineering courses.
- Exhibit such supporting language techniques and personality grooming which cater to the requirements of the corporate sector.

**Course Outline:**

What is Communication, The Importance of Communication, Communication Skills. The Communication Process. Perspectives in Communication. Factors Affecting Communication Perspective. Language as a Representational System. Internal Representation of Our World: Visual, Auditory & inaeesthetic Representational System. Elements of Communication. Communication Styles. Listening: Self-Awareness, Pseudo Listening, Active Listening, Effective Listening, Total Listing. Types of Expression. Rules of Effective Expression. Body Language. Para-language and Meta-messages. Hidden Agendas. Language Models. Communication Styles. Assertiveness. Responding to Criticism. Making Contact. The Art of Conversation. Negotiations. Prejudgment. The Power of Validation. Validation Strategies. Influencing Others. Public Speaking. Preparing Formal Oral Presentations: Design elements, elements of effective Delivery, Tension & Nerves, Handle Questions, Handling Tough Situations, Common Mistakes & Their Remedies, Dealing with Unexpected Disasters, Presentation for International Audience, Dealing People with Disabilities. Interviewing. Elements of Effective Written Communication. Building Rapports.

**Reference Materials:**



1. Effective Communication Skills, MTD Training & Ventus Publishing ApS. 2010) ISBN 978 -87-7681-598-1 (TB1)
2. Messages: The Communication Skills Book by Matthew McKay PhD, Martha Davis PhD, and Patrick Fanning, New Harbinger Publications; 3rd Edition (March 3, 2009). ISBN-10: 1572245921
3. Secrets of Successful Presenters: A Guide for Successful Presenters by Dr. M. A. Pasha & Dr. S. Pasha, Lambert Academic Publishing 2012. ISBN-10:3659217557

Course Name:	
Week #	Week wise distribution
1.	What is Communication, The Importance of Communication, Communication Skills. The Communication Process. Perspectives in Communication.
2.	Factors Affecting Communication Perspective. Language as a Representational System.
3.	Internal Representation of Our World:
4.	Visual, Auditory & inaeesthetic Representational System.
5.	Elements of Communication. Communication Styles.
6.	Expression. Body Language. Para-language and Meta-messages. Hidden Agendas.
7.	Listening: Self-Awareness, Pseudo Listening, Active Listening, Effective Listening, Total Listing. Types of Expression. Rules of Effective
8.	Language Models. Communication Styles. Assertiveness. Responding to Criticism. Making Contact. The Art of Conversation. Negotiations. Prejudgment. The Power of Validation.
9.	Validation Strategies. Influencing Others
10.	Public Speaking. Preparing Formal Oral Presentations: Design elements, elements of effective Delivery,
11.	Tension & Nerves, Handle Questions, Handling Tough Situations,



12.	Common Mistakes & Their Remedies, Dealing with Unexpected Disasters,
13.	Presentation for International Audience,
14.	Dealing People with Disabilities. Interviewing.
15.	Elements of Effective Written Communication. Building Rapports.
16.	Wrap up work and presentations



**Course Name: Technical and Business Writing**

**Course Code: ENG-351**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives:** To enable students with technical writings, writings in business. The reports, applications, leaflets, brochures, handbooks, research papers, manuals feasibility reports and documentation is the main objectives of the course.

**Learning outcome:**

- Demonstrate paragraph unity, support, and coherence
- Use transitions for clarity of thought and readability
- Construct all sentence types
- Recognize and correct major grammatical errors
- Edit their writing assignments
- Give constructive peer feedback
- Use the computer for assignment submissions

**Course Outline:**

Characteristics of Academic, Public, Work and Electronic Communities. Myths and Realities about Writing. Effective Writing: Discovering and Planning; Purpose, Thesis, and Audience; Drafting: Drafting Collaboratively, Drafting in Digital Environments; Revising, Editing, and Proofreading. Paragraphs: Unfocused Paragraphs, Incoherent Paragraphs, Poorly Developed Paragraphs, Special-Purpose Paragraphs. Unclear, Clear and Emphatic Sentences. Reasoning Critically. Reading Critically. Arguing Persuasively & Logically. Designing Documents. Writing in Online Communities. Speaking Effectively. Academic Writing for Social and Natural Sciences: Goals of Writing, Audiences, Writing Tasks, Types of Writing: Abstract, Informative Report, Lab Report, Research Report, Project Reports Public Writing: Goals of Public Writing, Public Audiences, Public Writing Tasks, Types of Public Writing, Public Flyer, Letter to the Editor. Researching and Writing: Types of Research Writing, Developing a Research Question, Developing a Preliminary Thesis, Creating a Research File and a Timeline, Reading and Note taking, Summarizing, Paraphrasing, and Synthesizing. Writing a Position Paper.

**Reference Materials:**



1. Writer's Companion – The Longman by Chris M. Anson, Robert A. Schwegler and Marcia F. Muth, Pearson Longman, 4th Edition 2007. ISBN10: 0-20556-252-3
2. Technical English: Writing, Reading, and Speaking by Pickett and Laster. 8th Edition
3. The Technical Writer's Companion by Alfred, Gerald, Charles T. Brusaw and Walter E. Oliu, 3rd Edition. ISBN 0-312-25978-6.

Course Name:	
Week #	Week wise distribution
1.	Characteristics of Academic, Public, Work and Electronic Communities. Myths and Realities about Writing.
2.	Effective Writing: Discovering and Planning; Purpose, Thesis, and Audience; Drafting: Drafting Collaboratively, Drafting in Digital Environments;
3.	Revising, Editing, and Proofreading. Paragraphs: Unfocused Paragraphs, Incoherent Paragraphs, Poorly Developed Paragraphs,
4.	Special-Purpose Paragraphs. Unclear, Clear and Emphatic Sentences. Reasoning Critically. Reading Critically. Arguing Persuasively & Logically.
5.	Designing Documents. Writing in Online Communities. Speaking Effectively.
6.	Academic Writing for Social and Natural Sciences: Goals of Writing, Audiences,
7.	Writing Tasks, Types of Writing: Abstract, Informative Report, Lab Report,
8.	Research Report,
9.	Project Reports Public Writing:
10.	Goals of Public Writing, Public Audiences,
11.	Public Writing Tasks, Types of Public Writing,
12.	Public Flyer, Letter to the Editor.
13.	Researching and Writing: Types of Research Writing,



---

14.	Developing a Research Question, Developing a Preliminary Thesis, Creating a Research File and a Timeline,
15.	Reading and Note taking, Summarizing, Paraphrasing, and Synthesizing. Writing a Position Paper.
16.	Wrap up work and presentations



**Course Name: Islamic Studies**

**Course Code: ISL-101**

**Credit Hours: 2**

**Prerequisites: None**

**Objectives:** This course is aimed at: To provide Basic information about Islamic Studies, enhance understanding of the students regarding Islamic Civilization, improve Students skill to perform prayers and other worships and enhance the skill of the students for understanding of issues related to faith and religious life.

**Learning Outcome:** The students are able:

- To know about the basics of the religion of Islam.
- To know about the personal words of Allah Taa'la.
- To clear their minds about the sources of Shariah.
- To awake the students about their political, social, cultural economic and religious.
- They are able to understand the legal political, economic, social, moral and spiritual aspects of Islam and to implement in their daily lives

**Course Outline:** Introduction to Quranic Studies, Basic Concepts of Quran, History of Quran, Uloom-ul-Quran, Study of Selected Text of Holly Quran, Verses of Surah Al-Baqra Related to Faith Verse No -284-286), Verses of Surah Al-Hujrat Related to Adab Al-Nabi Verse No -1-18), Study of Selected Text of Holly Quran, Verses of Surah Al-Ihzab Related to Adab al-Nabi (Verse No.6,21,40,56,57,58. ), Verses of Surah Al-Hashar (18,19,20) Related to thinking, Day of Judgment, Verses of Surah Al-Saf Related to Tafakar,Tadabar (Verse No-1,14) , Seerat of Holy Prophet S.A.W I, Seerat of Holy Prophet S.A.W II, Introduction To Sunnah, Selected Study from Text of Hadith, Introduction To Islamic Law & Jurisprudence, Islamic Culture & Civilization, Islamic Economic System, Political System of Islam, Islamic History, and Social System of Islam

**Reference Materials:**

1. Hameed ullah Muhammad, “Emergence of Islam” , IRI, Islamabad
2. Hameed ullah Muhammad, “Muslim Conduct of State”
3. Hameed ullah Muhammad, „Introduction to Islam
4. Mulana Muhammad Yousaf Islahi,”
5. Hussain Hamid Hassan, “An Introduction to the Study of Islamic Law” leaf Publication Islamabad, Pakistan.



6. Ahmad Hasan, "Principles of Islamic Jurisprudence" Islamic Research Institute, International Islamic University, Islamabad (1993)
7. Mir Waliullah, "Muslim Jurisprudence and the Quranic Law of Crimes" Islamic Book Service (1982)
8. H.S. Bhatia, "Studies in Islamic Law, Religion and Society" Deep & Deep Publications New Delhi (1989)
9. Dr. Muhammad Zia-ul-Haq, "Introduction to Al Sharia Al Islamia" Allama Iqbal Open University, Islamabad (2001)

Course Name:	
Week #	Week wise distribution
1.	This course is aimed at: To provide Basic information about Islamic Studies,
2.	enhance understanding of the students regarding Islamic Civilization,
3.	improve Students skill to perform prayers
4.	and other worships
5.	and enhance the skill of the students for understanding of issues related to faith
6.	enhance the skill of the students for understanding of issues related to faith
7.	enhance the skill of the students for understanding of issues related to faith
8.	enhance the skill of the students for understanding of issues related to faith
9.	enhance the skill of the students for understanding of issues related to faith
10.	enhance the skill of the students for understanding of issues related to faith
11.	religious life.
12.	religious life.



---

13.	religious life.
14.	and religious life.
15.	Wrap up work and presentations



**Course Name: Pakistan Studies**

**Course Code: PS-151**

**Credit Hours: 2**

**Prerequisites: None**

**Objectives:** Develop vision of historical perspective, government, politics, contemporary Pakistan, ideological background of Pakistan. Study the process of governance, national development, issues arising in the modern age and posing challenges to Pakistan.

**Learning Outcome:**

- Ability to know historical perspective, politics, contemporary Pakistan and ideological background of Pakistan
- Understanding about process of governance, national development, issues arising in modern age and posing challenges to Pakistan

**Course Outline:**

Historical Perspective, Ideological rationale with special reference to Sir Syed Ahmed Khan, Allama Muhammad Iqbal and Quaid-e-Azam Muhammad Ali Jinnah, Indus Civilization, Muslim advent, Location and geo-physical features, Government and Politics in Pakistan, Political and constitutional phases, Contemporary Pakistan

**Reference Materials:**

1. Burki, Shahid Javed. State & Society in Pakistan, The Macmillan Press Ltd 1980.
2. Akbar, S. Zaidi. Issue in Pakistan's Economy. Karachi: Oxford University Press, 2000.
3. S.M. Burke and Lawrence Ziring. Pakistan's Foreign policy: An Historical analysis. Karachi: Oxford University Press, 1993.
4. Mehmood, Safdar. Pakistan Political Roots & Development. Lahore, 1994.
5. Wilcox, Wayne. The Emergence of Bangladesh., Washington: American Enterprise, Institute of Public Policy Research, 1972.
- 1.6. Mehmood, Safdar. Pakistan Kayyun Toota, Lahore: Idara-e-Saqafat-e-Islamia, Club Road, nd.
6. Amin, Tahir. Ethno - National Movement in Pakistan, Islamabad: Institute of Policy Studies, Islamabad.
7. Ziring, Lawrence. Enigma of Political Development. Kent England: WmDawson & sons Ltd, 1980.



8. Zahid, Ansar. History & Culture of Sindh. Karachi: Royal Book Company, 1980.
9. Afzal, M. Rafique. Political Parties in Pakistan, Vol. I, II & III. Islamabad: National Institute of Historical and cultural Research, 1998.
10. Sayeed, Khalid Bin. The Political System of Pakistan. Boston: Houghton Mifflin, 1967.
11. Aziz, K.K. Party, Politics in Pakistan, Islamabad: National Commission on Historical and Cultural Research, 1976.

Course Name:	
Week #	Week wise distribution
1.	Historical Perspective,
2.	Ideological rationale with special reference to Sir Syed Ahmed Khan,
3.	Allama Muhammad
4.	Iqbal and Quaid-e-Azam Muhammad Ali Jinnah,
5.	Indus Civilization,
6.	Muslim advent, Location and geo-physical features,
7.	Government and Politics in Pakistan,
8.	Political and constitutional phases,
9.	Political and constitutional phases,
10.	Political and constitutional phases,
11.	Political and constitutional phases,
12.	Contemporary Pakistan
13.	Contemporary Pakistan
14.	Contemporary Pakistan
15.	Contemporary Pakistan
16.	Wrap up work and presentations



**Course Name: Professional Practices**

**Course Code: MGT-201**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives:** To learn the professional approach of software development using latest technologies, security and copyright issues etc. The role of ethics in software development.

**Learning Outcome:** At the completion of this course, students will be able to:

- Work proficiently and effectively in small teams;
- Understand the need for lifelong learning for continuous professional development;
- Present technical material in an interesting manner for a non-technical audience;
- Explain in basic terms the ethical responsibilities of professional engineers and apply this knowledge in simple scenarios.

**Course Outline:**

Computing Profession, Computing Ethics, Philosophy of Ethics. The Structure of Organizations, Finance and Accounting, Anatomy of a Software House, Computer Contracts, Intellectual Property Rights, The Framework of Employee Relations Law and Changing Management Practices, Human Resource Management and IT, Health and Safety at Work, Software Liability, Liability and Practice, Computer Misuse and the Criminal Law, Regulation and Control of Personal Information. Overview of the British Computer Society Code of Conduct, IEEE Code of Ethics, ACM Code of Ethics and Professional Conduct, ACM/IEEE Software Engineering Code of Ethics and Professional Practice. Accountability and Auditing, Social Application of Ethics.

**Reference Materials:**

1. Professional Issues in Software Engineering by Frank Bott, Allison Coleman, Jack Eaton and Diane Rowland, CRC Press; 3rd Edition 2000). ISBN-10: 0748409513
2. Computer Ethics by Deborah G. Johnson, Pearson; 4th Edition January 3, 2009). ISBN10: 0131112414
3. A Gift of Fire: Social, Legal, and Ethical Issues for Computing and the Internet 3 rd Edition) by Sara Baase, Prentice Hall; 3rd Edition 2008). ISBN-10: 0136008488
4. Applied Professional Ethics by Gregory R. Beabout, University Press of America 1993). ISBN -10: 0819193747.



5. The Dark Side of Software Engineering: Evil on Computing Projects by Johann Rost and Robert L. Glass, Wiley-IEEE Computer Society Pr; 1st Edition (2011. ISBN -10: 0470597178

Course Name:	
Week #	Week wise distribution
1.	Computing Profession, Computing Ethics,
2.	Philosophy of Ethics. The Structure of Organizations,
3.	Finance and Accounting, Anatomy of a Software House, Computer Contracts,
4.	Intellectual Property Rights,
5.	The Framework of Employee Relations Law and Changing Management Practices,
6.	Human Resource Management and IT,
7.	Health and Safety at Work,
8.	Software Liability,
9.	Liability and Practice,
10.	Computer Misuse and the Criminal Law,
11.	Regulation and Control of Personal Information.
12.	Overview of the British Computer Society Code of Conduct,
13.	IEEE Code of Ethics,
14.	ACM Code of Ethics and Professional Conduct, ACM/IEEE
15.	Software Engineering Code of Ethics and Professional Practice. Accountability and Auditing, Social Application of Ethics.
16.	Wrap up work and presentations



**Course Name:** Introduction to Information and Communication Technologies

**Course Code:** CS-101

**Credit Hours:** 3 (2+1)

**Prerequisites:** None

**Objectives:** To learn the basic concepts of Computer system and its types, about hardware and its types, operating system and its types, application software and its types especially MS Office.

**Learning Outcome:** At the completion of this course, students will be able to know:

- The basic concepts of Computer system and its types
- Hardware and its types,
- Operating system and its types,
- Application software and its types especially MS Office
- Work practically in MS office

**Course Outline:**

Basic Definitions & Concepts, Hardware: Computer Systems & Components. Storage Devices, Number Systems, Software: Operating Systems, Programming and Application Software, Introduction to Programming, Databases and Information Systems, Networks, Data Communication, The Internet, Browsers and Search Engines, The Internet: Email, Collaborative Computing and Social Networking, The Internet: E-Commerce, IT Security and other issues, Project Week, Review Week

**Reference Materials:**

1. *Introduction to Computers* 6<sup>th</sup> International Edition, Peter, N. McGraw-Hill
2. *Using Information Technology: A Practical Introduction to Computer & Communications*, 6<sup>th</sup> Edition. Williams, S. McGraw-Hills.
3. *Computers, Communications & information: A user's introduction*, Sarah, E. Hutchinson. Stacey, C. Swayer.
4. *Fundamentals of Information Technology*, Alexis L Mathewsleon LeonPress.

Course Name:	
Week #	Week wise distribution
1.	Basic Definitions & Concepts,
2.	Basic Definitions & Concepts,



3.	Hardware:
4.	Computer Systems & Components.
5.	Computer Systems & Components.
6.	Computer Systems & Components.
7.	Computer Systems & Components.
8.	Storage Devices,
9.	Number Systems,
10.	Software: Operating Systems,
11.	Programming and Application Software, Introduction to Programming,
12.	Databases and Information Systems, Networks,
13.	Data Communication, The Internet, Browsers and Search Engines,
14.	The Internet: Email, Collaborative Computing and Social Networking,
15.	The Internet: E-Commerce, IT Security and other issues, Project Week, Review Week
16.	Wrap up work and presentations



## Computer Science – Core Courses

**Course Name:** Computer Organization and Assembly Language

**Credit Hours:** 4 (3+1)

**Prerequisites:** Digital Logic and Design

### Objectives:

To learn the basic building blocks of computer system, their functions and the way these components interacts.

**Learning Outcome:** At the completion of this course, students will be able to know about:

- Microprocessor Bus Structure
- Addressing, Data and Control
- Arithmetic and Logic, Programmed Control, Stack and its operation
- Peripheral Control Interrupts, Assembler and Debugger
- Manipulate and translate machine and assembly code, Describe actions inside the processing chip.

### Course Outline:

Microprocessor Bus Structure: Addressing, Data and Control, Introduction to Registers and Flags. Addressing Modes, Instruction sets including Data Movement, Arithmetic and Logic, Programmed Control, Stack and its operation. Peripheral Control Interrupts. Introduction to the Assembler and Debugger, Manipulate and translate machine and assembly code, Describe actions inside the processing chip.

### Reference Materials:

1. *The Intel Microprocessor 8<sup>th</sup> ed*, Barry B Brey.
2. *Assembly Language for Intel-based Computers, 6<sup>th</sup> Ed* Irvine,  
<http://vig.prenhall.com/catalog/academic/product/0,1144,0132383101,00> *The 8086/8088 Microprocessor 4<sup>th</sup> Edition* by Avtar Singh.

Course Name:	
Week #	Week wise distribution
1.	Microprocessor Bus Structure: Addressing, Data and Control,
2.	Introduction to Registers and Flags.
3.	Introduction to Registers and Flags.



4.	Introduction to Registers and Flags.
5.	Flags.
6.	Addressing Modes,
7.	Instruction sets including Data Movement
8.	Data Movement
9.	Data Movement
10.	Arithmetic and Logic,
11.	, Programmed Control,
12.	Stack and its operation. Peripheral Control Interrupts.
13.	Introduction to the Assembler and Debugger,
14.	Manipulate and translate machine and assembly code,
15.	Describe actions inside the processing chip.
16.	Wrap up work and presentations



**Course Name:** Theory of Automata

**Course Code:** CS-253

**Credit Hours:** 3

**Prerequisites:** None

**Objectives:**

To enable the students to understand the problem solving techniques using state machine, theoretical models and logic machines, language parsing mechanisms etc.

**Learning Outcomes:** At the completion of this course, students will be able to:

- Understand the problem solving techniques using state machine,
- Theoretical models and logic machines,
- Language parsing mechanisms and
- Computations

**Course Outline:**

Finite State Models: Language definitions preliminaries, Regular expressions/Regular languages, Finite automata FAs, Transition graphs TGs), NFAs, Kleene's theorem, Transducers automata with output), Pumping lemma and non regular language Grammars and PDA: Context free grammars, Derivations, derivation trees and ambiguity, Simplifying CFLs , Normal form grammars and parsing, Decidability, Context sensitive languages, grammars and linear bounded automata LBA), Chomsky's hierarchy of grammars Turing Machines Theory: Turing machines, Post machine, Variations on TM, TM encoding, Universal Turing Machine, Defining Computers by TMs.

**Reference Materials:**

1. *Introduction to computer theory*, Daniel I. A. Cohen, 2<sup>nd</sup> Edition 2. *Automata, Computability and Complexity: Theory and Applications*, by Elaine Rich, 2011 3.
2. *An Introduction to Formal Languages and Automata*, By Peter Linz, 4<sup>th</sup> edition, Jones & Bartlett Publishers, 2006
3. *Theory of Automata, Formal Languages and Computation*, By S. P. Eugene, Kavier, 2005, New Age Publishers, ISBN 10): 81 -224-2334-5, ISBN (13): 978-81-224-2334-1.
4. *Introduction to Automata Theory, Languages, and Computation*, John Hopcroft and Jeffrey Ullman, 2<sup>nd</sup> edition, 2001, Addison-Wesley.

<b>Course Name:</b>
---------------------



Week #	Week wise distribution
1.	Finite State Models: Language definitions preliminaries, Regular expressions/Regular languages,
2.	Finite automata FAs, Transition graphs TGs),
3.	NFAs, Kleene's theorem,
4.	Transducers automata with output),
5.	Pumping lemma and
6.	non regular language Grammars and PDA:
7.	Context free grammars, Derivations, derivation trees and ambiguity,
8.	Simplifying CFLs , Normal form grammars and parsing,
9.	Decidability, Context sensitive languages,
10.	grammars and linear bounded automata LBA),
11.	Chomsky's hierarchy of grammars Turing Machines Theory:
12.	Turing machines,
13.	Post machine, Variations on TM
14.	TM encoding, Universal Turing Machine,
15.	, Defining Computers by TMs.
16.	Wrap up work and presentations



**Course Name:** Design and Analysis of Algorithms

**Course Code:** CS-252

**Credit Hours:** 3

**Prerequisites:** Data Structure and Algorithms

**Objectives:**

Introduces to students asymptotic performance of algorithm, methods of analysis, important paradigms for designing and analysis of algorithms, and synthesize efficient algorithms.

**Learning Outcomes:** At the completion of this course, students will be able to:

- Basic of algorithms and its analysis parameters
- Asymptotic performance of algorithm,
- Methods of analysis, important paradigms for designing and analysis of algorithms, and
- Synthesize efficient algorithms.

**Course Outline:**

Introduction; Asymptotic notations; Recursion and recurrence relations; Divide-and-conquer approach; Sorting; Search trees; Heaps; Hashing; Greedy approach; Dynamic programming; Graph algorithms; Shortest paths; Network flow; Disjoint Sets; Polynomial and matrix calculations; String matching; NP complete problems; Approximation algorithms.

**Reference Materials:**

1. *Introduction to Algorithms*, T. H. Cormen, C. E. Leiserson, and R. L. Rivest, MIT Press, McGraw-Hill, 3<sup>rd</sup> Edition, New York, NY, 2010.
2. *Algorithms in C++*; Robert Sedgewick

Course Name:	
Week #	Week wise distribution
1.	Introduction; Asymptotic notations;
2.	Introduction; Asymptotic notations;
3.	Introduction; Asymptotic notations;
4.	Recursion and recurrence relations;
5.	Recursion and recurrence relations;
6.	Recursion and recurrence relations;



7.	Divide-and-conquer approach;
8.	Sorting; Search trees;
9.	Heaps;
10.	Hashing;
11.	Greedy approach;
12.	Dynamic programming; Graph algorithms;
13.	Shortest paths; Network flow; Disjoint Sets;
14.	Polynomial and matrix calculations;
15.	String matching; NP complete problems; Approximation algorithms.
16.	Wrap up work and presentations



Course Name: **Artificial Intelligence**

Course Code: CS-351

**Credit Hours:** 3 (2+1)

**Prerequisites:** Data Structure and Algorithms

**Objective:** Explain the basic knowledge representation, problem solving, and learning methods of Artificial Intelligence. Understand the role of knowledge representation, problem solving, and learning in intelligent-system engineering.

**Learning Outcomes:** At the completion of this course, students will be able to learn:

- The basic concepts
- Knowledge representation and problem solving techniques
- Learning methods of Artificial Intelligence
- Understand the role of knowledge representation, problem solving, and learning in intelligent-system engineering.

### **Course Outline:**

Artificial Intelligence: Introduction, AI Paradigms and Hypothesis, Intelligent Agents. Difference between Cybernetic Intelligence and Artificial Intelligence, Objectives and Scope of Weak AI and Strong AI, Problem-solving: Solving Problems by Searching, Informed Search and Exploration, Constraint Satisfaction Problems, Adversarial Search. Knowledge and reasoning: Logical Agents, First-Order Logic, Inference in First-Order Logic, Knowledge Representation. Planning and Acting in the Real World. Uncertain knowledge and reasoning: Uncertainty, Probabilistic Reasoning, Probabilistic Reasoning over Time, Making Simple Decisions, Making Complex Decisions. Learning: Learning from Observations, Knowledge in Learning; Learning Methods, Reinforcement Learning. Communicating, perceiving, and acting: Communication, Probabilistic Language Processing, Perception and Robotics. Introduction to LISP/PROLOG and Expert Systems (ES) and Applications; Artificial General Intelligence, Issues in Safe AI, Introduction to Cognitive and Conscious Systems.

### **Reference Materials:**

1. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving:* International Edition by George F. Luger, 6<sup>th</sup> edition: Pearson Education, 2008.
2. *Artificial Intelligence: A Modern Approach*, by Stuart Jonathan Russell, Peter Norvig, John F. Canny, 3<sup>rd</sup> Edition, Prentice Hall.



3. *Prolog Programming for Artificial Intelligence*, Ivan Bratko, 3<sup>rd</sup> Edition, Addison Wesley, 2001.

Course Name:	
Week #	Week wise distribution
1.	Artificial Intelligence: Introduction, AI Paradigms and Hypothesis,
2.	Intelligent Agents. Difference between Cybernetic Intelligence and Artificial Intelligence,
3.	Objectives and Scope of Weak AI and Strong AI, Problem-solving:
4.	Solving Problems by Searching, Informed Search and Exploration,
5.	Constraint Satisfaction Problems, Adversarial Search. Knowledge and reasoning:
6.	Logical Agents, First-Order Logic, Inference in First-Order Logic, Knowledge Representation.
7.	Planning and Acting in the Real World. Uncertain knowledge and reasoning: Uncertainty,
8.	Probabilistic Reasoning, Probabilistic Reasoning over Time, Making Simple Decisions,
9.	Making Complex Decisions. Learning: Learning from Observations, Knowledge in Learning;
10.	Learning Methods, Reinforcement Learning. Communicating, perceiving, and acting: Communication,
11.	Probabilistic Language Processing, Perception and Robotics.
12.	Introduction to LISP/PROLOG and Expert Systems (ES) and Applications;
13.	Artificial General Intelligence,
14.	Issues in Safe AI,
15.	Introduction to Cognitive and Conscious Systems.
16.	Wrap up work and presentations



Course Name: **Computer Architecture and Organization**

**Credit Hours:** 3

**Prerequisites:** Digital Logic and Design

**Objective:** To have a thorough understanding of the basic structure and operation of a digital Computer. To discuss in detail the operation of the arithmetic unit including the algorithms & Implementation of fixed-point and floating-point addition, subtraction, multiplication & division, communicating with I/O devices and standard I/O Interfaces and the hierarchical memory system.

**Learning Outcomes:** At the completion of this course, students will be able to learn: The basic structure and operation of a digital Computer

- The details of the operation of the arithmetic unit including the algorithms & Implementation of fixed-point and floating-point
- Addition, subtraction, multiplication & division, communicating with I/O devices and
- Standard I/O Interfaces and the hierarchical memory system.

**Course Outline:**

The design of computer systems and components. Processor design, instruction set design, and addressing; control structures and microprogramming; memory management, caches, and memory hierarchies; and interrupts and I/O structures. Pipelining of processor Issues and Hurdles, exception handling, Parallelism, Multiprocessor Systems.

**Reference Materials:**

1. *Computer Architecture: A Quantitative Approach* by Hennessy & Patterson, Morgan & Kauffman Series 2006) 4<sup>th</sup> Edition.
2. *Computer Organization & Design: The Hardware/Software Interface* By Patterson & Hennessy, Morgan & Kauffman Series (2008) 4<sup>th</sup> Edition.

Course Name:	
Week #	Week wise distribution
1.	The design of computer systems and components.
2.	Processor design,
3.	instruction set design, and addressing;
4.	control structures and microprogramming;



5.	control structures and microprogramming;
6.	control structures and microprogramming;
7.	memory management,
8.	memory management,
9.	memory management,
10.	caches,
11.	and memory hierarchies;
12.	and interrupts and I/O structures.
13.	Pipelining of processor Issues and Hurdles,
14.	exception handling, Parallelism
15.	Multiprocessor Systems.
16.	Wrap up work and presentations



**Course Name:** Compiler Construction

**Course Code:** CS-303

**Credit Hours:** 3

**Prerequisites:** Theory of Automata

**Objective:** Explain the basic techniques that underlie the practice of Compiler Construction. The course will introduce the theory and tools that can be standardly employed in order to perform syntax-directed translation of a high-level Programming language into an executable code.

**Learning Outcomes:** At the completion of this course, students will be able to learn:

- The basic techniques that underlie the practice of Compiler Construction
- The course will introduce the theory and tools
- The tools that can be employed in order to perform syntax-directed translation of a highlevel Programming language into an executable code.

### Course Outline:

Introduction to interpreter and compiler. Compiler techniques and methodology; Organization of compilers; Lexical and syntax analysis; Parsing techniques. Types of parsers, top-down parsing, bottom-up parsing, Type checking, Semantic analyser, Object code generation and optimization, detection and recovery from errors.

### Reference Materials:

1. *Compilers: Principles, Techniques, and Tools* By Alfred V. Aho, RaviSethi, Jeffrey D. Ullman, Contributor Jeffrey D. Ullman, Addison-Wesley Pub. Co., 2<sup>nd</sup> edition, 2006 Original from the University of Michigan
2. *Modern Compiler Design*, by Dick Grune, Henri E. Bal, Criel J. H.Jacobs, Koen G. Langendoen, John Wiley, 2000.
3. *Modern Compiler Implementation in C*, by Andrew W. Appel, MaiaGinsburg, Contributor Maia Ginsburg, Cambridge University Press, 2004.
4. *Modern Compiler Design* by Dick Grune, Henri E. Bal, Criel J. H. Jacobs,Koen G. Langendoen, 2003, John Wiley & Sons.

Course Name:	
Week #	Week wise distribution



1.	Introduction to interpreter and compiler.
2.	Compiler techniques and methodology;
3.	Organization of compilers;
4.	Organization of compilers;
5.	Lexical and syntax analysis;
6.	Parsing techniques.
7.	Types of parsers, top-down parsing,
8.	top-down parsing,
9.	bottom-up parsing,
10.	Type checking,
11.	Semantic analyser,
12.	Semantic analyser,
13.	Object code generation and
14.	optimization, detection and.
15.	recovery from errors
16.	Wrap up work and presentations



## Course Name: Information Security

Course Code: CS-363

Credit Hours: 3

**Prerequisites:** Data Communications and Computer Networks

**Objective:** Explain the basic techniques that underlie the practice of Information Security. To strengthen internal control and prevent unauthorized and improper access to data, thereby ensuring the appropriate protection of information assets. To appropriately protect the confidentiality and integrity of information assets.

**Learning Outcomes:** At the completion of this course, students will be able to learn:

- The basic techniques that underlie the practice of Information Security
- The strengthen internal control and prevent unauthorized and improper access to data
- Thereby ensuring the appropriate protection of information assets
- To appropriately protect the confidentiality and integrity of information assets

### Course Outline:

Basic notions of confidentiality, integrity, availability; authentication models; protection models; security kernels; Encryption, Hashing and Digital Signatures; audit; intrusion detection and response; database security, host-based and network-based security issues operational security issues; physical security issues; personnel security; policy formation and enforcement; access controls; information flow; legal and social issues; identification and authentication in local and distributed systems; classification and trust modeling; risk assessment.

### Reference Materials:

1. 2. *Computer Security: Art and Science* Cryptography and Network Security , Matthew Bishopby William Stalling 6th Edition, 2012

3. *Principles of Information Security* 3<sup>rd</sup>E by Michael E. Whitman and Herbert J. Mattord

Course Name:	
Week #	Week wise distribution
1.	Basic notions of confidentiality, integrity, availability; authentication models;
2.	protection models; security kernels; Encryption, Hashing and Digital Signatures;
3.	audit; intrusion detection and response; database security, host-based



4.	and network-based security
5.	issues operational security issues;
6.	physical security issues
7.	personnel security;
8.	policy formation and enforcement;
9.	access controls; information flow;
10.	legal and social issues;
11.	identification and
12.	authentication in local and
13.	distributed systems;
14.	classification and trust modeling;
15.	risk assessment.
16.	Wrap up work and presentations



## Computer Science – Supporting Courses

**Course Name:** Numerical Computing

**Course Code:** MATH-302

**Credit Hours:** 3 (2+1)

**Prerequisites:** Calculus and Analytical Geometry

**Objective:** Numerical analysis is the study of algorithms that use numerical approximation for the problems of mathematical analysis. Numerical methods, based upon sound computational mathematics, are the basic algorithms underpinning computer predictions in modern systems science.

**Learning Outcomes:** At the completion of this course, students will be able to learn:

- Numerical analysis is the study of algorithms that use numerical approximation
- The problems of mathematical analysis
- Numerical methods, based upon sound computational mathematics
- The basic algorithms underpinning computer predictions in modern systems science

### Course Outline:

The concepts of efficiency, reliability and accuracy of a method; Minimising computational errors; Theory of Differences, Difference Operators, Difference Tables, Forward Differences, Backward Differences and Central Differences. Mathematical Preliminaries, Solution of Equations in one variable, Interpolation and Polynomial Approximation, Numerical Differentiation and Numerical Integration, Initial Value Problems for Ordinary Differential Equations, Direct Methods for Solving Linear Systems, Iterative Techniques in Matrix Algebra, Solution of non-linear equations.

### Reference Materials:

1. *Numerical Methods in Scientific Computing* by Germund, D. Åke, B.
2. *Numerical Methods for Scientific Computing* by J. H. Heinbockel.
3. *Numerical Analysis* by I. A. Khubaza.
4. *Numerical Analysis and Programming* by Shan S Kuo.
5. *Numerical Analysis* by Berden, F.
6. *Numerical Analysis* by Gerald.

<b>Course Name:</b>	
<b>Week #</b>	<b>Week wise distribution</b>



1.	The concepts of efficiency, reliability and accuracy of a method; Minimising computational errors; Theory of Differences, Difference Operators, Difference Tables,
2.	Forward Differences, Backward Differences and Central Differences.
3.	Mathematical Preliminaries,
4.	Mathematical Preliminaries,
5.	variable,
6.	Solution of Equations in one
7.	Interpolation and
8.	Polynomial Approximation,
9.	Numerical
10.	Differentiation and Numerical Integration,
11.	Initial Value Problems for Ordinary Differential Equations,
12.	Direct Methods for Solving
13.	Linear Systems,
14.	Iterative Techniques in Matrix Algebra,
15.	Solution of non-linear equations.
16.	Wrap up work and presentations



**Course Name: Multivariate Calculus**

**Course Code: MATH-**

**Credit Hours: 3**

**Prerequisites:** Calculus and Analytical Geometry

**Objective:** Explain a clear understanding of the fundamental concepts of multivariable calculus and a range of skills allowing the students to work effectively with the concepts like Derivatives as rates of change, computed as a limit of ratios and Integrals as a 'sum,' computed as a limit of Riemann sums.

**Learning Outcomes:** At the completion of this course, students will be able to learn:

- The fundamental concepts of multivariable calculus
- Range of skills allowing the students to work effectively with the concepts like o Derivatives as rates of change, o Computed as a limit of ratios and o Integrals as a sum, computed as a limit of Riemann sums

**Course Outline:**

Functions of Several Variables and Partial Differentiation. Multiple Integrals, Line and Surface Integrals. Green's and Stoke's Theorem. Fourier Series: periodic functions, Functions of any period P-2L, Even & odd functions, Half Range expansions, Fourier Transform; Laplace Transform, Z-Transform.

**Reference Materials:**

1. Multivariable Calculus, 6<sup>th</sup> edition James, Stewart 2007 Cengage Learning publishers.
2. Calculus and Analytical Geometry, 6<sup>th</sup> edition. Swokowski, Olinick and Pence.1994. Thomson Learning EMEA, Ltd.
3. Multivariable Calculus, 5<sup>th</sup> edition Howard, A. Albert, H. 1995, John Wiley.

Course Name:	
Week #	Week wise distribution
1.	Functions of Several Variables
2.	and Partial Differentiation.
3.	Multiple Integrals,
4.	Line and Surface Integrals.



5.	Green's and Stoke's Theorem.
6.	Fourier Series:
7.	periodic functions,
8.	Functions of any period $P=2L$ ,
9.	Even & odd functions,
10.	Half Range expansions,
11.	Fourier Transform;
12.	Fourier Transform;
13.	Fourier Transform;
14.	Laplace Transform,
15.	Z-Transform.
16.	Wrap up work and presentations



**Course Name: Differential Equations**

**Course Code: MATH-201**

**Credit Hours: 3**

**Prerequisites:** Calculus and Analytical Geometry

**Objective:** To explain how differential equations appear in real life and physical phenomena, and teach them the main three methods, namely analytic, geometric and numerical methods, for studying differential equations.

**Learning Outcomes:** At the completion of this course, students will be able to learn:

- Differential equations appear in real life
- Physical phenomena
- Teach them the main three methods, namely o Analytic, o Geometric and o Numerical methods, for studying differential equations.

**Course Outline:**

Ordinary Differential Equations of the First Order: Geometrical Considerations, Isoclines, Separable Equations, Equations Reducible to Separable Form, Exact Differential Equations, Integrating Factors, Linear First-Order Differential Equations, variation of Parameters. Ordinary Linear Differential Equations; Homogeneous Linear Equations of the Second Order, Homogeneous Second-Order Equations with Constant Coefficients, General Solution, Real Roots, Complex Roots, Double Root of the Characteristic Equation, Differential Operators, Cauchy Equation, Homogeneous Linear Equations of Arbitrary Order, Homogeneous Linear Equations of Arbitrary Order with Constant Coefficients, Non-homogeneous Linear Equations. Modelling of Electrical Circuits. Systems of Differential Equations. Series Solutions of Differential Equations. Partial Differential Equations: Method of Separation of variables, wave, Heat & Laplace equations and their solutions by Fourier series method.

**Reference Materials:**

1. Advanced Engineering Mathematics Michael, G.1996, Prentice Hall Publishers.
2. Advanced Engineering Mathematics, 7<sup>th</sup> edition, Erwin, K. 1993, John Wiley & Sons Inc.
3. A First Course in Differential Equation Zill. Prindle. Weber. Schmidt.1996. Brooks/Cole Publishing.



4. Differential Equations with Boundary-Value Problems, Dennis. G. Zill, Michael, R. Cullen. 1996, Brooks/Cole Publishing
5. Elementary Differential Equations with Applications C. H. Edwards. David, E. 1993. Penney, Prentice Hall.

***Computer Science - Elective Courses and University Elective Courses***



**Course Name: Principles of Accounting**

**Course Code:** MGT-151

**Credit Hours:** 3

**Prerequisites:** None

**Objectives:** This is a basic course in financial accounting that covers the fundamental concepts and techniques of introductory accounting. Its focus includes a study of balance sheet accounting and the preparation and analysis of financial statements.

**Learning Outcome:**

- Apply basic computational techniques to solve quantitative financial accounting problems.
- Draw from financial information to construct a debit/credit transaction in good form
- Demonstrate knowledge of the business accounting cycle for the corporate form of business
- Identify and describe terms associated with financial accounting
- Prepare and interpret a multiple-step income statement, retained earnings statement, and classified balance sheet for a merchandising firm organized as a corporation
- Demonstrate knowledge of accounting for short-term liquid assets, long-term assets, current liabilities, long term liabilities and stock holders' equity.

**Course Outline:**

Introduction to Accounting, Accounting Principles, Book Keeping, Basics of Financial Statements, Adjustments to Financial Statements, The Cash Book, Bank Reconciliation, Control Accounts, Statement of Cash Flows, Financial Activities, Property, Plant and Equipment (PPE), Accounting Errors, Accounting for Partnerships, Balance Sheet.

**Reference Materials:**

1. Fundamental Accounting Principles with Connect Plus by John Wild, Ken Shaw, and Barbara Chiappetta, McGraw-Hill/Irwin; 20th Edition December 27, 2010). ISBN-10: 0077505980
2. Financial & Managerial Accounting by Jan Williams, Sue Haka, Mark Bettner and Joseph Carcello, McGraw-Hill/Irwin; 16th Edition 2011). ISBN - 10: 0078111048
3. Principles of Managerial Finance by Lawrence J. Gitman and Chad J. Zutter, Prentice Hall; 13th Edition (2011). ISBN-10: 0136119468
4. Fundamentals of Financial Management by J. Van Horne and John M Wachowicz, Prentice Hall; 13th Edition (2008). ISBN-10: 0273713639.



Course Name:	
Week #	Week wise distribution
1.	Introduction to Accounting,
2.	Accounting Principles,
3.	Book Keeping, Basics of Financial Statements,
4.	Adjustments to Financial Statements,
5.	The Cash Book, Bank Reconciliation,
6.	Control Accounts, Statement of Cash Flows,
7.	Financial Activities,
8.	Property,
9.	Plant and Equipment PPE), Accounting Errors, Accounting for Partnerships, Balance Sheet.
10.	Accounting Errors,
11.	Accounting for Partnerships,
12.	Accounting for Partnerships,
13.	Accounting for Partnerships,
14.	Balance Sheet.
15.	Balance Sheet.
16.	Wrap up work and presentations



**Course Name: Principles of Philosophy**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives:** Students will obtain knowledge of the main philosophical terms and categories and the ways of philosophical thinking for better interpretation of recognized reality.

**Learning Outcome:**

- Be able to read philosophy; that is, to identify the thesis of a piece of philosophical writing and the arguments or evidence adduced in support of that thesis.
- Be able to write philosophy; that is, to present a claim in clear terms and to defend it in a logically coherent manner.
- Be able to reconstruct and debate some foundational issues in the Western philosophical tradition.

**Course Outline:**

The Nature of Philosophy, Philosophical Theories: History and Back Ground, Realism and Idealism, Monism and Dualism, Rationalism, Empiricism, Criticism, and Empiricism, The Start of Modern Philosophy, Perception and Reality, Knowledge, Belief and Logic. Space, Time, Causality and Substance, Mind & Body. Knowledge, Language. Science, Morality, Politics, Law, Metaphysics.

**Reference Materials:**

1. An Introduction to Philosophy by Jon Nuttall, Polity; 1st Edition July 29, 2002). ISBN-10: 0745616631
2. An Introduction To Philosophy by George Stuart Fullerton, Create Space Independent Publishing Platform (July 18, 2011). ISBN-10: 1463688881
3. Philosophy: An Introduction to the Art of Wondering by James L. Christian, Wadsworth Publishing; 11th Edition January 26, 2011). ISBN -10: 1111298084
4. Pleasures of Philosophy by Durant, Touchstone; Revised Edition December 31, 1999. ISBN -13: 978-0671581107
5. Philosophy Basics: A Jargon-Free Guide for Beginners by Doug Erlandson, Doug Erlandson (September 15, 2011). ASIN: B005NJRTUW

Course Name:	
Week #	Week wise distribution
1.	The Nature of Philosophy,



2.	The Nature of Philosophy,
3.	Philosophical Theories:
4.	History and Back Ground,
5.	Realism and Idealism,
6.	Monism and Dualism,
7.	Rationalism,
8.	Empiricism,
9.	Criticism, and
10.	Empiricism
11.	, The Start of Modern Philosophy,
12.	Perception and Reality,
13.	Knowledge, Belief and Logic. Space, Time,
14.	Causality and Substance, Mind & Body. Knowledge, Language.
15.	Science, Morality, Politics, Law, Metaphysics.
16.	Wrap up work and presentations



**Course Name: Principles of Psychology**

**Course Code: PSY-401**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives:** Students acquire the knowledge and skills to enable them to take up professional activity as a psychologist. In addition to academic psychological research, students engage in diagnostic, advisory, evaluative and psychotherapeutic tasks in the health and social care sectors, education, administration, business and industry. Students acquire a scientifically based academic aptitude for choosing or independently developing theories and methodologies for analysis, verification and assessment of psychological activities. They acquire the basis of a vocational profession and an identity as a psychologist.

**Learning Outcome:**

- Describe key concepts, principles, and overarching themes in psychology.
- Describe applications of psychology.
- Use scientific reasoning to interpret psychological phenomena.
- Demonstrate effective writing for different purposes

**Course Outline:**

Basics concepts of Psychology and Research Methods. Brain and Behavior. Human Development. Sensation and Perception. States of Consciousness. Conditioning and Learning. Memory. Cognition, Language, Creativity, and Intelligence. Motivation and Emotion. Sex, Gender, and Sexuality. Personality. Health, Stress, and Coping. Social Behaviour

**Reference Materials:**

1. Psychology: Modules for Active Learning by Dennis Coon and John O. Mitterer, Wadsworth Publishing; 12th Edition January 1, 2011. ISBN -10: 1111342849
2. Introduction to Psychology by James W. Kalat, Wadsworth Publishing; 9th Edition (January 1, 2010). ISBN-10: 0495810762
3. Introduction to Psychology by Rod Plotnik and Haig Kouyoumdjian, Wadsworth Publishing; 9th Edition February 25, 2010. ISBN -10: 0495903442
4. Psychology by David G. Myers, Worth Publishers; 9th Edition January 10, 2009). ISBN10: 1429215976



Course Name:	
Week #	Week wise distribution
1.	Basics concepts of Psychology and
2.	Research Methods.
3.	Research Methods.
4.	Brain and Behavior.
5.	Human Development.
6.	Sensation and Perception..
7.	States of Consciousness
8.	Conditioning and Learning. Memory.
9.	Cognition, Language,
10.	Creativity, and Intelligence.
11.	Motivation and Emotion.
12.	Sex, Gender, and Sexuality.
13.	Personality.
14.	Health, Stress, and Coping.
15.	Social Behaviour
16.	Wrap up work and presentations



**Course Name:** Network Security

**Course Code:** CS 807

**Credit Hours:** 3

**Prerequisites:** System and Network Administration

**Objective:** To explain the policies adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network.

**Learning Outcome:**

- Define information security and outline its major components.
- Identify the major types of threats to information security and the associated attacks.
- Develop strategies to protect organization information assets from common attacks.
- Understand how security policies, standards and practices are developed.
- Understand the role of management in enforcing security policies, standards and practices.
- Identify the major techniques, approaches and tools used to discover network and system vulnerabilities.
- Apply foot printing, scanning, enumeration and similar techniques to discover network and system vulnerabilities.

**Course Outline:**

Security Concepts, Such as Confidentiality, Integrity, Authenticity, Availability etc. Symmetric and Asymmetric Cryptography and Their Uses; Key Distribution and Digital Signatures; Discretionary and Mandatory Access Control Policies for Confidentiality and Integrity. Communication Protocols for Authentication, Confidentiality and Message Integrity. Network Security; System Security, Intrusion Detection and Malicious Code. Security Models and Security Evaluation. Administration of Security. Legal Aspects of Computer Security.

**Reference Materials:**

1. Security in Computing by Charles P. Pfleeger and Shari Lawrence Pfleeger, Prentice Hall; 4th Edition (2006. ISBN -10: 0132390779
2. Network Security Fundamentals by Gert DeLaet and Gert Schauwers, Cisco Press; 1st Edition (September 18, 2004). ISBN-10: 1587051672
3. Network Security Bible by Eric Cole, Wiley; 2nd Edition (September 8, 2009). ISBN10: 0470502495



4. Network Security Essentials: Applications and Standards by William Stallings, Prentice Hall; 4th Edition (March 22, 2010). ISBN-10: 0136108059

Course Name:	
Week #	Week wise distribution
1.	Security Concepts,
2.	Security Concepts,
3.	Security Concepts,
4.	Confidentiality,
5.	Integrity,
6.	Authenticity, Availability etc.
7.	Symmetric and
8.	Asymmetric Cryptography and;
9.	Their Uses; Key Distribution and Digital Signatures
10.	Discretionary and Mandatory Access Control Policies for Confidentiality and Integrity.
11.	Communication Protocols for Authentication, Confidentiality and Message Integrity.
12.	Network Security; System Security, Intrusion Detection and Malicious Code.
13.	Security Models and
14.	Security Evaluation. Administration of Security.
15.	Legal Aspects of Computer Security.
16.	Wrap up work and presentations



## **Course Name: Cloud Computing**

**Course Code:** IT-455

**Credit Hours:** 3

**Prerequisites:** Internet Architecture and Protocols

**Objectives:** After completing this seminar, participants will be able to: Discuss, with confidence, what is cloud computing and what are key security and control considerations within cloud computing environments. Identify various cloud services. Assess cloud characteristics and service attributes, for compliance with enterprise objectives. Explain the four primary cloud category “types”. Evaluate various cloud delivery models.

**Learning Outcome:** Students will be able to:

- Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.
- Apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency and cost.
- Discuss system virtualization and outline its role in enabling the cloud computing system model.
- Illustrate the fundamental concepts of cloud storage and demonstrate their use in storage systems such as Amazon S3 and HDFS.
- Analyze various cloud programming models and apply them to solve problems on the cloud.

## **Course Outline:**

Overview of Distributed Computing, Emergence of Cloud Computing, Global Nature of the Cloud, Cloud-Based Service Offerings, Grid Computing, Reliability of Cloud Model, Benefits of Cloud Model, Legal Issues, Key Characteristics of Cloud Computing, Challenges for the Cloud. The Evolution of Cloud Computing. Web Services Delivered from the Cloud: Communication-as-a-Service CaaS), Infrastructure -as-a-Service, Monitoring-as-a-Service MaaS), Platform -as-a-Service PaaS), Software -as-a -Service

SaaS), Infrastructure as a Service IaaS), Platform as a Service PaaS), Software as a Service SaaS). Building Cloud Networks. Virtualization. Federation, Presence, Identity, and Privacy in the Cloud. Security in the Cloud. Common Standards in Cloud Computing. End-User Access to Cloud Computing. Mobile Internet Devices and the Cloud.

## **Reference Materials:**



1. Cloud Computing Implementation, Management, and Security by John W. Rittinghouse and James F. Ransome, Taylor & Francis Group, LLC 2010). ISBN 978 -1-4398-0680-7.
2. Cloud Computing Explained: Implementation Handbook for Enterprises by John Rhoton, Recursive Press (2009). ISBN-10: 0956355609.
3. Cloud Computing Bible by Barrie Sosinsky, Wiley; 1st Edition 2011). ISBN-10: 0470903562.

Course Name:	
Week #	Week wise distribution
1.	Overview of Distributed Computing, Emergence of Cloud Computing, Global Nature of the Cloud,
2.	Cloud-Based Service Offerings, Grid Computing, Reliability of Cloud Model, Benefits of Cloud Model,
3.	Legal Issues, Key Characteristics of Cloud Computing, Challenges for the Cloud. The Evolution of Cloud Computing.
4.	Web Services Delivered from the Cloud: Communication-as-a- Service CaaS), Infrastructure -as-a-Service,
5.	Monitoring-as-a-Service MaaS), Platform -as-a-Service PaaS), Software -as-a -Service SaaS),
6.	Infrastructure as a Service IaaS),
7.	Platform as a Service PaaS), Software as a Service SaaS). Building Cloud Networks.
8.	Virtualization. Federation,
9.	Presence,
10.	Identity, Cloud.
11.	and Privacy in the
12.	Security in the Cloud.
13.	Common Standards in Cloud Computing.
14.	End-User Access to Cloud Computing.
15.	Mobile Internet Devices and the Cloud.
16.	Wrap up work and presentations



**Course Name:** Object-Oriented Analysis and Design

**Credit Hours:** 3

**Prerequisites:** Programming Fundamentals

**Objectives:** After the course, students should be able, To use an object-oriented method for analysis and design, To know techniques aimed to achieve the objective and expected results of a systems development process, To know different types of prototyping, To know how to use UML for notation

**Learning Outcome:** The students should be able:

- Be able to analyse information systems in real-world settings and to conduct methods such as interviews and observations
- Have a general understanding of a variety of approaches and perspectives of systems development, and to evaluate other IS development methods and techniques

**Course Outline:**

Principles of Object Technology. OOP Review. Principles of Modeling. OOA&D Overview. OO Development Process. Requirements Engineering, Analysis, and Specification: Requirements Engineering, Use Cases, Prototyping, Class Models. Interaction Diagrams.

Verification and Validation. Architectural and Detailed Design. Class Diagrams. Interaction Diagrams. State Machines and Diagrams. Implementation, Package Diagrams. Activity Diagrams. OO Patterns, Verification and Validation. Note: Students may also be introduced to Object Diagram, Component Diagram, Package Diagram, Deployment Diagram, Network Diagram.

**Reference Materials:**

1. Applying UML and patterns: An introduction to Object-Oriented Analysis and Design and Iterative Development by Craig Larman, Prentice Hall; 3rd Edition (October 30, 2004). ISBN-10: 0131489062
2. Using UML: Software Engineering with Objects and Components by Perdita Stevens, Addison-Wesley; 2nd Edition February 13, 2006). ISBN -10:0321269675
3. Fundamental of Object-Oriented Design in UML by Meiler Page-Jones, Addison Wesley, 2000. ISBN: 020169946X.
4. The Unified Modeling Language User Guide by G. Booch, J. Rumbaugh and I. Jakobson, Addison-Wesley Professional; 2nd Edition 2005). ISBN - 10:0321267974.



5. The Unified Modeling Language Reference Manual by James Rumbaugh, Ivar Jacobson and Grady Booch, Addison-Wesley Professional; 2nd Edition (2004. ISBN -10: 032171895X.

Course Name:	
Week #	Week wise distribution
1.	Principles of Object Technology. OOP Review. Principles of Modeling. OOA&D Overview.
2.	OO Development Process. Requirements Engineering, Analysis, and Specification:
3.	Requirements Engineering, Use Cases,
4.	Prototyping, Class Models.
5.	Interaction Diagrams.
6.	Verification and Validation
7.	Architectural and Detailed Design
8.	Class Diagrams.
9.	Interaction Diagrams.
10.	State Machines and Diagrams. Implementation,
11.	Package Diagrams. Activity Diagrams. OO Patterns,
12.	Verification and Validation. Note: Students may also be introduced to Object Diagram, Component Diagram,
13.	Package Diagram,
14.	Deployment Diagram,
15.	Network Diagram.
16.	Wrap up work and presentations



## **Course Name: Database Administration and management**

Course Code: CS-463

**Credit Hours:** 3

**Prerequisites:** Database Systems

**Objectives:** At the completion of this course, students should be able to do the following: Construct simple and moderately advanced database queries using Structured Query Language (SQL). Understand and successfully apply logical database design principles, including E-R diagrams and database normalization. Design and implement a small database project using Microsoft Access. Understand the concept of a database transaction and related database facilities, including concurrency control, journaling, backup and recovery, and data object locking and protocols. describe and discuss selected advanced database topics, such as distributed database systems and the data warehouse.

### **Learning Outcome:**

- ☐ Understand the role of a database management system in an organization.
- Understand basic database concepts, including the structure and operation of the relational data model.
- Understand the role of the database administrator.

### **Course Outline:**

Installation of DBMS; SQL\* Plus; DBA Tools. DBMS Physical Structure & Architectural Components: Server, Instance, SGA, Shared Pool, Library Cache, Data Dictionary Cache, Large Pool, Processes. Startup and Shutdown Database. Managing Instances. Managing Files. Creating Database and Data dictionary. Managing Tablespaces. Operations with Tablespaces. Data File Management, Segments, Block. Managing Undo Data, Undo Data Statistics: Managing Tables and Users. Indexes Management, Maintaining Data Integrity, Constraints. Managing Privileges. Server Side Configuration. Client Side Configuration.

Usage and Configuration of Oracle Shared Server. Backup and Recovery. Sizing Shared Pool, Sizing Buffer Cache, I/O Issues. Tuning Rollback Segments. Tuning Shared Servers, Types of Locks, Block Efficiency, Storage hierarchy, Avoiding Dynamic allocation, Statistics, PCTFREE and PCTUSED, Monitoring Index Usage.

### **Reference Materials:**



1.Database Administration: The Complete Guide to DBA Practices and Procedures by Craig S. Mullins, Addison-Wesley Professional; 2nd Edition October 21, 2012). ISBN -10: 0321822943

2.Database Systems: A Practical Approach to Design, Implementation and Management by Thomas M. Connolly and Carolyn E. Begg, Addison-Wesley; 5th Edition (2009). ISBN-10: 0321523067

Course Name:	
Week #	Week wise distribution
1.	Installation of DBMS; SQL* Plus; DBA Tools. DBMS Physical Structure & Architectural Components:
2.	Server, Instance, SGA, Shared Pool, Library Cache, Data Dictionary Cache, Large Pool,
3.	Processes. Startup and Shutdown Database. Managing Instances. Managing Files.
4.	Creating Database and Data dictionary.
5.	Managing Tablespaces.
6.	Operations with Tablespaces.
7.	Data File Management,
8.	Segments, Block
9.	Managing Undo Data, Undo Data Statistics:
10.	Managing Tables and Users. Indexes Management,
11.	Maintaining Data Integrity,
12.	Constraints.
13.	Managing Privileges.
14.	Server Side Configuration.
15.	Client Side Configuration.
16.	Wrap up work and presentations



## **Course Name: Computer Game Development**

**Credit Hours:** 3

**Prerequisites:** Data Structures & Algorithms

**Objectives:** at the completion of this course, you will be able to: Discuss the history of electronic game development. Distinguish between the different game platforms and player modes. Distinguish between the different game goals and genres. Discuss various aspects of gameplay that can be used to design game interaction. Discuss the design and use of levels. Evaluate the game industry and market. Discuss the future of game design.

### **Learning Outcome:**

- Define elements related to game strategy, theory, and gameplay.
- Apply story and character development to games. Discuss the use of the interface for game design. Use audio to enrich the game atmosphere.
- Identify the distinct roles and responsibilities of game development team members.
- Discuss the production and management of the game design process.
- Analyze games. Use game design software.
- Develop game design documentation. Design games.

### **Course Outline:**

Introduction to Game Development, Platform and Player Modes, What Is The Framework? Goals And Genres? What Are The Possibilities? Player Elements, Player Motivation, Geographic, Psychographics. Demographics, Gender, Generation, Rating, Applying Player Market to Platform. Story and Character Development: Classic Charters, Traditional Story Structure, Story Element. Plot, Game Story Devices, Game Characters. Character Development Element, Point-of-view, Visual Character Development, Verbal Character Development, Movement. Visual Character Development, Verbal Character Development, Movements, Character Description, Game Storytelling and Documentation. Gameplay: Rules to Play, Interactivity Modes, Game theory, Challenges, Balance. Levels: Level Design, Structure, Time, Space. Interface: Playe-Centerd Design, Interface & Game Feature, Interface Types, Usability. Audio: Importance of Game Audio, Sound Effect, Voiceover, Music. Company Role, Team Roles, Tools, Business Side of Game Development. Production and Management, Development Phases, Game Documentation.

### **Reference Materials:**



1. Game Development Essentials by Jeannie Novak, Delmar Cengage Learning; 3rd Edition (August 17, 2011). ISBN-10: 1111307652
2. Game Development Essentials: An Introduction by Jeannie Novak, Delmar Cengage Learning; 3rd Edition (2011). ISBN-10: 1111307652
3. Game Development Essentials: Mobile Game Development by Kimberly Unger and Jeannie Novak, Delmar Cengage Learning; 1st Edition (2011). ISBN-10: 1418052655
4. Game Development Essentials: Game Interface Design by Kevin Saunders and Jeannie Novak, Delmar Cengage Learning; 2nd Edition (2012). ISBN-10: 1111642885

Course Name:	
Week #	Week wise distribution
1.	Introduction to Game Development, Platform and Player Modes, What Is The Framework? Goals And Genres? What Are The Possibilities?
2.	Player Elements, Player Motivation, Geographic, Psychographics. Demographics, Gender, Generation, Rating, Applying Player Market to Platform. Development, Verbal Character Development, Movements, Character Description, Game Storytelling and Documentation.
3.	Story and Character Development: Classic Charters, Traditional Story Structure, Story Element. Plot, Game Story Devices, Game Characters. Character Development Element, Point-of-view, Visual Character Development, Verbal Character Development, Movement. Visual Character
4.	Gameplay: Rules to Play, Interactivity Modes, Game theory,
5.	Challenges, Balance. Levels: Level Design, Structure,
6.	Time, Space. Interface:
7.	Playe-Centerd Design, Interface & Game:.,
8.	Feature, Interface Types, Usability. Audio
9.	Importance of Game Audio, Sound Effect, Voiceover
10.	Music. Company Role,.
11.	Team Roles,



---

12.	Tools, Business Side of Game Development.
13.	Production and Management,
14.	Development Phases,
15.	Game Documentation
16.	Wrap up work and presentations



**Course Name: Mobile Computing**

**Course Code: CS-359**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives:** This course introduces the basic concepts and principles in mobile computing. This includes the major techniques involved, and networks & systems issues for the design and implementation of mobile computing systems and applications.

**Learning Outcome:**

To understand the key components and technologies involved and to gain hands-on experiences in building mobile applications.

**Course Outline:**

Introduction to Mobile Computing, Architecture of Mobile Software Applications, Mobile Development Frameworks and Tools. Creating Consumable Web Services for Mobile Devices. Memory Management. Mobile Applications. Mobile User-Interface Design. Dynamic Linking. Concurrency. Managing Resources. Introduction to Mobile Application Development with Android. Introduction to Mobile Application Development with IOS. Introduction to Mobile Application Development with Windows Phone. Introduction to Mobile Application Development with Blackberry.

**Reference Materials:**

1. Programming Mobile Devices: An Introduction for Practitioners by Tommi Mikkonen, Wiley; 1st Edition (March 19, 2007). ISBN-10: 0470057386.
2. Professional Mobile Application Development by Jeff McWherter & Scott Gowell, Wrox; 1st Edition (September 4, 2012). ISBN-10: 1118203909
3. Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML by Reza B'Far and Roy T. Fielding, Cambridge University Press (2004). ISBN-10: 0521817331.
4. Fundamentals of Mobile and Pervasive Computing by Frank Adelstein, Sandeep KS Gupta, Golden Richard III and Loren Schwiebert, McGraw-Hill Professional; 1st Edition (2004). ISBN-10: 0071412379.

**Course Name:**



Week #	Week wise distribution
1.	Introduction to Mobile Computing,
2.	Architecture of Mobile Software Applications,
3.	Mobile Development Frameworks and Tools
4.	Mobile Development Frameworks and Tools
5.	Mobile Development Frameworks and Tools
6.	Mobile Development Frameworks and Tools.
7.	Creating Consumable Web Services for Mobile Devices.
8.	Memory Management.
9.	Mobile Applications.
10.	Mobile User-Interface Design.
11.	Dynamic Linking. Concurrency. Managing Resources.
12.	Introduction to Mobile Application Development with Android.
13.	Introduction to Mobile Application Development with IOS.
14.	Introduction to Mobile Application Development with Windows Phone.
15.	Introduction to Mobile Application Development with Blackberry.
16.	Wrap up work and presentations



**Course Name: E-Commerce**

**Course Code: IT-458**

**Credit Hours: 3**

**Prerequisites: Web Technologies**

**Objectives:** Develop, deploy, and maintain electronic commerce (e-commerce) applications.

**Learning Outcome:**

- Understand the process of setting up an interactive web site, displaying product catalogue, deploying
- shopping carts, handling credit card transaction
- Identify e-business models.
- Describe issues of concern in the design and development of an e-commerce.
- Discuss the techniques and technologies used to process online payments.
- Understand the process of maintaining security on the E-commerce site.
- Have knowledge in XML technology related to Business-to-Business E-commerce.
- Discuss the issues facing businesses that are considering worldwide marketing of their products and services.
- Discuss how the "digital divide" is impacting on our society and impact of the Internet on education.

**Course Outline:**

An overview of e-Commerce & Models, Planning an e-Commerce Framework, Managing Products and Categories, Product Variations and User Uploads, Enhancing the User Experience, The Shopping Basket, The Checkout and Order Process, Shipping and Tax, Discounts, Vouchers, and Referrals, Checkout, Taking Payment for Orders, User Account Management, Administration: Dashboard, Managing Products and Categories, Managing Orders, Customers, Refunds, Voucher Codes, Shipping, Deploying, Security, and Maintenance, SEO.

**Reference Materials:**

1. PHP 5 E-commerce Development by Michael Peacock, Packt Publishing January 20, 2010). ISBN -10: 184719964X
2. Introduction to E-Commerce by Jeffrey F. Rayport, McGraw-Hill, 2nd Edition (2007. ISBN -10: 0071232664



3. E-Commerce by Kenneth Laudon and Carol Guercio Traver, Prentice Hall; 8th Edition (2011). ISBN-10: 0138018812
4. e-Business and e-Commerce How to Program by Harvey M. Deitel, Paul J. Deitel and Tem R. Nieto, Prentice Hall; 1st Edition 2000). ISBN -10: 013028419X
5. The Complete E-Commerce Book: Design, Build & Maintain a Successful Web-based Business by Janice Reynolds, Cmp Books 2000). ISBN -10: 157820061X

Course Name:	
Week #	Week wise distribution
1.	An overview of e-Commerce & Models,
2.	Planning an e-Commerce Framework,
3.	Managing Products and Categories,
4.	Product Variations and User Uploads,
5.	Enhancing the User Experience,
6.	The Shopping Basket, The Checkout and Order Process,
7.	Shipping and Tax, Discounts,
8.	Vouchers, and Referrals, Checkout,
9.	Taking Payment for Orders,
10.	User Account Management, Administration: Dashboard,
11.	Managing Products and Categories,
12.	Managing Orders, Customers, Refunds,
13.	Voucher Codes, Shipping, ,
14.	Deploying, Security
15.	and Maintenance, SEO.
16.	Wrap up work and presentations



## **Course Name: Object Oriented Software Engineering**

**Credit Hours: 4**

**Prerequisites: Object Oriented Programming**

### **Objectives:**

To learn and understand various O-O concepts along with their applicability contexts. Given a problem, identify domain objects, their properties and relationships among them. How to identify and model/represent domain constraints on the objects and (or) on their relationships. Develop design solutions for problems on various O-O concepts. To learn various modeling techniques to model different perspectives of object-oriented software design (UML). To learn software development life cycle for Object-Oriented solutions for Real-World Problems. To learn O-O design solutions for the recurring problems.

### **Outcomes:**

At the end of the course the student will learn various O-O concepts along with their applicability contexts. The students will also be able to learn and develop design solutions for problems on various O-O concepts and will learn various modeling techniques to model different perspectives of object-oriented software design (UML).

### **Course Outline:**

Introduction to Software Engineering, Modeling with UML, Project Organization and Communication, Requirements Elicitation, Analyses, System Design: Decomposing the System, System Design: Addressing Design Goals, Object Design: Reusing Pattern Solutions, Object Design: Specifying Interfaces, Mapping Models to Code, Testing, Rationale Management, Configuration Management, Project Management, Software Life Cycle, Methodologies: Putting It All Together.

### **Reference Materials:**

1. *Object-Oriented Software Engineering: Using UML, Patterns, and Java*, Bernd Bruegge, Allen H. Dutoit, Prentice Hall, 2010 ( or Latest Edition)
2. *Object-Oriented Software Construction*, Bertrand Meyer, 2nd Edition, Prentice Hall in 1997 (or Latest Edition)
3. *Formal Methods in Computing*, M. Ferenczi, and Andras Pataricza , Akademiai Kiado, 2005 ( or Latest Edition)



4. *Code Complete: A practical handbook of software construction*, Microsoft Press, 2004. (or Latest Edition)
5. *Software Engineering*, Ian Sommerville, 8th edition, Addison & Wesley. 2006 (or Latest Edition)

Course Name:	
Week #	Week wise distribution
1.	Introduction to Software Engineering,
2.	Modeling with UML
3.	Modeling with UML
4.	Organization and Communication
5.	Requirements Elicitation,
6.	Analyses, System Design:
7.	Decomposing the System, System Design:
8.	Addressing Design Goals, Object Design:
9.	Reusing Pattern Solutions,
10.	Object Design:
11.	Specifying Interfaces, Mapping Models to Code,
12.	Testing,
13.	Rationale Management,
14.	Configuration Management, Project Management, Software Life Cycle,
15.	Methodologies: Putting It All Together.
16.	Wrap up work and presentations



## **Course Name: Software Requirements and Specifications**

**Credit Hours:** 3

**Prerequisites:** Introduction to Software Engineering

**Objective:** To explain the major issues in contemporary software development and maintenance, as related to complex and critical software systems. Become more aware about the practices which typically apply in software development projects from cradle to grave. See software as a part of a larger system, and be aware of the principles of systems engineering as they are relevant to the engineering of software.

**Outcomes:** Students will learn, in depth, the various selected models, tools, notations and validation techniques for the analysis and specification of system requirements that will enable the students to apply these in subsequent projects and work experiences.

### **Course Outline:**

Definition of Requirements Engineering and role in system development, Fundamental concepts and activities of Requirements Engineering, Information elicitation techniques, Modeling scenarios. Fundamentals of goal-oriented Requirements Engineering, Modelling behavioural goals, Modelling quality goals, Goal modelling heuristics, Object modelling for Requirements Engineering, Object modelling notations, Object modelling heuristics, Identifying objects from goals, Modelling Use Cases and state machines, Deriving operational requirements from goals, Requirements Specification, Requirements verification and validation. Management of inconsistency and conflict, Techniques for requirements evaluation, selection and prioritization; Requirements management; Requirements traceability

### **Reference Materials:**

1. *Requirements Engineering: Processes and Techniques*, Gerald Kotonya and Sommerville, John-Wiley Sons, 1998 (or Latest Edition).
2. *Software Requirements*, Karl E. Wiegers, Microsoft Press, 2003(or Latest Edition).
3. *Software Requirements Specification*, David Tuffley, CreateSpace Independent Publishing Platform, 2010 (or Latest Edition).
4. *System Requirements Engineering*, Loucopoulos and Karakostas, McGraw-Hill, 1995 (or Latest Edition).



Course Name:	
Week #	Week wise distribution
1.	Definition of Requirements Engineering and role in system development,
2.	Fundamental concepts and activities of Requirements Engineering,
3.	Information elicitation techniques, Modeling scenarios.
4.	Fundamentals of goal-oriented Requirements Engineering, Modelling behavioural goals,
5.	Modelling quality goals,
6.	Goal modelling heuristics,
7.	Object modelling for Requirements Engineering,
8.	Object modelling notations, Object modelling heuristics,
9.	Identifying objects from goals, Modelling Use Cases and state machines, Deriving operational requirements from goals,
10.	Requirements Specification, Requirements verification and validation.
11.	Management of inconsistency and conflict,
12.	Techniques for requirements evaluation,
13.	selection and prioritization;
14.	Requirements management;
15.	Requirements traceability
16.	Wrap up work and presentations



## **Course Name: Software Design and Architecture**

**Course Code:** SE-251

**Credit Hours:** 3

**Prerequisites:** Software Engineering

**Objectives:** An in-depth look at software design. Continuation of the study of design patterns, frameworks, and architectures. Survey of current middleware architectures. Design of distributed systems using middleware. Component based design. Measurement theory and appropriate use of metrics in design. Designing for qualities such as performance, safety, security, reusability, reliability, etc. Measuring internal qualities and complexity of software. Evaluation and evolution of designs. Basics of software evolution, reengineering, and reverse engineering.

Upon completion of this course, students will have the ability to:

- Apply a wide variety of design patterns, frameworks, and architectures in designing a wide variety of software
- Design and implement software using several different middleware technologies
- Use sound quality metrics as objectives for designs, and then measure and assess designs to ensure the objectives have been met
- Modify designs using sound change control approaches

Use reverse engineering techniques to recapture the design of software

### **Outcomes:**

1. Argue the importance and role of software architecture in large-scale softwaresystems.
2. Design and motivate software architecture for large-scale software systems.
3. Recognise major software architectural styles, design patterns, and frameworks.
4. Describe a software architecture using various documentation approaches and architectural description languages.
5. Generate architectural alternatives for a problem and selection among them.
6. Use well-understood paradigms for designing new systems.
7. Identify and assess the quality attributes of a system at the architectural level.
8. Motivate the architectural concerns for designing and evaluating a system's architecture.
9. Discuss and evaluate the current trends and technologies such as model-driven and service-oriented architectures.
10. Evaluate the coming attractions in software architecture research and practice.



## Course Outline:

Introduction to the discipline of design, generic design processes, and design management; software product design, including analysis activities such as needs elicitation and documentation, requirements development activities such as requirements specification and validation, prototyping, and use case modelling; engineering design analysis, including conceptual modelling and both architectural and detailed design; survey of patterns in software design, including architectural styles and common mid-level design patterns.

## Reference Materials:

1. *Software Architecture and Design Illuminated*, Kai Qian, Xiang Fu, Lixin Tao, Chong-Wei Xu, Jorge L. Diaz-Herrera, Jones and Bartlett Publishers, 1st Edition, 2009 (or Latest Edition).
2. *Introduction to Software Engineering Design: Processes, Principles and Patterns with UML2*, Christopher Fox, Addison-Wesley Professional, 2006 (or Latest Edition).
3. *Software Engineering Design: Theory and Practice*, Carlos Otero, CRC Press, 2012 (or Latest Edition).
5. *Software Engineering Techniques: Design for Quality*, Krzysztof Sacha, Springer, 2006 (or Latest Edition).

Course Name:	
Week #	Week wise distribution
1.	Introduction to the discipline of design,
2.	generic design processes,
3.	and design management;
4.	software product design,
5.	including analysis activities such as needs elicitation and documentation
6.	requirements development activities such as requirements specification and validation
7.	prototyping, and use case modelling;
8.	engineering design analysis,
9.	engineering design analysis,



10.	including conceptual modelling and
11.	both architectural and
12.	detailed design;
13.	survey of patterns in software design,
14.	including architectural styles and
15.	common mid-level design patterns.
16.	Wrap up work and presentations



**Course Name: Software Verification and Validation**

**Credit Hours: 3**

**Prerequisites:** Software Engineering

**Objective:**

- Understand the concepts and theory related to software testing.
- Understand different testing techniques used in designing test plans, developing test suites, and evaluating test suite coverage
- Understand the relationship between black-box and white-box testing and know how to apply as appropriate.
- Learn to use automated testing tools in order to measure code coverage.
- Understand how software developers can integrate a testing framework into code development in order to incrementally develop and test code.

**Outcomes:**

After having completed this course, the students will be able to apply main stream testing techniques, understand their main strengths and weaknesses, and determine when they are appropriate. In addition, the students will have acquired basic knowledge of concepts related to safety and reliability analyses, fault tolerance, and defensive programming.

**Course Outline:**

Introduction to software quality assurance, The Quality Challenge, Quality Control v/s Quality Assurance, Quality Assurance in Software Projects (Phases), Principles and Practices, Quality Management, Quality Assurance and Standards, Quality Planning and Quality Control, Verification and Validation, Planning Verification and Validation, Critical System Validation, Reliability Validation, Safety Assurance, Security assessment, Inspections and reviews, Principles of software validation, Software verification, Planning for Software Quality Assurance, Software Quality Assurance (SQA) Plans, SQA-Organizational Level Initiatives, SQA Planning (Observations, Numbers, Results), Software Testing, Specification based test construction techniques, White-box and grey-box testing, Others comprehensive software testing techniques for SDLC, Control flow oriented test construction techniques, Data flow oriented test construction techniques, Clean-room approach to quality assurance, Product Quality and Process Quality, Standards for process quality and standards for product quality, Walkthroughs and Inspections, Structure, Checklist, Audits, Roles



and Responsibilities (Reviews, Inspections, etc), How to make Reviews and Inspections most effective.

## Reference Materials:

1. *Fundamentals of Software Testing*, Bernard Hom, Wiley, 2012, (or Latest Edition).
2. “Software Quality Assurance: Principles and Practice”, Nina S. Godbole, Alpha Science, 2004 (or Latest Edition).
3. *Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement*, Jeff Tian, John-Wiley & sons, 2005 (or Latest Edition).
4. *Software Testing in the Real World: Improving the Process*, Kit, Edward, Addison-Wesley, 1998 (or Latest Edition).
5. *Perfect Software: And other illusions about testing*, Gerald M. Weinberg, Dorest House, 2008 (or Latest Edition).

Course Name:	
Week #	Week wise distribution
1.	Introduction to software quality assurance, The Quality Challenge, Quality Control v/s Quality Assurance,
2.	Quality Assurance in Software Projects (Phases), Principles and Practices, Quality Management, Quality Assurance and Standards,
3.	Quality Planning and Quality Control, Verification and Validation,
4.	Planning Verification and Validation, Critical System Validation, Reliability Validation,
5.	Safety Assurance, Security assessment, Inspections and reviews, Principles of software validation, Software verification,
6.	Planning for Software Quality Assurance, Software Quality Assurance (SQA) Plans,
7.	SQA-Organizational Level Initiatives, SQA Planning (Observations, Numbers, Results), Software Testing, Specification based test construction techniques, White-box and grey-box testing,



8.	Others comprehensive software testing techniques for SDLC, Control flow oriented test construction techniques, Data flow oriented test construction techniques,
9.	Clean-room approach to quality assurance, Product Quality and Process Quality,
10.	Standards for process quality and
11.	standards for product quality,
12.	Walkthroughs and
13.	Inspections, Structure, Checklist, Audits,
14.	Roles and Responsibilities (Reviews, Inspections, etc),
15.	How to make Reviews and Inspections most effective.
16.	Wrap up work and presentations



## **Course Name: Software Project Management**

**Course Code:** SE-401

**Credit Hours:** 3

**Prerequisites:** (Intro. to) Software Engineering

**Objective:** The student will understand the requirements for the content of a project management plan. The student will be able to write a plan for a small project according to an established standard. The student will understand the role of the manager in each phase of the software development life cycle.

### **Outcomes:**

Examine the fundamentals of the software project and the factors involved in using a methodology in the context of project management. Explain the risks, issues, and critical success factors associated with technology projects and software projects in particular. Create a project plan, including scope definition, risk assessment, task breakdown, team selection, estimates, communication mechanisms and progress evaluation and reporting using an appropriate project lifecycle. Present and justify a fully documented project plan. Employ appropriate software for project planning, estimation, monitoring and control, communication and reporting.

### **Course Outline:**

Software Crisis and Software Engineering, Classic Mistakes, Overview of Project Management, PMI Process Groups, Software project Phases, Project charter, Statement of Work (SOW), Planning Phase: Development lifecycle models, matching lifecycles to projects, Project plans, Work Breakdown Structures (WBS), Estimation of effort and cost (Expert Judgment, FP and Use Case point methods), Scheduling: Project network diagram fundamentals, CPM, PERT, Gantt charts, Critical chain scheduling, Using MS-Project, Assigning Resources, Resource leveling, Team models, Managing conflict and motivating, Project Monitoring and Control: Status reporting, Project metrics, EVM, Communications Techniques, Risk management and Change control Project Recovery, Documentation, Cutover/Migration, Post Project Reviews, Closing.

### **Reference Materials:**

1. *Software Project Management*, Bob Hughes, Mike Cotterell, McGraw-Hill Higher Education, 5th Edition, 2009 (or Latest Edition).
2. *The Software Project Manager's Handbook: Principles that work at work*, Dwayne Phillips, IEEE Computer Society Press and Wiley Interscience, 2nd Edition, 2004 (or Latest Edition).



Course Name:	
Week #	Week wise distribution
1.	Software Crisis and Software Engineering, Classic Mistakes, Overview of Project Management,
2.	PMI Process Groups, Software project Phases,
3.	Project charter, Statement of Work (SOW),
4.	Planning Phase: Development lifecycle models
5.	, matching lifecycles to projects, Project plans,
6.	Work Breakdown Structures (WBS),
7.	Estimation of effort and cost (Expert Judgment,
8.	FP and Use Case point methods), Scheduling: Project network diagram fundamentals, CPM, PERT, Gantt charts, Critical chain scheduling,
9.	Using MS-Project, Assigning Resources, Resource leveling, Team models, Managing conflict and motivating,
10.	Project Monitoring and Control: Status reporting,
11.	Project metrics, EVM,
12.	Communications Techniques,
13.	Risk management and Change control Project Recovery,
14.	Documentation,
15.	Cutover/Migration, Post Project Reviews, Closing.
16.	Wrap up work and presentations



## **Course Name: Formal Methods in Software Engineering**

**Course Code:** SE-358

**Course Structure:**

**Lectures:** 3

**Credit Hours:** 3

**Prerequisites:** Discrete Structures

**Objective:** This course provides a hands-on introduction to formal methods for software engineering. The purpose of formal methods is to enable the construction of highly reliable software. Their foundation is the precise specification of run-time properties that a software system is expected to satisfy. Formal methods are concerned with specifications that are precise for being stated in languages endowed with a formal syntax, semantics, and theory. Formality helps the specification process in at least two ways:

It naturally leads to unambiguous, high-quality specifications, and it provides the bases for automated tool support.

As we will see, formal specification techniques allow for the construction of highly automated verification tools that help software developers analyze specifications and corresponding code, looking for errors in requirements, models, designs, and implementations.

### **Outcomes:**

Upon completion of this course, the student will be able to do the following:

1. Apply the concepts of standard mathematical logic to produce proofs or refutations of well-formed propositions or arguments phrased in English or in a variety of formal notations (first order logic, discrete mathematics or Hoare Logic).
2. Given a description of a regular language, either in English, as a regular expression or as a grammar, generate a finite state automaton that recognizes that language. Similarly, given a deterministic or nondeterministic automaton, give a description of the language which it accepts.
3. Given an inductive definition of a simple data structure, write a recursive definition of a given simple operation on data of that type. Given some such recursively defined operations, prove simple properties of these functions using the appropriate structural induction principle.
4. Prove simple programs correct using Hoare Logic and Separation Logic.



5. Prove correctness and termination of a simple program using the weakest precondition calculus. Design a Turing Machine which will accomplish simple tasks.

## Course Outline:

Introduction to formal methods, developing and acquiring formal methods, using and applying formal methods, a brief introduction to logic and set theory, Introduction to Hoare's Logic, logic and theorem proving, modelling software systems, sequential, concurrent and reactive systems, states, state spaces, transition systems, combining state spaces, fairness, partial order view, modelling formalism; Formal Specifications Linear temporal logic, automata on infinite words, specifications using Buchi-automata, completeness of specification; Automatic verification, state space verification, representing states, the automata framework, combining Buchi-automata, checking emptiness, translating LTL into automata, model checking examples, checking complexity of model checking, safety properties, state space explosion problem. Z-Specification, Structure and Schema.

## Reference Materials:

1. Software Reliability Methods, Doron A. Peled , 2001 Springer-Verlag
2. Logic in Computer Science Modelling and Reasoning about Systems 2nd Edition Michael Huth, Imperial College of Science, Technology and Medicine, London
3. Principles of Model Checking, Christel Baier and Joost-Pieter Katoen MIT Press, 2008.

Course Name:	
Week #	Week wise distribution
1.	Introduction to formal methods, developing and acquiring formal methods,
2.	using and applying formal methods, a brief introduction to logic and set theory, Introduction to Hoare's Logic, logic and theorem proving,
3.	modelling software systems, sequential, concurrent and reactive systems, states, state spaces, transition systems, combining state spaces, fairness,
4.	partial order view, modelling formalism; Formal Specifications Linear temporal logic,
5.	automata on infinite words, specifications using Buchi-automata, completeness of specification;



6.	Automatic verification, state space verification,
7.	representing states,
8.	the automata framework, combining
9.	Buchi-automata, checking emptiness,
10.	translating LTL into automata,
11.	checking examples,
12.	model checking complexity of model checking,
13.	safety properties, state space explosion problem.
14.	Z-Specification,
15.	Structure and Schema.
16.	Wrap up work and presentations



## Course Name: Physics II: Mechanics

**Prerequisites:** None

**Objective:** The course aims at students developing a good working understanding of the basic and fundamental principles of rock mechanics as applied to designing and stabilizing excavations in rock masses.

### Course Outline:

Measurement. Motion along a straight line. Vectors. Motion in 2 and 3 dimensions. Force and motion. Kinetic energy and work. Potential energy and conservation of energy. Center of mass and rotation. Center of mass and linear momentum. Torque and angular momentum. Equilibrium and elasticity. Gravitation. Fluids. Oscillations. Waves. First and second law of thermodynamics.

**Recommended Book:** 1. Halliday, Resnick and Walker, "Fundamental of Physics" (Latest Ed.)

Course Name:	
Week #	Week wise distribution
1.	Measurement. Motion along a straight line.
2.	Vectors. Motion in 2 and 3 dimensions.
3.	Force and motion. Kinetic energy and work.
4.	Potential energy and conservation of energy. Center of mass and rotation.
5.	Center of mass and linear momentum.
6.	Torque and angular momentum.
7.	Equilibrium and elasticity.
8.	Equilibrium and elasticity.
9.	Gravitation. Fluids.
10.	Gravitation. Fluids.
11.	Oscillations. Waves.
12.	Oscillations. Waves.
13.	First and second law of thermodynamics.
14.	First and second law of thermodynamics.
15.	First and second law of thermodynamics.
16.	Wrap up work and presentations



## **Course Name: Software Engineering Economics**

**Credit Hours:** 3

**Prerequisites:** (Intro. to) Software Engineering

**Objectives:** Determine how new software development technologies affect the economics and risks of software development. Understand and characterize how the paradigm shift affects or replaces our current methods of software cost, schedule and risk estimation. Identify best practices and lessons learned with Web-based developments. Identify acquisition and lifecycle risks

### **Outcomes:**

The student will be able to learn,

1. Understand and be able to apply the key software engineering economic fundamentals to real-world software economic issues.
2. Illustrate through example the key software life cycle economics, including product and process life cycles; portfolios; proposals; investment decisions; pricing and costing, and earned value management (EVM).
3. Apply the concepts of risk and uncertainty to real world software development projects, including goals; estimates; prioritization and decision making.
4. Perform best-practice economic analysis methods.
5. Relate and interpret the “good-enough” principle; friction-free economy; ecosystems and outsourcing

### **Course Outline:**

Programming aspects, economic aspects, human relations aspects, software trends: cost, social impact, the plurality of SE Means, The GOALS Approach to Software Engineering, The Software Work Breakdown Structure (WBS), Software Maintenance, introduction to COCOMO, definitions and assumptions, development effort and schedule, phase distribution, The Rayleigh Distribution, interpolation, basic software maintenance effort estimation. Performance Models, Optimal Performance, Sensitivity Analysis, Cost-Effectiveness Models.

### **Reference Materials:**

1. *Software Engineering Economics*, Boehm, Prentice Hall, 1981(or Latest Edition).
2. *Software Cost Estimation with COCOMO II*, Boehm et al., Prentice Hall, 2000 (or Latest Edition).
3. *Making the Software Business Case: Improvement*, Reifer, Don, Addison Wesley, 2001, (or Latest Edition).

Course Name:	
Week #	Week wise distribution



1.	Programming aspects, economic aspects, human relations aspects, software trends: cost, social impact, the plurality of SE Means,
2.	The GOALS Approach to Software Engineering,
3.	The GOALS Approach to Software Engineering,
4.	The Software Work
5.	Breakdown Structure (WBS),
6.	Software Maintenance, introduction to COCOMO,
7.	definitions and assumptions,
8.	development effort and schedule,
9.	phase distribution,
10.	The Rayleigh Distribution, interpolation,
11.	basic software maintenance effort estimation.
12.	Performance Models,
13.	Optimal Performance,
14.	Sensitivity Analysis,
15.	Cost-Effectiveness Models.
16.	Wrap up work and presentations



**Course Name: Data Mining**

**Course Code: CS-455**

**Credit Hours: 3**

**Prerequisites:** Introduction to Database Systems

**Objective:**

To demonstrate background knowledge about the process of data mining and knowledge discovery. Describe the methods involved in data mining, their scope and limitations. Apply practical knowledge on data mining and pattern discovery, and analysis skills

**Outcomes:**

By the end of the course, the student should

- Display a comprehensive understanding of different data mining tasks and the algorithms most appropriate for addressing them.
- Evaluate models/algorithms with respect to their accuracy.
- Demonstrate capacity to perform a self directed piece of practical work that requires the application of data mining techniques.
- Critique the results of a data mining exercise.
- Develop hypotheses based on the analysis of the results obtained and test them.
- Conceptualise a data mining solution to a practical problem.

**Course Outline:**

Concepts of Data mining, Data Preparation Techniques: outlier and missing data analysis, Data Reduction Techniques, learning methods in Data mining, Statistical Methods in Data Mining, Cluster Analysis, hierarchal, agglomerative and Naïve Bayesian methods, Decision Trees and Decision Rules, Association Rules, Other Soft Computing Approaches in Data Mining, Artificial Neural Networks, Fuzzy Logic and Fuzzy Set Theory, Genetic Algorithm, evolutionary algorithms.

**Reference Materials:**

1. Mehmed Kantatardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*, 2003, John Wiley and Sons. (latest edition)
2. Margaret H. Dunham and S. Sridhar, *Data Mining, Introductory and Advanced Topics*, 2006, Pearson Education, (latest edition)
3. David Hand, Heikki MANNILA and Padhraic Smyth, *Principles of Data Mining*, 2001, The MIT Press. (latest edition)



Course Name:	
Week #	Week wise distribution
1.	Concepts of Data mining, Data Preparation Techniques: outlier and missing data analysis,
2.	Data Reduction Techniques,
3.	learning methods in Data mining,
4.	Statistical Methods in Data Mining,
5.	Cluster Analysis, hierarchal,
6.	agglomerative and Naïve Bayesian methods,
7.	Decision Trees and Decision Rules,
8.	Association Rules,
9.	Other Soft Computing
10.	Approaches in Data Mining,
11.	Artificial
12.	Neural Networks,
13.	Fuzzy Logic and Fuzzy Set Theory,
14.	Genetic Algorithm,
15.	evolutionary algorithms.
16.	Wrap up work and presentations



**Course Name: Web Engineering**

**Course Code: CS-453**

**Credit Hours: 3**

**Prerequisites: Programming Fundamentals**

**Objective:** To introduce the methods and techniques used in Web-based system development. To provide students with conceptual and practical knowledge, and skills required to develop web applications and web services. To develop a web application using server side programming languages and components. To apply the web engineering methodologies for Web application development.

**Outcomes:** On successful completion of the course students will be able to:

1. Develop a web application using server side programming languages and components.
2. Apply the web engineering methodologies for Web application development
3. Develop a component based web solution and use UML diagrams to describe such a solution.
4. Identify and discuss the security risk of a Web application.

**Course Outline:**

Introduction, Web Basics, Servlets and Sessions, Java Server Pages, XML and the Web Database Access for Web Applications, Design Patterns for Web Applications, Security and Privacy, Performance and Scalability, Performance Analysis, WAP Protocol, Wireless application development, Android Programming.

**Reference Materials:**

1. *Web Engineering: A Practitioners' Approach*, Roger S. Pressman, McGraw Hill (2008) or Latest Edition
2. *Web Engineering: The Discipline of Systematic Development of Web Applications*, Gerti Kappel, Birgit Prýýll, Siegfried Reich and Werner Retschitzegger, McGraw-Hill, (2006) or Latest Edition
3. *Web Engineering*, Emilia Mendes and Nile Mosley, Springer Verlag, (2010) or Latest Edition.

Course Name:	
Week #	Week wise distribution
1.	Introduction, Web Basics,
2.	Servlets and Sessions,
3.	Java Server Pages,
4.	XML and the Web Database Access for Web Applications,



5.	Design Patterns for Web Applications,
6.	Security and Privacy,
7.	Performance and Scalability,
8.	Performance Analysis,
9.	WAP Protocol,
10.	Wireless application development,
11.	Wireless application development,
12.	Wireless application development,
13.	Android Programming.
14.	Android Programming.
15.	Android Programming.
16.	Wrap up work and presentations



## **Course Name: Modern Programming Language**

**Credit Hours: 3**

### **Prerequisites: Object Oriented Programming**

**Objective:** The course objective is to outline a concept of general JAVA features, tools and utilities, to teach students to develop applications for wide range of tasks, to give a basis for the further study of JAVA-technologies. Program helps to get the key trade.

**Outcomes:** Upon successful completion of this course, students should be able to:

- Analyze and explain the behavior of programs involving the fundamental program constructs
- Write short programs that use the fundamental program constructs, including standard conditional and iterative control structures
- Identify and correct syntax and logic errors in short programs
- Write short programs that use arrays or array lists
- Design and implement a class based on attributes and behaviors of objects
- Construct objects using a class and activate methods on them
- Use static and instance members of a class properly
- Identify and describe the properties of a variable such as its associated value, scope and lifetime
- Describe the parameter passing mechanisms in terms of formal parameters, actual parameters, non-object parameters and object parameters
- Write a graphics program that draws simple shapes
- Identify super- and subclasses in a class hierarchy
- Recognize and trace overridden and inherited methods in a class hierarchy
- Write javadoc comments for classes and methods
- Be able to use an integrated development environment and a debugger

### **Course Outline:**

Introduction to Java, Java Characteristics, Java Environment, JDK installation, Different ways of java programs execution, Java Data types. Introduction to control Structures, Selection Statements, Iteration Statements, Jump Statements, Introduction to Java Packages, Types of packages. Hierarchy, Why to use packages (Purpose), Creating and using packages, Default package, Assignment checking. Introduction to java interfaces, Methods and variable in interfaces, Interface implementation, Extended interfaces. Introduction to java threads, Multiple threading, Thread



interface & implementation, Extended threads. Introduction to Exception, Exception handling, Default exception handling, How to handle multiple exceptions, Why and when to use exception. Introduction to GUI, JOptionPane class, GUI Components, JFrame class, Window creation, JLabel class, JTextField class, JButton class, Event Handling, ActionListener class. GUI other components, Introduction to Applets, Life Cycle of Applets, Advantages & disadvantages of Applets, JApplet class, Font, Color and Graphics classes. Introduction to JDBC, JDBC API, JDBC steps, Connectivity model, ODBC driver (other drivers), JDBC steps with Explanation. Introduction to Servlets, Life Cycle of Servlets, Advantages & disadvantages of Servlets, Servlets class, Assignment Checking, Presentations.

### Reference Material:

- 1) *The Complete Reference, Java 2, 5th Edition*
- 2) *Web Enable Commercial Applications Development Using Java 2.0 by Ivan Bayross*

Course Name:	
Week #	Week wise distribution
1.	Introduction to Java, Java Characteristics, Java Environment, JDK installation, Different ways of java programs execution, Java Data types.
2.	Introduction to control Structures, Selection Statements, Iteration Statements, Jump Statements, Introduction to Java Packages,
3.	Types of packages. Hierarchy, Why to use packages (Purpose), Creating and using packages,
4.	Default package, Assignment checking. Introduction to java interfaces, Methods and variable in interfaces, Interface implementation,
5.	Extended interfaces. Introduction to java threads, Multiple threading, Thread interface & implementation, Extended threads. Introduction to Exception,
6.	Exception handling, Default exception handling, How to handle multiple exceptions,
7.	Why and when to use exception. Introduction to GUI, J Option Pane class,
8.	GUI Components, JFrame class, Window creation, JLabel class, JTextField class,
9.	JButton class, Event Handling, ActionListener class. GUI other components,



10.	Introduction to Applets, Life Cycle of Applets, Advantages & disadvantages of Applets,
11.	JApplet class, Font, Color and Graphics classes. Introduction to JDBC, JDBC API, JDBC steps,
12.	Connectivity model, ODBC driver (other drivers), JDBC steps with Explanation.
13.	Introduction to Servlets, Life Cycle of Servlets,
14.	Advantages & disadvantages of Servlets, Servlets class,
15.	Assignment Checking, Presentations
16.	Wrap up work and presentations



**Course Name: Visual Programming**

**Course Code: CS-401**

**Credit Hours: 3**

**Prerequisites: Object Oriented Programming**

**Objective:** Align the students with latest available technologies in software development.

Students should have strong programming base.

Students should have sound knowledge in Microsoft .Net technology

Students should be able to tackle any programming related problem dealing with .Net.

Students should be able to switch out any other version of .net deployed by Microsoft

**Outcomes:** The student will use Visual Basic.Net to build Windows applications using structured and object-based programming techniques. Students will be exposed to the following concepts and/or skills at an introductory concepts level:

- Analyze program requirements
- Design/develop programs with GUI interfaces
- Code programs and develop interface using Visual Basic .Net
- Perform tests, resolve defects and revise existing code

### **Course outline:**

Introduction, Visual Studio .NET Integrated Development Environment (IDE) Overview, Menu Bar and Toolbar, Visual Studio .NET Windows, Solution Explorer, Toolbox, Properties Window, Using Help, Simple Program: Displaying Text and an Image. Control structures (if/Else, Switch, continue, and break), Loops (for, while, do while, for each), arrays, methods, creating custom methods, build in methods. Objects and classes, inheritance, polymorphism, exception handling (try, catch, throw and throws), introduction to win- forms. Windows Forms, Event-Handling Model, Basic Event Handling, Control Properties and Layout, Labels, TextBoxes and Buttons, GroupBoxes and Panels, CheckBoxes and RadioButtons, PictureBoxes, Mouse Event Handling, Keyboard Event Handling, Menus, Tab Control, Multiple-Document-Interface (MDI) Windows, Visual Inheritance, User-Defined Controls, Introduction to multithreading, Thread States: Life Cycle of a Thread, Thread Priorities and Thread Scheduling. Introduction to files and streams, Data Hierarchy, Files and Streams, Classes File and, Writing Data Randomly to a Random-Access File, Reading Data



Sequentially from a Random-Access File. Connecting with database, executing queries, retrieving data, creating reports, ADO.net object model, Introduction to Asp.net and web Form. Introduction to XML.

Course <b>Visual Programming</b>	
Week #	Distribution
1.	Introduction, Visual Studio .NET Integrated Development Environment (IDE) Overview, Menu Bar and Toolbar, Visual Studio .NET Windows, Solution Explorer, Toolbox,
2.	Properties Window, Using Help, Simple Program: Displaying Text and an Image. Control structures (if/Else, Switch, continue, and break), Loops (for, while, do while, for each),
3.	arrays, methods, creating custom methods, build in methods. Objects and classes, inheritance, polymorphism, exception handling (try, catch, throw and throws), introduction to win- forms
4.	Windows Forms, Event-Handling Model, Basic Event Handling, Control Properties and Layout, Labels, TextBoxes and Buttons, GroupBoxes and Panels, CheckBoxes and RadioButtons, PictureBoxes
5.	Mouse Event Handling, Keyboard Event Handling, Menus, Tab Control, Multiple-Document-Interface (MDI) Windows, Visual Inheritance, User-Defined Controls
6.	Introduction to multithreading, Thread States: Life Cycle of a Thread, Thread Priorities and Thread Scheduling
7.	Introduction to files and streams, Data Hierarchy, Files and Streams, Classes File and, Writing Data Randomly to a Random-Access File
8.	Reading Data Sequentially from a Random-Access File
9.	Connecting with database, executing queries, retrieving data, creating
10.	reports, ADO.net object model
11.	Introduction to Asp.net
12.	and web Form
13.	Introduction to XML



14.	Introduction to XHTML
15.	Presentation
16.	Warp Up Your Work And Presentation



## Course Name: Parallel & Distributed Computing

Course Code: CS-403

Credit Hours: 3

Prerequisites: Operating Systems

**Objective:** This course covers abstractions and implementation techniques for the design of distributed systems. At the end of this course students will be familiar with the design and implementation issues of distributed systems. Studying the core ideas behind modern coordination and communication paradigms and distributed data structures. Introduce a variety of methodologies and approaches for reasoning about concurrent and distributed programs

**Outcomes:** To be able to:

- Distinguish the theoretical and conceptual foundations of distributed computing.
- Recognize the inherent difficulties that arise due to distributed-ness of computing resources.
- Recognize the feasibilities and the impossibilities in managing resources.
- Identify the problems in developing distributed applications.

### Course Outline:

Introduction to Parallel and Distributed Systems, Software Architectures: Threads and Shared memory, Processes and Message passing, Distributed Shared Memory (DSM), Distributed Shared Data (DSD). System Models, Networking and Internetworking, Communication Models and Abstractions (Message passing, stream-oriented communications, remote procedure calls, remote method invocation), Naming in Distributed Systems, Concurrency and Synchronization, Process Synchronization, Distributed Transaction and Concurrency Control, Distributed Data Replication, Security and Access Control, Overview of Web Services, Cloud Computing.

### Reference Materials:

1. *Distributed Systems: Principles and Paradigms*, Andrew S. Tanenbaum and Maarten van Steen. Prentice-Hall, 2002.
2. *Distributed Systems: Concepts and Design* by 4th edition, George Coulouris, Jean Dollimore and Tim Kindberg. Addison-Wesley, 2005
3. *Web Services: Principles and Technology*. Michael P. Papazoglou. Pearson Prentice Hall, 2007.

Course <b>Distributed Computing</b>	
Week #	Distribution
1.	Introduction to Parallel and Distributed Systems



2.	Software Architectures: Threads and Shared memory
3.	Processes and Message passing
4.	Distributed Shared Memory (DSM), Distributed Shared Data (DSD). System Models
5.	Networking and Internetworking, Communication Models and Abstractions (Message passing, stream-oriented communications
6.	remote procedure calls, remote method invocation), Naming in Distributed Systems
7.	Concurrency and Synchronization
8.	Process Synchronization
9.	Distributed Transaction
10.	Concurrency Control
11.	Distributed Data Replication
12.	Security and Access Control
13.	Overview of Web Services
14.	Cloud Computing.
15.	Presentation
16.	Presentation and Wrap Up Your Work



**Course Name: Information System Audit**

**Course Code:** IT-454

**Credit Hours:** 3

**Prerequisites:** None

**Objectives:** To provide basic concept of information system audit and control, policies and procedures as defined by ISACA. To review and evaluate or conduct IS audits of an organization.

**Outcomes:** Students will be able to learn,

- Provide a conceptual framework of internal controls in a computer environment
- Discuss the primary steps in conducting a risk assessment of an IT system
- Review the audit implications of recent technological changes
- Review the evaluation and testing procedures for General and Business Process Application Controls

**Course Outline:**

IS Audit charter, Policies, Procedures, Audit computer networks and communication, Auditing software development, Acquisition, Maintenance, Auditing IT infrastructure, Auditing Management and Organization, Business process re-engineering: IS audit proposal, report, evidence and follow-up, complaint to standard, Enterprise service agreement, IP pro count policies and process, Backup and procedures

**Reference Materials:**

1. *Auditing Information Systems*, Jack J. Champlain, John-Wiley & Sons, 2003 (or Latest Edition).
2. *Information Systems Control and Audit*, Ron Weber, Pearson, 2011 (or Latest Edition).
3. *CISA Review Manual*, Information System Audit and Control Association, 2004, [www.isaca.org](http://www.isaca.org).

Course <b>Information System Audit</b>	
Week #	Distribution
1.	IS Audit charter,
2.	Policies
3.	Procedures
4.	Audit computer networks and communication
5.	Auditing software development



6.	Acquisition
7.	Maintenance
8.	Auditing IT infrastructure
9.	Auditing Management and Organization, Business
10.	process re-engineering: IS audit proposal,
11.	report
12.	evidence and follow-up,
13.	complaint to standard, Enterprise service agreement, IP pro count policies
14.	and process, Backup and procedures
15.	Presentation
16.	Wrap Up Your Work and Presentation



## **Course Name: Business Process Re-Engineering**

**Credit Hours: 3**

### **Prerequisites: Introduction to Software Engineering**

**Objective:** To explain the major issues in contemporary software development and maintenance, as related to complex and critical software systems. Become more aware about the practices which typically apply in software development projects from cradle to grave. See software as a part of a larger system, and be aware of the principles of systems engineering as they are relevant to the engineering of software.

**Outcomes:** Participants will experience and learn to:

1. Map and evaluate business process using Value Stream Mapping tool
2. Initiate reengineering project given by stakeholders or sponsors
3. Identify and select processes for reengineering
4. Draw Process Chart using the mapping tool/software
5. Identify value-added and non values-added activities within a process using the reengineering technique
6. Evaluate the process outcomes in term of cost, duration and service quality using the reengineering technique
7. Reengineer the processes to meet business objectives
8. Develop a reengineering project plan endorsed by stakeholders
9. Simulate the new processes for optimum results according to project plan
10. Structure the organization and jobs to support the new process using the reengineering technique
11. Introduce change management to support the new process
12. Conduct post reengineering review according to organizational procedures
13. Drive continuous improvement programs according to organizational procedures

### **Course Outline:**

Why Focus on Business Processes? Setting the Stage for Business Process; Organizing for Process Improvement; Flowcharting: Drawing a Process Picture; Understanding the Process Characteristics; Streamlining the Process; Measurements, Feedback, and Action; Process Qualification; Measurements, Feedback, and Action.

### **Reference Materials:**



1. *Business Process Improvement; The Breakthrough Strategy for Total Quality, Productivity, and Competitiveness*, H. J. Harrington,

Course <b>Business Process Re-Engineering</b>	
Week #	Distribution
1.	Intro To Business Processes
2.	Why Focus on Business Processes?
3.	Setting the Stage for BusinessProcess
4.	Organizing for Process Improvement;
5.	Flowcharting
6.	Drawing
7.	Process Picture
8.	Understanding the Process Characteristics
9.	Streamlining theProcess;
10.	Measurements
11.	Feedback, and Action
12.	Process Qualification
13.	Measurements
14.	Feedback, and Action.
15.	Presentation
16.	Presentation and Wrap Up Your Work



## **Course Name: Introduction to Soft Computing**

**Credit Hours: 3**

### **Prerequisites: Object Oriented Programming**

**Objectives:** The course provides an in-depth overview of the theoretical and the practical aspects of the soft computing paradigm. The main focus is on the theory and application of probabilistic graphical models (commonly known as Bayesian networks in the Artificial Intelligence community) and related topics, such as, knowledge elicitation issues, belief updating in singly and multiply connected networks, simulation schemes for belief updating, parameter and structure learning of Bayesian networks, and integration of time and uncertainty. Alternative models of uncertain reasoning (including belief function theory and fuzzy logic) and biologically inspired computational models (neural networks and evolutionary algorithms) are also presented.

**Outcomes:** To give students knowledge of soft computing theories fundamentals, i.e. of fundamentals of non-traditional technologies and approaches to solving hard real-world problems, namely of fundamentals of artificial neural networks, fuzzy sets and fuzzy logic and genetic algorithms.

### **Course Outline:**

Neural Networks, Classification Tree, Naïve Bayes, Applications of Predictive

Models, Probabilistic Reasoning using Bayesian Networks: Knowledge Acquisition, Belief Updating, Exact and Simulation-based Propagation Algorithms, Parameter and Structure Learning. Influence Nets, Dempster- Shafer Theory of Belief Functions, Fuzzy Logic.

### **Reference Materials:**

1. "Bayesian Artificial Intelligence", Kevin Korb, Ann Nicholson, CRC Press, 2010 (or Latest Edition).
2. "Artificial Intelligence Illuminated", Ben Coppin, Jones and Bartlett, 2004 (or Latest Edition).
3. "Soft Computing & Intelligent Systems Design: Theory, Tools and Applications", Fakhreddine O. Karray, Clarence W De Silva, Addison-Wesley, 1st. Edition, 2005 (or Latest Edition).

Course <b>Introduction to Soft Computing</b>	
Week #	Distribution
1.	Neural Networks
2.	Classification Tree



3.	Naïve Bayes
4.	Applications of Predictive
5.	Models, Probabilistic
6.	Reasoning using Bayesian Networks
7.	Knowledge Acquisition, Belief Updating
8.	Exact and Simulation-based Propagation Algorithms
9.	Parameter and Structure Learning
10.	Influence Nets
11.	Dempster- Shafter
12.	Dempster- Shafter Theory
13.	Dempster- Shafter Theory of Belief Functions
14.	Dempster- Shafter Theory of Belief Functions, Fuzzy Logic.
15.	Presentation
16.	Warp Up Your Work And Presentation



## Course Name: Real Time Systems

**Credit Hours:** 3

### Prerequisites: Operating Systems

**Objective:** The course provides an in-depth overview of the fundamental concepts and terminology of real-time systems, explain and address the fundamental problems of real-time systems, analyze real-time systems designs, design a real-time system (at least partially) and identify and assess the relevant literature and research trends of real-time systems.

**Outcomes:** Student will be able to learn,

1. Real-time scheduling and schedulability analysis
2. Formal specification and verification of timing constraints and properties
3. Design methods for real-time systems
4. Development and implementation of new techniques to advance the state-of-the-art real-time systems research

### Course Outlines:

Introduction to Real-Time and Embedded Systems: Reference Model, Hard versus soft real-time, Job Scheduling: Clock driven scheduling algorithms, Priority driven scheduling algorithms, Schedulers in commodity and real-time operating systems, Resource access control: Algorithms, Implementation, Real-time communication: On best-effort networks, enhanced quality of service.

### Reference Material:

1. *Real-Time Systems*, Jane W. S. Liu, Prentice Hall, 1st Edition, 2000 (or Latest Edition).
2. *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Hermann Kopetz, Springer; 2nd Edition, 2011, (or Latest Edition).
3. *Real-Time Systems Design and Analysis: Tools for the Practitioner*, Phillip A. Laplante, Seppo J. Ovaska, Wiley-IEEE Press; 4th Edition, 2011 (or Latest Edition).

Course Name: Real Time Systems	
Week #	Distribution
1.	Introduction to Real-Time
2.	Embedded Systems
3.	Reference Model
4.	Hard versus soft real-time



5.	Job Scheduling: Clock driven scheduling algorithms
6.	Priority driven scheduling algorithms
7.	Schedulers in commodity
8.	and real-time operating systems
9.	Resource access control
10.	Algorithms
11.	Implementation
12.	Real-time communication
13.	On best-effort networks
14.	enhanced quality of service.
15.	Presentation
16.	Warp Up Your Work and Presentation



## Course Name: Introduction to Complex Networks

**Credit Hours:** 3

**Prerequisites:** None

### Course Outline:

Fundamentals of networks, Mathematics of Graphs and Networks, Measures and Metrics Degree and eccentricity Centrality, Shortest path between Centrality, Clustering coefficient, Matching index, Large-scale Nature of Networks, Network Models (Erdos-Renyi random, Small-world , Scale-free network models, Calculation of basic measures in networks), Network Modeling tools overview (Pajek, Network Workbench, Gephi, Visone, Cytoscape, Centibin etc.), Evolution of Online Social networks: Facebook, Google+, Twitter, LinkedIn and Beyond, Social Network Analysis, Modeling Software Components as Agents in Networks

### Outcomes:

The student will acquire knowledge about the main mathematical properties characterizing a network, and he will receive an overview of the most recent and important applications of network models to real situations, in particular related to biology. Moreover, he will understand (and apply) the main algorithms for graph analysis and for implementing dynamical models embedded in networks of different topological structure.

### Reference Materials:

1. *Networks: An Introduction* by Mark Newman, Oxford University Press, 2010 (latest Ed.)
2. *Exploratory Social Network Analysis with Pajek (Structural Analysis in the Social Sciences)* by Wouter De Nooy, Andrej Mrvar and Vladimir Batagelj, Cambridge University Press, 2011 (latest Ed.)
3. *Six Degrees: The Science of a Connected Age*, Duncan J. Watts, Vintage, 2004 (latest Ed).

Course Name: <b>Introduction to Complex Networks</b>	
Week #	Distribution
1.	Fundamentals of networks,
2.	Mathematics of Graphs and Networks
3.	Measures and Metrics Degree and eccentricity Centrality
4.	Shortest path between Centrality



5.	Clustering coefficient
6.	Matching index, Large-scale Nature of Networks
7.	Network Models (Erdos-Renyi random, Small-world
8.	Scale-free network models, Calculation of basic measures in networks)
9.	, Network Modeling tools overview
10.	Pajek, Network Workbench, Gephi, Visone, Cytoscape, Centibin etc.)
11.	Network Modeling tools overview
12.	Network Modeling tools overview (Evolution of Online Social networks: Facebook, Google+, Twitter, LinkedIn
13.	Network Modeling tools overview Beyond, Social Network Analysis
14.	Modeling Software Components as Agents in Networks
15.	Presentation
16.	Wrap Up Your Work and Presentation



**Course Name: Data Warehousing**

**Course Code:** CS-355

**Course Structure:** Lectures: 2 Lab: 1 Credit Hours: 3

**Prerequisites:** Introduction to Database Systems

**Objective:** This course will provide an overview of the concepts and techniques how to Improving information access, Bringing the user in touch with their data, Enhancing the quality of decisions, providing cross-function integration and a database designing techniques to enable business intelligence activities.

**Outcomes:** Students should be able to:

- Discuss the role of data warehousing and enterprise intelligence in industry and government.
- Summarise the dominant data warehousing architectures and their support for quality attributes.
- Recognise and describe at least three computational approaches to data clustering, taking cognizance of the contribution of paradigms from the fields of Artificial Intelligence and Machine learning.
- Compare and contrast the dominant data mining algorithms.
- Construct a lightweight prototype or simulation that supports the concept of data mining.
- Analyze the results generated from the constructed artifact to determine if patterns of clusters were detected in the data sets.
- Demonstrate an appreciation of the importance of paradigms from the fields of Artificial Intelligence and Machine Learning to data mining.

**Course Outline:**

Need for DW, Evolution of Business intelligence, DW building blocks, Intro to data marts, architectural types, Trends, Web enabled DW, Planning and Project management, Defining Requirements, Metadata, Storage Specifications, Info delivery strategy, Architectural components and frameworks, Tools, Types of functional areas for metadata, Schemas, Star Schema, Dimensional Modelling, Data Extraction, Transformation and loading, OLAP Models, Data Quality.

**Reference Material:**

1. *Data warehousing fundamentals for IT Professionals* by Paulraj ponniah, (2011). John Wiley and Sons. (Latest Edition)



Week #	Distribution
1.	Need for DW
2.	Evolution of Business intelligence,
3.	DW building blocks, Intro to data marts
4.	architectural types
5.	Trends, Web enabled DW, Planning and Project management
6.	Defining Requirements
7.	Metadata, Storage Specifications
8.	Info delivery strategy
9.	Architectural components and frameworks
10.	Tools, Types of functional areas for metadata
11.	Schemas, Star Schema
12.	Dimensional Modelling, Data Extraction
13.	Transformation and loading
14.	OLAP Models, Data Quality.
15.	Presentation
16.	Wrap Up Your Work And Presenation



## **Course Name: Data Security and Encryption**

### **Credit Hours: 3**

**Prerequisites:** Discrete Structures, Data Structures and Algorithms

**Objectives:** This is an introductory course on the methods, algorithms, techniques, and tools of data security and cryptography. After studying the theoretical aspects of cryptographic algorithms and protocols, we show how these techniques can be integrated to solve particular data and communication security problems. This course material is of use to computer and communication engineers who are interested in embedding security into an information system, and thus, providing integrity, confidentiality, and authenticity of the documents and the communicating parties.

**Outcomes:** Students should be able to:

1. Understand basic concepts in number theory and apply modular arithmetic in problem solving
2. Explain the setups, protocols, and security issues of some conventional and modern cryptosystems
3. Design secure crypto-schemes to achieve simple tasks and explain their security issues.

### **Course Outline:**

The course consists of three parts: mathematical background, cryptography, and network security. The first part (mathematical background) introduces the principle of number theory and some results from probability theory, including Primes, random numbers, modular arithmetic and discrete logarithms. The second part (cryptography) covers cryptographic algorithms and design principles, including conventional and symmetric encryption (DES, IDEA, Blowfish, Rijndael, RC-4, RC-5), public key or asymmetric encryption (RSA, Diffie-Hellman), key management, hash functions (MD5, SHA-1, RIPEMD-160, HMAC), digital signatures, and certificates. The third part (network security) deals with practical applications that have been implemented and are in use to provide network security, including authentication protocols (X.509, Kerberos), electronic mail security (S/MIME, PGP), web security and protocols for secure electronic commerce (IPSec, SSL, TLS, SET).

### **Reference Material:**

1. *Cryptography and Network Security: Principles and Practice*, William Stallings, 6th edition (latest), Prentice Hall, 2005.



Week #	Distribution
1.	mathematical background, cryptography
2.	network security
3.	The first part (mathematical background)
4.	including Primes, random numbers, modular arithmetic and discrete logarithms
5.	The second part (cryptography) covers cryptographic algorithms and design principles, including conventional and symmetric encryption (DES, IDEA, Blowfish, Rijndael, RC-4, RC-5)
6.	public key or asymmetric encryption (RSA, Diffie-Hellman)
7.	key management, hash functions (MD5, SHA-1, RIPEMD-160
8.	, HMAC), digital signatures, and certificates. The third part (network security
9.	deals with practical applications that have been implemented and are in use to provide network security
10.	including authentication protocols (X.509, Kerberos), electronic mail security (S/MIME, PGP)
11.	web security
12.	web security and protocols
13.	secure electronic commerce (IPSec, SSL, TLS, SET).
14.	web security and protocols for secure electronic commerce (IPSec, SSL, TLS, SET).
15.	Presentation
16.	Wrap Up Your Work And Presentaion



## **Course Name: Secure Software Systems**

**Credit Hours: 3**

### **Prerequisites: Software Architecture and Design**

**Objective:** The objective of course is to introduce the student to Secure Software Development Life Cycle (will now on be referenced to as S-SDLC). There are multiple reasons why programs like these have gained popularity. We can say to a certain extent that they have become mandated in certain organizations. While this article will give a brief explanation about SDLC, for the sake of completeness, it does not explain SDLC in detail and all of its aspects.

**Outcomes:** Students who complete the course will have demonstrated the ability to do the following:

- Explain the most common weaknesses in software security and understand how such problems can be avoided in software.
- Identify common security threats, risks, and attack vectors for software systems.
- Evaluate and assess current security best practices and defense mechanisms for current software systems. Become aware of limitations of existing defense mechanisms and how to avoid them.
- Identify security problems in source code and binaries, assess the associated risks, and reason about their severity and exploitability.
- Assess the security of given source code or applications.

### **Course Outline:**

Different techniques to prevent or detect problems including: threat modeling, check lists and coding standards, To grasp static analysis tools, Understand code reviews, typing and static analysis, To comprehend language-based security (or platform-based security), security middleware and runtime monitoring.

### **Reference Materials:**

1. Building Secure Software, by John Viega and Gary McGraw. Addison- Wesley, 2002. (latest edition)
2. The 24 Deadly Sins of Software Security, by Michael Howard, David LeBlanc and John Viega, McGraw-Hill, 2009 (latest edition)
3. Secure Coding: Principles & Practices, by Mark G. Graff and Kenneth R. van Wyk. O'Reilly, 2003. (latest edition)



4. Writing Secure Code, by Michael Howard and David LeBlanc, Microsoft Press, 2002. (latest edition).

Course Name: <b>Secure Software Systems</b>	
Week #	Distribution
1.	Different techniques to prevent
2.	Different techniques to detect problems
3.	threat modeling
4.	check lists and coding
5.	coding standards
6.	static analysis tools
7.	coding standards, To grasp static analysis tools
8.	Understand code reviews, typing and static analysis
9.	typing and static analysis
10.	language-based security
11.	platform-based security
12.	To comprehend language-based security (or platform-based security
13.	security middleware
14.	runtime monitoring.
15.	Presentation
16.	Wrap Up Your work and Presentation



## **Course Name: Introduction to Bioinformatics**

**Course Code:** CS-357

**Credit Hours:** 3

**Prerequisites:** Introduction to Computing, Discrete Structures

**Objectives:** This course introduces the scientist to Bioinformatics, which uses computer databases to store, retrieve and assist in understanding biological information. Genome-scale sequencing projects have led to an explosion of genetic sequences available for automated analysis. These gene sequences are the codes, which direct the production of proteins that in turn regulate all life processes. The student will be shown how these sequences can lead to a much fuller understanding of many biological processes allowing pharmaceutical and biotechnology companies to determine for example new drug targets or to predict if particular drugs are applicable to all patients. Students will be introduced to the basic concepts behind Bioinformatics. Hands-on sessions will familiarize students with the details and use of the most commonly used online tools and resources.

### **Course Outline:**

Origin of the field , Advances in biology and computers , Brief overview of key Biological concepts related to DNA, RNA, nucleotides, amino acids, proteins, protein interaction, Databases and web resources, Algorithms how to write them, and calculate their complexities, etc. , Nucleotide analysis principals and tools , Sequence similarity, Dot Matrix, Dynamic Programming for local, Global pair wise alignment using Smith-Waterman and Needle-Wunsch algorithms, Gap penalties including Affine gap penalty, Scoring and Substitution Matrices (PAM & BLOSUM), Multiple Sequence Alignment, BLAST and FASTA, etc., Dynamic programming algorithms, Statistical models, Artificial intelligence algorithms, Protein analysis including protein structure prediction from a sequence. , Phylogenetics, Mutations, evolution and protein families, clustering, predictions using distance methods (such as UPGMA), etc.

### **Reference Materials:**

1. *Introduction to Bioinformatics* by T K Attwood, D J Parry-Smith, Samiron Phukan, Pearson Education (Latest edition)
2. *Introduction to Bioinformatics* by Arthur Lesk
3. *Algorithms in Bioinformatics* by Gary Benson, Roderic Page, Springer
4. *Algorithmic Aspects of Bioinformatics* by Hans-Joachim Bockenhauer, Dirk Bongartz, Springer.

Course Name: <b>Introduction to Bioinformatics</b>
--



Week #	Distribution
1.	Origin of the field , Advances in biology and computers ,
2.	Brief overview of key Biological concepts related to DNA
3.	RNA, nucleotides, amino acids, proteins, protein interaction
4.	Databases and web resources, Algorithms how to write them
5.	and calculate their complexities, etc. , Nucleotide analysis principals and tools
6.	Sequence similarity, Dot Matrix, Dynamic Programming for local, Global pair wise alignment using Smith-Waterman and Needle-Wunsch algorithms,
7.	Gap penalties including Affine gap penalty, Scoring and Substitution Matrices (PAM & BLOSUM), Multiple Sequence Alignment,
8.	BLAST and FASTA, etc., Dynamic programming algorithms
9.	Statistical models, Artificial intelligence algorithms, Protein analysis including protein structure prediction from a sequence
10.	Phylogenetics, Mutations,
11.	evolution
12.	protein families
13.	clustering, prediction
14.	clustering, prediction using distance methods (such as UPGMA
15.	Presentation
16.	Wrap Up Your Work and Presentation



**Course Name: System Analysis and Design**

**Course Code: SE-353**

**Credit Hours: 3**

**Prerequisites:** Programming Fundamentals

**Objective:** Upon successful completion of this course, the student will be able to:

- Define and describe the five phases of the system development life cycle.
- State at least five expected benefits from systems projects.
- Explain at least three ways in which information systems support business requirements.
- Describe how systems analysts interact with users, management, and other information systems professionals.
- Develop data flow diagrams and decision tables.
- Perform a feasibility study.
- Evaluate systems development alternatives.
- Solve realistic systems analysis problems.
- Determine methods for evaluating the effectiveness and efficiency of a system.
- Work as an effective team member on assigned projects.

**Course Outline:**

Concept of Entity; Relationships; System Outputs; System Inputs; People and Organizations; products; ordering of products; shipments; Invoicing; Account and Budgeting; Human resources; Creating the Data Warehouse Data model from the Enterprise Data Model.

**Reference Materials:**

- The Data Model Resource*, Revised Edition Volume 1; A Library of Universal Data Models for All Enterprises; Len Silverston; John Wiley & Sons, Inc.; (2001) or Latest Edition.

Course Name: : <b>System Analysis and Design</b>	
Week #	Distribution
1.	Concept of Entity
2.	Relationships
3.	System Outputs
4.	System Inputs
5.	People and Organizations



6.	products
7.	ordering of products;
8.	shipments
9.	Invoicing;
10.	Account and Budgeting
11.	Human resources
12.	Creating the Data Warehouse
13.	Data mode
14.	Data model from the Enterprise Data Model
15.	Presentation
16.	Wrap Up Your Work and Presentation



## **Course Name: Event Driven Programming**

**Course Code:** SE-354

**Credit Hours:** 3

**Prerequisites:** Object Oriented Programming

**Objective:** The course provides an introduction to finites state machines, and the event-driven programming paradigm. Upon completion of this course you will be able to:

1. Describe the different ways a finite-state machine can be represented;
2. Explain how a finite state machine recognizes an input string;
3. Explain how a non-deterministic finite state machine works;
4. Explain the behavior of regular expressions;
5. Translate a regular expression into a corresponding finite-state machine;
6. Build GUI-based software systems;
7. Explain a few basic design-patterns, and know when to apply them.

### **Course Outline:**

Introduction to the course and its importance, history and course format, Introduction to C# or Java, Event-driven programming basics, Maintaining state, On-demand Rendering in event-driven applications, Timer and perpetual tasks, Multithreading and event-driven programming, A "window/frame" as a drawing surface + event-handling unit , Interesting widgets, GUI design patterns, Performance issues, GUI on mobile devices/smart phones.

### **Reference Materials:**

1. Event Processing in Action, Opher Etzion (Latest Edition)
2. Windows Presentation Foundation Unleashed, Adam Nathan (Latest Edition)
3. Event-Based Programming: Taking Events to the Limit, Ted Faison, Apress (Latest Edition)

Course Name: : <b>Event Driven Programming</b>	
Week #	Distribution
1.	Introduction to the course
2.	history and course format
3.	Introduction to C#
4.	Introduction to Java



5.	Event-driven programming basics
6.	Maintaining state
7.	On-demand Rendering in event-driven applications
8.	Timer and perpetual tasks
9.	Multithreading and event-driven programming
10.	A "window/frame" as a drawing surface
11.	event-handling unit
12.	Interesting widgets, GUI design patterns
13.	Performance issues, GUI on mobile devices
14.	Performance issues, GUI on smart phones
15.	Presentation
16.	Wrap Up Your Work and Presentation



## Course Name: Aspect Oriented Software Design

**Credit Hours:** 3

### Prerequisites: Object Oriented Software Engineering

**Objectives:** Learn new skills in software development which allow you to develop significantly more flexible software. Acquire a working understanding of AOP through the use of AspectJ. Acquire a clear understanding of the adaptive object-oriented paradigm through five architectural patterns; Structure-Shy Traversal, Selective Visitor, Structure-Shy Object, Class Graph and Growth Plan. Apply the aspect-oriented paradigm and the adaptive object-oriented paradigm to problems solving, including the implementation of a project. Understand the connections between the adaptive object-oriented approach and the aspect-oriented approach and how they fit into generative programming.

### Course Outline:

AOSD is a novel programming paradigm that aims at a better separation of concerns. AOSD emerged from the academic world in the late nineties and experiences a significant acceptance by industry. The topics covered in the course are: Separation of Concerns and AOSD, Aspect-Oriented Programming in Aspect J, Case study: Design Patterns in AOP, Framework-based AOP approaches, Case study: Aspects in Enterprise Software, Current research topics in AOSD

### Reference Materials:

1. Using Aspect-Oriented Programming for Trustworthy Software Development, Vladimir O. Safonov, John Wiley and Sons (2008) or Latest Edition
2. *Aspect in Action: Enterprise AOP with Spring Applications*, Ramnivas Laddad. Manning Publication Co. (2009) or Latest Edition.

Course Name: : <b>Aspect Oriented Software Design</b>	
Week #	Distribution
1.	Intro to AOSD
2.	Intro To a novel programming paradigm
3.	separation of concerns
4.	academic world
5.	experiences a significant acceptance
6.	Separation of Concerns
7.	Aspect-Oriented Programming in Aspect J
8.	Case study:



9.	Design Patterns in AOP
10.	Framework-based AOP approaches
11.	Case study:
12.	Aspects in Enterprise Software
13.	Current research topics in AOSD
14.	Presentation
15.	Presentation
16.	Wrap Up Your Work And Presentation



## Course Name: Agent Based Computing

Credit Hours: 3

### Prerequisites: Object Oriented Programming

**Objective:** After taking this course, the participants will have an understanding of the agent system terminology and development process of agent-based systems, will have learned techniques to design agent-based system, and will know how to modify architecture of the current software systems and re-structure them to be agent-based.

### Course Outlines:

Introduction to Agent-based Computing, Agent-based Models, Modeling Complex Real-world Problems using Agents, Introduction to Net Logo, Describing ABMs, From Animation to Science, Model Verification & Validation, Model Design, Emergence, Observation, Adaptive Behaviour, Prediction, Software Multiagent Systems, Intelligent Agents, Deductive Reasoning Agents, Agent Methodologies.

### Reference Materials:

1. *Agent-Based and Individual-Based Modeling: A Practical Introduction*, Steven F. Railsback and Volker Grimm (Latest Edition)
2. *An Introduction to MultiAgent Systems*, Michael Woodridge, Wiley (Latest Edition)
3. *Managing Business Complexity: Discovering Strategic Solutions with Agent-Based Modelling and Simulation*, Michael J. North and Charles M. Macal, Oxford University Press (Latest Edition)
4. *Cognitive Agent-based Computing-I*, Muaz A. Niazi and Amir Hussain, Springer (Latest Edition).

Course Name: : <b>Agent Based Computing</b>	
Week #	Distribution
1.	Introduction to Agent-based Computing
2.	Agent-based Models,
3.	Modeling Complex Real-world Problems using Agents
4.	Introduction to Net Logo
5.	Describing ABMs
6.	From Animation to Science
7.	Model Verification & Validation
8.	Model Verification & Validation



9.	Model Design, Emergence
10.	Observation, Adaptive Behaviour
11.	Prediction, Software Multiagent Systems
12.	Intelligent Agents
13.	Deductive Reasoning Agents
14.	Agent Methodologies
15.	Presentation
16.	Wrap Up Your Work And Presentation



**Course Name: Social Networks**

**Course Code:** IT-456

**Credit Hours:** 3

**Prerequisites:** None

**Objective:** This course will teach students how to use the major new tools in social media with a focus on best use and efficient use. Students will learn how to use social media to reach personal and professional goals. Whether goals be web, business, blog or personal reasons students will be empowered to use to the tools. Upon completion of this course students will have a strong grasp of new tools in social media and be able to leverage those tools to advance their goals. The course will include guest speakers from the frontlines in social media and networking.

**Course Outline:** Link analysis and network community detection, diffusion and information propagation on the web, virus outbreak detection in networks, and connections with work in the social sciences and economics.

**Reference Material:**

1. *Social and Economic Networks*, Mathew O Jackson, Princeton University Press, (2010). or Latest Edition

Course Name: : <b>Social Networks</b>	
Week #	Distribution
1.	Link analysis
2.	network community
3.	detection
4.	detection
5.	diffusion
6.	diffusion
7.	information propagation on the web
8.	virus outbreak
9.	virus outbreak detection
10.	virus outbreak detection in networks
11.	connections
12.	connections with work



13.	connections with work in the social sciences
14.	connections with work in the economics.
15.	Presentation
16.	Wrap Up Your Work Presentation



**Course Name: Functional Programming**

**Course Code: SE-355**

**Credit Hours: 3**

**Prerequisites: Programming Fundamentals**

**Objective:** The goal of the course is to delve deeper in to the principles of program design, implementation and understanding. We wish to help students become superb programmers who can design, implement and reason about software that is elegant, efficient, and correct, and whose code can be maintained and reused.

**Learning Outcome:** Explain Computer Programming concepts, like;

- Ability to understanding and design algorithmic solution to problems
- Ability to design programs with Interactive Input and Output
- Ability to design programs utilizing arithmetic expressions
- Ability to design programs utilizing repetition
- Ability to design programs utilizing decision making
- Ability to test and verifying programs
- Ability to develop efficient, and correct, and whose code can be maintained and reused.

**Course Outline:**

Introduction. Functions, lists and Recursion; map, filter, fold; binding; algebraic data types, abstract types; type classes; logic and programs; applications and domains for functional programming.

**Reference Materials:**

1. *The Craft of Functional Programming*, Simon Thompson, Addison Wesley. (Latest Edition)
2. *Programming in Haskell*, Graham Hutton, Cambridge University Press. (Latest Edition)
3. *The Haskell School of Expression*, Paul Hudak, Cambridge University Press, (Latest Edition).

Course Name: : <b>Functional Programming</b>	
Week #	Distribution
1.	Introduction. Functions
2.	Introduction lists
3.	Introduction Recursion
4.	Introduction map
5.	Introduction filter



6.	Introduction fold
7.	Introduction binding
8.	algebraic data types
9.	abstract types
10.	type classes
11.	applications
12.	logic and programs
13.	domains for functional programming
14.	domains for functional programming.
15.	Presentation
16.	Wrap Up Your Work and Presentation



**Course Name: Operations Research**

**Course Code: SE-303**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives:** Students will become familiar with techniques of modeling real world problems. They will gain facility in working with a number of the most common models and modeling patterns. They will understand and be able to apply the notions of sensitivity analysis. They will be able to select appropriate deterministic or stochastic models in a wide variety of common situations.

**Course Outcomes:** At the end of the course the students will be able to:

1. Learn the characteristics of different types of decision making environments, appropriate decision making approaches and tools to be used in each type.
2. Solve the Transportation Models and Assignment Models.
3. Understand the basic methodology for the solution of linear programs and integer programs.

**Course Learning Outcomes (CLOs):**

At the end of the course the students will be able to:

1. Learn the characteristics of different types of decision-making environments appropriate decision making approaches and tools to be used in each type.
2. Solve the Transportation Models and Assignment Models.
3. Understand the basic methodology for the solution of linear programs and integer programs.

**Course Outline:**

Introduction to mathematical modeling. Linear program models, simplex method for solving LP models, sensitivity analysis, other solution techniques for LP models, specialized LP models (transport, assignment, etc.). Network based models, shortest path, min weight spanning tree, max flow, PERT/CPM. Decision models, dynamic programming, games theory. Probabilistic models, expected return models, Markov chains, stochastic processes, queuing models, stochastic inventory models.

***Sample labs and assignments:***

Given a scenario, select and develop an appropriate model, solve it for the given parameters, and analyze the sensitivity of the solution to changes in the problem parameters.

**Reference Materials:**

1. *Operations Research: An Introduction*, Hamdi A. Taha, (8/e), (2006) or Latest Edition



2. *Introduction to Operations Research*, F.S. Hillier, and G. J. Leibermann, (8/e), (2005) or Latest Edition

Course Name: : <b>Operations Research</b>	
Week #	Distribution
1.	Introduction to mathematical modeling
2.	Linear program models, simplex method for solving LP models
3.	sensitivity analysis, other solution techniques for LP models
4.	specialized LP models (transport, assignment, etc.)
5.	Network based models, shortest path
6.	min weight spanning tree, max flow
7.	PERT/CPM.
8.	Decision models
9.	dynamic programming
10.	games theory
11.	Probabilistic models, expected return models
12.	Markov chains, stochastic processes
13.	queuing models
14.	stochastic inventory models
15.	Presentation
16.	Wrap Up Your and presentation



## **Course Name: Simulation and Modeling**

**Course Code:** SE-304

**Credit Hours:** 3

**Prerequisites:** Probability and Statistics, Data Structures

**Objectives:** This course emphasizes the development of modeling and simulation concepts and analysis skills necessary to design, program, implement, and use computers to solve complex systems/products analysis problems regarding software engineering discipline. The key emphasis is on problem formulation, model building, data analysis, solution techniques, and evaluation of alternative designs/processes in complex systems/products. Overview of modeling techniques and methods used in decision analysis, including Monte Carlo simulation and systems dynamics modeling are presented.

1. To apply modern software packages to conduct analysis of real world data.
2. To understand the technical underpinning of modern computer simulation software.
3. The ability to apply the appropriate analytical technique to a wide variety of real world problems and data sets.

To summarize and present the analysis results in a clear and coherent manner.

### **Course Outline:**

Introduction to Simulation and Modeling, Discrete-Event Simulation, Simulation of a Single-Server Queueing System, Alternative Approaches to Modeling and Simulations; Review of Basic Probability and Statistics; Estimation of Means, Variances, and Correlations, Confidence Intervals and Hypothesis Tests for the Mean, The Laws of Large Numbers; Random number generators; Simulation of discrete, continuous probability distributions and empirical distributions; tests on simulated distributions, rejection method, simulation of multivariate distributions, correlations, and stochastic processes, simulation of models of arrival processes, Poisson Processes, Nonstationary Poisson Processes, Batch Arrivals, tests on generators, Markov- Chain Monte-Carlo simulations; Variance-Reduction Techniques.

### **Reference Materials:**

1. *Simulation Modelling and Analysis*, A.M. Law and W.D. Kelton, McGraw Hill, (2000) or Latest Edition
2. *Discrete-event System Simulation*, J. Banks, J.S. Carson and B.L. Nelson, Prentice Hall International, (1994) or Latest Edition



3. *Probabilistic Modelling*, Mitrani, Cambridge University Press, (1998) or Latest Edition
4. *Simulation and Modelling*, Sheldon M. Ross, (2002) or Latest Edition
5. *Stochastic Simulations*, Brian Ripley. (Latest Edition)

Course Name: : <b>Simulation and Modeling</b>	
Week #	Distribution
1.	Introduction to Simulation and Modeling
2.	Discrete-Event Simulation
3.	Simulation of a Single-Server Queueing System
4.	Alternative Approaches to Modeling and Simulations
5.	Review of Basic Probability and Statistics
6.	Estimation of Means, Variances, and Correlations
7.	Confidence Intervals and Hypothesis Tests for the Mean
8.	The Laws of Large Numbers; Random number generators
9.	Simulation of discrete, continuous probability distributions and empirical distributions
10.	tests on simulated distributions, rejection method
11.	simulation of multivariate distributions, correlations, and stochastic processes
12.	simulation of models of arrival processes, Poisson Processes, Nonstationary Poisson Processes
13.	Batch Arrivals, tests on generators, Markov- Chain Monte-Carlo simulations; Variance-Reduction Techniques
14.	Presentation
15.	Presentation
16.	Wrap up your work and presentaion



## **Course Name: Principles of Management**

**Credit Hours: 3**

**Prerequisites: None**

### **Objectives:**

1. Relate, discuss, understand, and present management principles, processes and procedures in consideration of their effort on individual actions.
2. Participate, summarize and/or lead class discussions, case problems and situations from both the text and student experience that relate to the text material.
3. Knowledge and understanding of the Principles of Management will enable the student manager and/ or employee and gain valuable insight into the workings of business and other organizations.

**Course Outline:** Introduction to Managers and Management.

Organizational Culture and Environment. Decision Making. The Essence of Manager's Job. Planning. Organization Structure and Design. Motivation. Leadership. Communication. Controlling. The Personnel Function. Job Design and Analysis. Human Resource Planning. Recruitment and Selections/Testing and Interview. Union and Management, Compensation Administration, Health and Safety.

### **Reference Materials:**

1. Management by Robbins, S.P. & Coulter, Mary, Prentice Hall; 10th Edition (November 3, 2008). ISBN-10: 0132090716
2. Fundamentals of Management by Robbins, S.P. & DeCenzo, David A, Prentice Hall; 7th Edition (January 13, 2010). ISBN-13: 978-0132090711
3. Principles of Management by Charles W. L. Hill and Steven McShane, McGraw-Hill/Irwin; 1st Edition (2006). ISBN-10: 0073530123
4. Management by Richard L. Daft, South-Western College Pub; 10th Edition (January 27, 2011). ISBN-10: 0538479531
5. Fundamentals of Management by Stephen P. Robbins, David A. DeCenzo and Mary Coulter, Prentice Hall; 7th Edition (January 13, 2010). ISBN-10: 0136109829.

Course Name: : <b>Principles of Management</b>	
Week #	Distribution
1.	Introduction to Managers and Management.
2.	Organizational Culture and Environment
3.	Decision Making
4.	The Essence of Manager's Job



5.	Planning
6.	Organization Structure
7.	Design, Motivation and Leadership
8.	Communication
9.	Controlling. The Personnel Function.
10.	Job Design and Analysis. Human Resource Planning
11.	Recruitment and Selections
12.	Testing and Interview
13.	Union and Management
14.	Health and Safety.
15.	Presentation
16.	Wrap Up Your Work And Presentation



**Course Name: Human Resources Management**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives:** After successfully completing this program, you should be able to:

- Effectively manage and plan key human resource functions within organizations
- Examine current issues, trends, practices, and processes in HRM
- Contribute to employee performance management and organizational effectiveness
- Problem-solve human resource challenges
- Develop employability skills for the Pakistanis workplace
- Develop effective written and oral communication skills

**Course Outline:** Managing Human Resources. Understanding the External and Organizational Environments. Ensuring Fair Treatment and Legal Compliance. HR Planning for Alignment and Change. Using Job Analysis and Competency Modeling. Recruiting and Retaining Qualified Employees. Selecting Employees to Fit the Job and the Organization. Training and Developing a Competitive Workforce. Conducting Performance Management. Developing an Approach to Total Compensation. Using Performance-Based Pay to Achieve Strategic Objectives. Providing Benefits and Services for Employees' Well-Being, Risk Management. Employee Relations. Risk Management. Health, Safety, and Employee Well-Being. Understanding Unionization and Collective Bargaining.

**Reference Materials:**

1. Managing Human Resources by Susan E. Jackson, Randall S. Schuler and Steve Werner, South-Western College Pub; 11th Edition (June 16, 2011).ISBN-10: 1111580227.
2. Management of Human Resources by Gary Dessler, Carolin Rekar Munro and Nina D. Cole, Pearson Education Canada; 3rd Edition (February 28, 2010). ISBN-10: 0321687140 .
3. Human Resource Management by Robert L. Mathis and John H. Jackson, South-Western Cengage Learning; 13th Edition (August 19, 2010). ISBN- 10: 053845315X.
4. Human Resource Management Applications: Cases, Exercises, Incidents, and Skill Builders by Stella M. Nkomo, Myron D. Fottler and R. Bruce McAfee, South-Western Cengage Learning; 7th Edition (September 29, 2010). ISBN-10: 053846

Course Name: : <b>Human Resources Management</b>
--



Week #	Distribution
1.	Managing Human Resources.
2.	Understanding the External and Organizational Environments
3.	Ensuring Fair Treatment and Legal Compliance
4.	HR Planning for Alignment and Change
5.	Using Job Analysis and Competency Modeling
6.	Recruiting and Retaining Qualified Employees
7.	Selecting Employees to Fit the Job and the Organization
8.	Training and Developing a Competitive Workforce
9.	Conducting Performance Management
10.	Developing an Approach to Total Compensation. Using Performance-Based Pay to Achieve Strategic Objectives
11.	Providing Benefits and Services for Employees' Well-Being
12.	Risk Management. Employee Relations
13.	Risk Management. Health, Safety
14.	Employee Well-Being. Understanding Unionization and Collective Bargaining.
15.	Presentation
16.	Wrap Up Your Work and Presentation



**Course Name: Network Management**

**Credit Hours: 3 (Lab may be assigned or adjusted by the University)**

**Prerequisites: Computer Communication and Networks**

**Objectives:**

- Understand the challenges of network communication.
- Understand the basics of network communication.
- Understand the operation of the protocols that are used inside the Internet.

Course Name: : <b>Network Management</b>	
Week #	Distribution
1.	Analyzing Business Goals and Constraints
2.	Top-Down Network Design Methodology.
3.	Designing a Network Topology. Designing
4.	Models for Addressing and Numbering. Selecting Switching and Routing Protocols
5.	Developing Network Security Strategies. Developing Network Management Strategies
6.	Physical Network Design.
7.	Selecting Technologies and Devices for Enterprise
8.	Networks. Testing Network Design
9.	Optimizing Network Design. Documenting Network Design.
10.	Network Management Standards & Models
11.	SNMP Management.
12.	SNMP Management
13.	Broadband Network Management
14.	Broadband Network Management
15.	Presentation
16.	Presentation And Wrap Up Your Work

**Course Outline:** Analyzing Business Goals and Constraints. Top-Down Network Design Methodology. Characterizing the Existing Internetwork. Designing a Network Topology. Designing



Models for Addressing and Numbering. Selecting Switching and Routing Protocols. Developing Network Security Strategies. Developing Network Management Strategies. Physical Network Design. Selecting Technologies and Devices for Enterprise Networks. Testing Network Design. Optimizing Network Design. Documenting Network Design. Network Management Standards & Models. SNMP Management. SNMP Management. Broadband Network Management.

### Reference Materials:

1. Top-Down Network Design by Priscilla Oppenheimer, Cisco Press; 3rd Edition (September 3, 2010). ISBN-10: 1587202832 (TB1)
2. Network Management: Principles and Practice by Mani Subramanian; Timothy A. Gonsalves; N. Usha Rani, Pearson Education India (2010). ISBN-10: 81-3172-759-9
3. Networking Systems Design and Development by Lee Chao, CRC Press; 1st Edition (December 21, 2009). ISBN-10: 142009159X (TB2)
4. Networks: Design and Management by Steven Karris, Orchard Publications (August 2002). ISBN-10: 0970951140

Course Name: : Analyzing Business Goals	
Week #	Distribution
1.	Analyzing Business Goals and Constraints
2.	Top-Down Network Design Methodology
3.	Characterizing the Existing Internetwork
4.	Designing a Network Topology. Designing Models for Addressing
5.	Numbering
6.	Selecting Switching and Routing Protocols
7.	Developing Network Security Strategies
8.	Developing Network Management Strategies
9.	Physical Network Design.
10.	Selecting Technologies and Devices for Enterprise Networks
11.	Testing Network Design. Optimizing Network Design
12.	Documenting Network Design
13.	Network Management Standards & Models. SNMP Management. SNMP Management



---

14.	Broadband Network Management.
15.	Presentation
16.	Wrap Up Your Work And Presentation

5. Network Warrior by Gary A. Donahue, O'Reilly Media; 2nd Edition (May 13, 2011). ASIN: B004W8ZL3W.



**Course Name: Database Administration & Management**

**Course Code: CS-463**

**Credit Hours: 3**

**Prerequisites: Database Systems**

**Objectives:** At the completion of this course, students should be able to do the following:

1. Understand the role of a database management system in an organization.
2. Understand basic database concepts, including the structure and operation of the relational data model.
3. Construct simple and moderately advanced database queries using Structured Query Language (SQL).
4. Understand and successfully apply logical database design principles, including E-R diagrams and database normalization.
5. Design and implement a small database project using Microsoft Access.
6. Understand the concept of a database transaction and related database facilities, including concurrency control, journaling, backup and recovery, and data object locking and protocols.
7. Describe and discuss selected advanced database topics, such as distributed database systems and the data warehouse.
8. Understand the role of the database administrator.

**Learning Outcome:**

- Understand the role of a database management system in an organization.
- Understand basic database concepts, including the structure and operation of the relational data model.
- Understand the role of the database administrator.

**Course Outline:** Installation of DBMS; SQL\* Plus; DBA Tools. DBMS Physical Structure & Architectural Components: Server, Instance, SGA, Shared Pool, Library Cache, Data Dictionary Cache, Large Pool, Processes. Startup and Shutdown Database. Managing Instances. Managing Files. Creating Database and Data dictionary. Managing Table spaces. Operations with Table spaces. Data File Management, Segments, Block. Managing Undo Data, Undo Data Statistics: Managing Tables and Users. Indexes Management, Maintaining Data Integrity, Constraints. Managing



Privileges. Server Side Configuration. Client Side Configuration. Usage and Configuration of Oracle Shared Server. Backup and Recovery. Sizing Shared

Pool, Sizing Buffer Cache, I/O Issues. Tuning Rollback Segments. Tuning Shared Servers, Types of Locks, Block Efficiency, Storage hierarchy, Avoiding Dynamic allocation, Statistics, PCTFREE and PCTUSED, Monitoring Index Usage.

### Reference Materials:

1. Database Administration: The Complete Guide to DBA Practices and Procedures by Craig S. Mullins, Addison-Wesley Professional; 2nd Edition (October 21, 2012). ISBN-10: 0321822943
2. Database Systems: A Practical Approach to Design, Implementation and Management by Thomas M. Connolly and Carolyn E. Begg, Addison- Wesley; 5th Edition (2009). ISBN-10: 0321523067
3. Online Material URL <http://www.oracle.com/technetwork/index.html> (this should be avoided)

Course Name: : <b>Database Systems</b>	
Week #	Distribution
1.	Installation of DBMS; SQL* Plus; DBA Tools. DBMS Physical Structure & Architectural Components
2.	Server, Instance, SGA, Shared Pool, Library Cache, Data Dictionary Cache, Large Pool,
3.	Processes. Startup and Shutdown Database. Managing Instances. Managing Files. Creating Database and Data dictionary
4.	Managing Tablespaces. Operations with Tablespaces. Data File Management
5.	Segments, Block. Managing Undo Data, Undo Data Statistics: Managing Tables and Users
6.	Indexes Management, Maintaining Data Integrity
7.	Constraints. Managing Privileges. Server Side Configuration
8.	Client Side Configuration. Usage
9.	Configuration of Oracle Shared Server
10.	Backup and Recovery and.. Sizing Shared



11.	Pool, Sizing Buffer Cache, I/O Issues. Tuning Rollback Segments
12.	Tuning Shared Servers, Types of Locks, Block Efficiency
13.	Storage hierarchy, Avoiding Dynamic allocation
14.	Statistics, PCTFREE and PCTUSED, Monitoring Index Usage.
15.	Presentation
16.	Wrap Up Your Work



**Course Name:** Web Technologies

**Course Code:** IT-451

**Course Structure:** Lectures: 2, Labs: 1

**Credit Hours:** 3(2+1)

**Prerequisites:** Fundamentals of programming

**Objective:** Students will learn to create and design web pages with text, graphics, and data tables. Students will then link these web pages to enable navigation between these web pages. Students will also create advanced web pages, test their validity and use new tools available in the market for web designing.

**Outcomes:** At the end of the course, the students will learn to develop a static web site using HTML, HTML5, CSS3 and Java script. Also be able to learn advance tools and technologies used for web development now a day in market.

**Course Outline:**

HTML, DHTML, CSS, clients side scripting, server side scripting, dynamic website development. Introduction to current technology e.g. MySQL, PHP, ASP, ASP.net. Introduction to related methods and tools e.g., website hosting, database connectivity, Macromedia. Overview of XML

**Reference Materials:**

1. *Web enabled Commercial application development using HTML, DHTML, JAVASCRIPT* by Ivon Bayross. BPS Publications.
2. *Beginning ASP* by Richard Anderson, Brain Francis. Wrox series Publications.

**Course Name:** Business Process Engineering

**Course Code:** SE-451

**Credit Hours:** 3

**Prerequisites:** None

**Objectives and Outcomes:** Upon completion of this course, learners should be able to:

- Define Business Process Engineering and articulate its business case.
- Articulate how Business Process Engineering will improve a business's processes and overall bottom line.
- Analyze private and public sector Business Process Engineering frameworks and their relative merits and limitations.



- Evaluate Business Process Engineering tools with respect to best practices and assess suitability for varying applications.
- Demonstrate how Business Process Engineering integrates strategic goals and business processes with supporting technology architectures to enable business objectives.
- Describe key business processes, and their integrative nature, in an organization.
- Explain the role of enterprise systems in supporting business processes.
- Describe the organizational data related to financial accounting Identify and discuss the various integration points among procurement, fulfillment, production, and warehouse management processes.
- Analyze the financial and material impacts of various steps in the integrated processes.
- Understand SAP organizational structures
- Combine business processes and SAP organizational structure (i.e., configuration)

## Course Outline:

## Reference Materials:

1. Business Process Improvement; the Breakthrough Strategy for Total Quality, Productivity, and Competitiveness, H. J. Harrington
2. Business Intelligence: A Managerial Approach by Turban, Sharda, Delen, King, 2nd Edition, Prentice Hall (2011). ISBN: 13-978-0-136-10066-9

Course Name: : <b>Business Process Engineering</b>	
Week #	Distribution
1.	Business process management
2.	Manufacturing
3.	services processes
4.	Modelling
5.	charting tools
6.	Lean processes
7.	Lean processes Improvement workshop techniques
8.	Business process
9.	Outsourcing
10.	Business process outsourcing



11.	Re-engineering
12.	Re-engineering
13.	improvement cases
14.	improvement cases
15.	Presentation
16.	Wrap Up Your Work And Presentation



**Course Name: Stochastic Processes**

**Credit Hours: 3**

**Prerequisites: Probability and Statistics**

**Objectives and Outcome:**

At the end of the course the students will be able to:

1. Define basic concepts from the theory of Markov chains and present proofs for the most important theorems.
2. Compute probabilities of transition between states and return to the initial state after long time intervals in Markov chains.
3. Derive differential equations for time continuous Markov processes with a discrete state space.
4. Solve differential equations for distributions and expectations in time continuous processes and determine corresponding limit distributions.

**Course Outline:** Discrete Markov chains, classification of states, first passage and recurrence times, absorption problems, stationary and limiting distributions. Chapman-Kolmogorov equations, Long run behavior of Markov chains, Absorption probabilities and expected times to absorption, Statistical aspects of Markov chains, The mover-stayer model, Application of a Markov chain and mover-stayer model to modeling repayment behavior of bank loans' grantees. Markov Processes in continuous time: Poisson processes, birthdeath processes. Poisson process The Kolmogorov differential equations, Limiting behavior of continuous time Markov chains The Q matrix, forward and backward differential equations, imbedded Markov Chain, stationary distribution. renewal theory, Brownian Motion and its generalizations, Discrete time martingales, Conditional expectation, Definition of a martingale and examples, Optional stopping theorem, Stochastic calculus

**Reference Materials:**

1. Introduction to Probability Models, 11th Ed, Sheldon M. Ross, Academic Press 2014.
2. Essentials of stochastic processes, Durrett, Richard. Springer Science & Business Media, 2nd Ed, 2012.
3. Introduction to Stochastic Processes, 2nd Ed, G.F. Lawler, Chapman and Hall, Probability Series, 2006

Course Name: <b>Stochastic Processes</b>	
Week #	Distribution
1.	Discrete Markov chains, classification of states



2.	first passage and recurrence times, absorption problems
3.	stationary and limiting distributions.
4.	Chapman-Kolmogorov equations
5.	Long run behavior of Markov chains, Absorption probabilities and expected times to absorption
6.	Statistical aspects of Markov chains, The mover-stayer model, Application of a Markov chain and mover-stayer model to modeling repayment behavior of bank loans' grantees
7.	Markov Processes in continuous time: Poisson processes
8.	Birth death processes. Poisson process The Kolmogorov differential equations
9.	Limiting behavior of continuous time Markov chains The Q matrix, forward and backward differential equations
10.	imbedded Markov Chain, stationary distribution
11.	renewal theory, Brownian Motion and its generalizations
12.	Discrete time martingales, Conditional expectation
13.	Definition of a martingale and examples
14.	Optional stopping theorem, Stochastic calculus
15.	Presentation
16.	Wrap Up Your Work And Presentation



## **Course Name: Software Construction and Development**

**Course Code:** SE-303

**Credit Hours:** 3 (2+1)

**Prerequisites:** Software Design and Architecture

### **Objectives and Outcomes:**

At the end of the course the students will be able to:

1. Understand the role of design and its major activities within the OO software development process, with focus on the Unified process
2. Develop Object-oriented design models and refine them to reflect implementation details
3. Evaluate different architectures for a medium size software.
4. Implement design model using an object-oriented programming language.

**Course Outline:** Software development process, Software engineering process infrastructure, Software engineering process improvement, Systems engineering life cycle models, Process implementation, Levels of process definition, Life cycle model characteristics, Individual and team software process, Lehman's Laws, code salvaging, and configuration management. Martin Fowler's refactoring concepts and their application to small projects.

Apply Michael Feathers' "legacy code" concepts. Exception handling, making methods robust by having them check their inputs sent from calling objects. Software configuration management, Release management, Software configuration management processes, Software deployment processes, Distribution and backup, Evolution processes and activities, Basic concepts of evolution and maintenance, Working with legacy systems, Refactoring, Error handling, exception handling, and fault tolerance. Personal reviews (design, code, etc.), Peer reviews (inspections, walkthroughs, etc.).

### **Reference Materials:**

1. Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, Prentice Hall, 2008.
2. The Pragmatic Programmer: From Journeyman to Master, Andrew Hunt and David Thomas, Addison-Wesley Professional, 1999.
3. Working Effectively with Legacy Code, Michael C. Feathers. Pearson Education, Prentice-Hall, 2004.



4. Refactoring: Improving the Design of Existing Code, Martin Fowler, Addison-Wesley Professional. 1999.

Course Name: <b>Software Construction and Development</b>	
Week #	Distribution
1.	Software development process, Software engineering process infrastructure
2.	Software engineering process improvement, Systems engineering life cycle models
3.	Process implementation, Levels of process definition, Life cycle model characteristics
4.	Individual and team software process, Lehman's Laws, code salvaging
5.	configuration management. Martin Fowler's refactoring concepts and their application to small projects
6.	Apply Michael Feathers' "legacy code" concepts. Exception handling
7.	making methods robust by having them check their inputs sent from calling objects.
8.	Software configuration management, Release management
9.	Software configuration management processes, Software deployment processes
10.	Distribution and backup, Evolution processes and activities, Basic concepts of evolution and maintenance
11.	Working with legacy systems, Refactoring
12.	Error handling, exception handling, and fault tolerance
13.	Personal reviews (design, code, etc.)
14.	Peer reviews (inspections, walkthroughs, etc.).
15.	Presentation
16.	Warp Up Your Work and Assignment



## **Course Name: Software Requirements Engineering**

**Course Code:** SE-201

**Credit Hours:** 3

**Prerequisites:** Software Engineering

**Objectives and Outcomes:** At the end of the course the students will be able to:

1. Describe the requirements engineering process
2. Effectively analyze software requirements for the development of cost-effective and efficient technical solutions.
3. Prepare both functional and non-functional requirements along with validation for a medium-size software system.
4. Document effective requirements in Software Requirements Specification (SRS) using clear, unambiguous requirements

**Course Outline:** Introduction to Requirements Engineering, Software Requirements, classification of requirements, Requirements process, Levels/layers of requirements, Requirement characteristics, Analyzing quality requirements, Software requirements in the context of systems engineering, Requirement evolution, requirement traceability, requirement prioritization, trade-off analysis, risk analysis and impact analysis, Requirement management, interaction between requirement and architecture, Requirement elicitation, elicitation sources and techniques, Requirement specification and documentation, specification sources and techniques, Requirements validation and techniques, Management of Requirements, Introduction to Management, Requirements Management Problems , Managing Requirements in an Acquisition Organization, Supplier Organizations, Product Organizations, Requirements engineering for agile methods.

### **Reference Materials:**

1. Software Requirements, Wiegers K. & Beatty J., 3rd Ed. Microsoft Press, 2013
2. Requirements Engineering, Elizabeth Hull, Ken Jackson and Jeremy Dick. 3rd Ed, Springer-Verlag London Limited, 2011.
3. Requirements Engineering and Management for Software Development Projects, Chemuturi M., Springer New York, 2013.

Course Name: <b>Software Requirements Engineering</b>	
Week #	Distribution



1.	Introduction to Requirements Engineering, Software Requirements
2.	classification of requirements, Requirements process
3.	Levels/layers of requirements, Requirement characteristics
4.	Analyzing quality requirements,
5.	Software requirements in the context of systems engineering
6.	Requirement evolution, requirement traceability
7.	requirement prioritization, trade-off analysis, risk analysis and impact analysis
8.	Requirement management
9.	interaction between requirement and architecture, Requirement elicitation, elicitation sources and techniques
10.	Requirement specification and documentation, specification sources and techniques, Requirements validation and techniques
11.	Management of Requirements, Introduction to Management
12.	Requirements Management Problems , Managing Requirements in an Acquisition Organization
13.	Supplier Organizations, Product Organizations
14.	Requirements engineering for agile methods.
15.	Presentation
16.	Presentation and Wrap Up your work



**Course Name: Software Re-Engineering**

**Course Code:** SE-359

**Credit Hours:** 3

**Prerequisites:** Software Engineering

**Objectives and Outcomes:** At the end of the course the students will be able to:

1. Explain the concepts and technique of software reengineering.
2. Apply reengineering techniques to maintain and modify software systems
3. Analyze and understand maintenance related problems associated with object oriented software systems.
4. Able to perform complex design reengineering and reverse engineering problems.

**Course Outline:** Salient topics include the terminology and the processes pertaining to software evolution, fundamental re-engineering techniques to modernize legacy systems including source code analysis, architecture recovery, and code restructuring, software refactoring strategies, migration to Object Oriented platforms, quality issues in re-engineering processes, migration to network-centric environments, and software integration, reverse engineering, program comprehension, source code transformation and refactoring strategies, software maintenance and re-engineering economics.

**Reference Materials:**

1. Re-engineering legacy software, David Lorge Parnas, Chris Birchall, Safari Books, Shelter Island, NY, 2016
2. Reengineering, Priyadarshi Tripathy and Kshirasagar Naik, John Wiley & Sons, Inc.2015
3. Software Maintenance and Evolution: a Roadmap, K.H.Bennett and V.T Rajlich, The Future of Software Engineering, ACM Press 2000.

Course Name: <b>Software Re-Engineering</b>	
Week #	Distribution
1.	Salient topics include the terminology
2.	the processes pertaining to software evolution
3.	fundamental re-engineering techniques to modernize legacy systems including source code analysis
4.	architecture recovery, and code restructuring
5.	software refactoring strategies



6.	migration to Object Oriented platforms, quality issues in re-engineering processes
7.	migration to network-centric environments
8.	software integration
9.	reverse engineering
10.	program comprehension, source code
11.	Transformation
12.	refactoring strategies
13.	software maintenance
14.	re-engineering economics.
15.	Presentation
16.	Wrap up Your Work and Presentation



**Course Name: Software Quality Engineering**

**Course Code: SE-358**

**Credit Hours: 3**

**Prerequisites: Software Engineering**

**Objectives and Outcomes:** At the end of the course the students will be able to:

1. Outline software testing and software quality assurance principles.
2. Prepare test case and test suites for completely testing all aspects of a system under test (SUT)
3. Analyze which of the software testing techniques are relevant for a particular case and know software reliability analysis tools and techniques.
4. Compile findings of a quality assurance cycle.

**Course Outline:** Software Quality, Software Quality Attributes, Quality Engineering., Testing: Concepts, Issues, and Techniques, Software testing lifecycle., Testing Scopes., Testing Approaches., Testing Concepts., Test Planning Process, Introduction to testing process, Requirement of software test planning, Testing documentation, Reporting and historical data recording., Software testing techniques, Testing philosophies , Testing strategies, Model based testing, Software testing techniques, Testing using models, Domain and combinatorial testing, Unit and integration testing, Acceptance testing, Test automation, Slicing, Software reliability models and engineering, Introduction, Exponential model., Reliability growth models, Modeling process, Software inspections, Software reviews, Inspection checks and metrics, Quality Models, Models for quality assessment, Product quality metrics, Quality Measurements, In-Process metrics for software testing, In-Process quality management, Effort/outcome models, System testing, Introduction to sub-system testing, From functional to system aspects of testing, System testing, Introduction to system testing, Scenarios development, System testing, Use-cases for testing, Specification-based testing, Open issues on software testing

Course Name: <b>Software Quality Engineering</b>	
Week #	Distribution
1.	Software Quality, Software Quality Attributes, Quality Engineering.,
2.	Testing: Concepts, Issues, and Techniques, Software testing lifecycle., Testing Scopes, Testing Approaches., Testing Concepts., Test Planning Process



3.	Introduction to testing process, Requirement of software test planning, Testing documentation,
4.	Reporting and historical data recording., Software testing techniques,
5.	Testing philosophies , Testing strategies, Model based testing,
6.	Software testing techniques, Testing using models, Domain and combinatorial testing
7.	Unit and integration testing, Acceptance testing, Test automation, Slicing
8.	Software reliability models and engineering, Introduction, Exponential model
9.	Reliability growth models, Modeling process, Software inspections
10.	Software reviews, Inspection checks and metrics, Quality Models
11.	In-Process metrics for software testing, In-Process quality management,
12.	Effort/outcome models, System testing,
13.	Introduction to sub-system testing, From functional to system aspects of testing
14.	System testing, Introduction to system
15.	Presentation
16.	Wrap up your work and Presentation

## Reference Materials:

1. Paul Jorgensen, Software Testing, A Craftsman's Approach, 4th Ed. CRC Press, Taylor and Francis Group, 2015
2. Bernard Homes, Fundamentals of Software Testing, ISTE, Wiley, 2012
3. Software Engineering, "Ian Sommerville, 9th Edition, Addison Wesley, 2011



**Course Name: Applied Physics**

**Course Code: PHY-101**

**Credit Hours: 3**

**Prerequisites: None**

**Objectives and Outcomes:** At the end of the course the students will be able to:

1. Describe the major concepts in physics.
2. Demonstrate an appropriate level of competency in both computer and research laboratory skills.
3. Formulate hypotheses and devise and perform experiments to test a hypothesis as individuals and in a team.
4. Effectively apply current technology and scientific methodologies for problem solving in various scientific, professional and community settings.
5. Effectively use and critically evaluate current technical/scientific research literature, online information, as well as information related to scientific issues in the mass media.
6. Integrate and relate scientific knowledge learned from classroom with real life situations.
7. Communicate in written and oral forms with interested citizens and professionals on key concepts in physics and general scientific issues.
8. Work cooperatively as part of a research team.
9. Maintain life-long learning in the sciences and incorporate new information into the existing body of knowledge.
10. Outline the applications of physics in industry and the role of physicists as entrepreneurs.

**Course Outline:** Electric force and its applications and related problems, conservation of charge, charge quantization, Electric fields due to point charge and lines of force. Ring of charge, Disk of charge, A point charge in an electric field, Dipole in an electric field, The flux of vector field, The flux of electric field, Gauss' Law, Application of Gauss' Law, Spherically symmetric charge distribution, A charge isolated conductor, Electric potential energy, Electric potentials, Calculating the potential from the field and related problem Potential due to point and continuous charge distribution, Potential due to dipole, equipotential surfaces, Calculating the field from the potential, Electric current, Current density, Resistance, Resistivity and conductivity, Ohm's law and its applications, The Hall effect, The magnetic force on a current, The Biot- Savart law, Line of B, Two parallel conductors, Amperes' s Law, Solenoid, Toroids, Faraday's experiments, Faraday's Law of



Induction, Lenz's law, Motional emf, Induced electric field, Induced electric fields, The basic equation of electromagnetism, Induced Magnetic field, The displacement current, Reflection and Refraction of light waves, Total internal reflection, Two source interference, Double Slit interference, related problems, Interference from thin films, Diffraction and the wave theory, related problems, Single-Slit Diffraction, related problems, Polarization of electromagnetic waves, Polarizing sheets, related problems.

### Reference Materials:

1. Fundamentals of Physics (Extended), 10th edition, Resnick and Walker
2. Narciso Garcia, Arthur Damask, Steven Schwarz., "Physics for Computer Science Students", Springer Verlag, 1998

Course Name:	
Week #	Distribution
1.	Electric force and its applications and related problems & conservation of charge, charge quantization Electric fields due to point charge and lines of force
2.	Ring of charge, Disk of charge A point charge in an electric field Dipole in a n electric field
3.	The flux of vector field The flux of electric field & Gauss' Law Application of Gauss' Law& Spherically symmetric charge distribution A charge isolated conductor
4.	Electric potential energy Electric potentials, Calculating the potential from the field
5.	Potential due to dipole, equipotential surfaces, Calculating the field from the potential &s Electric current &
6.	The Hall effect & The magnetic force on a current & The Biot- Savart law Line of B, Two parallel conductors & , Amperes' s Law, Solenoid, Toroids,
7.	The basic equation of electromagnetism, Induced Magnetic field, The displacement current, Reflection and Refraction of light waves,
8.	Total internal reflection, Two source interference, Double Slit interference, related problems



9.	Interference from thin films, Diffraction and the wave theory, related problems,
10.	Faraday's & , experiments, Faraday's Law of Induction, Lenz's law, Motional emf, Induced electric field, Induced electric fields,
11.	related problem Potential due to point and continuous charge distribution
12.	Current density, Resistance, Resistivity and conductivity
13.	Single-Slit Diffraction, related problems
14.	Polarization of electromagnetic waves
15.	polarizing sheets, related problems
16.	Wrap Up Your Work And Presentaion



**Course Name: Data Encryption and Security**

**Course Code:** CS-456

**Credit Hours:** 3

**Prerequisites:** None

**Objectives and Outcomes:** At the end of the course the students will be able to:

1. Identify computer and network security threats, classify the threats and develop a security model to prevent, detect and recover from the attacks. (ABET Outcomes: a, c, e, j, k)
2. Encrypt and decrypt messages using block ciphers, sign and verify messages using well known signature generation and verification algorithms. (ABET Outcomes: c, e, k)
3. Analyze existing authentication and key agreement protocols, identify the weaknesses of these protocols. (ABET Outcomes: c, e, k)
4. Download and install an e-mail and file security software, PGP, and efficiently use the code to encrypt and sign messages. (ABET Outcomes: c, e, k)
5. Develop SSL or Firewall based solutions against security threats, employ access control techniques to the existing computer platforms such as Unix and Windows NT. (ABET Outcomes: a, c, e, i, k)
6. Write an extensive analysis report on any existing security product or code, investigate the strong and weak points of the product or code. (ABET Outcomes: a, e, i, j, k, h)

**Course Outline:** Principle of number theory and probability theory, Primes, random numbers, modular arithmetic and discrete logarithms. Cryptographic algorithms and design principles, including conventional and symmetric encryption (DES, IDEA, Blowfish, Rijndael, RC- 4, RC-5), public key or asymmetric encryption (RSA, Diffie-Hellman), key management, hash functions (MD5, SHA-1, RIPEMD-160, HMAC), digital signatures and certificates. Authentication protocols (X.509, Kerberos), electronic mail security (S/MIME, PGP), web security and protocols for secure electronic commerce (IPSec, SSL, TLS, SET).

**Reference Materials:**

1. Cryptography and Network Security: Principles and Practice, William Stallings, 6th edition.



Course Name: Data Encryption and Security	
Week #	Distribution
1.	Principle of number theory and probability theory
2.	Primes
3.	random numbers
4.	modular arithmetic
5.	discrete logarithms
6.	Cryptographic algorithms
7.	design principles
8.	including conventional and symmetric encryption (DES, IDEA, Blowfish Rijndael, RC- 4, RC-5)
9.	public key or asymmetric encryption (RSA, Diffie-Hellman
10.	key management, hash functions (MD5, SHA-1, RIPEMD-160, HMAC)
11.	digital signatures and certificates
12.	Authentication protocols (X.509, Kerberos)
13.	electronic mail security (S/MIME, PGP)
14.	web security and protocols for secure electronic commerce (IPSec, SSL, TLS, SET).
15.	Presentaion
16.	Wrap up Your Work and Presentation



**Course Name: Micro Processor and Assemble Language**

**Credit Hours:** 3 (2+1)

**Prerequisites:** Digital Logic and Design

**Objectives:**

To learn the basic building blocks of computer system, their functions and the way these components interacts.

**Learning Outcome:** At the completion of this course, students will be able to know about:

- Microprocessor Bus Structure
- Addressing, Data and Control
- Arithmetic and Logic, Programmed Control, Stack and its operation
- Peripheral Control Interrupts, Assembler and Debugger
- Manipulate and translate machine and assembly code, Describe actions inside the processing chip.

**Course Outline:**

Microprocessor Bus Structure: Addressing, Data and Control, Introduction to Registers and Flags. Addressing Modes, Instruction sets including Data Movement, Arithmetic and Logic, Programmed Control, Stack and its operation. Peripheral Control Interrupts. Introduction to the Assembler and Debugger, Manipulate and translate machine and assembly code, Describe actions inside the processing chip.

**Reference Materials:**

3. *The Intel Microprocessor 8<sup>th</sup> ed*, Barry B Brey.
4. *Assembly Language for Intel-based Computers, 6<sup>th</sup> Ed* Irvine,  
<http://vig.prenhall.com/catalog/academic/product/0,1144,0132383101,00> The  
*8086/8088 Microprocessor 4<sup>th</sup> Edition* by Avtar Singh.

Course Name: Micro Processor and Assemble Language	
Week #	Distribution
1.	Microprocessor Bus Structure
2.	Addressing
3.	Data and Control
4.	Introduction to Registers and Flags



5.	Addressing Modes
6.	Instruction sets including Data Movement
7.	Arithmetic and Logic,
8.	Programmed Control
9.	Stack and its operation
10.	Peripheral Control Interrupts
11.	Introduction to the Assembler
12.	Introduction to the Debugger
13.	Manipulate and translate machine and assembly code
14.	Describe actions inside the processing chip
15.	Presentation
16.	Wrap up Your Work Presentation



## MS Computer Science

### Course Description

<b>Course Code</b>	CS 701
<b>Course Title</b>	Advanced Theory of Computation
<b>Credit Hours</b>	3
<b>Prerequisites by Course(s) and Topics</b>	Theory of Automata / Compiler Construction / Related Course
<b>Assessment Instruments</b> (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	Individual Assignments, Group Assignments, Quizzes, Class Tasks, Mid Term Exam, Final Exam
<b>Course Coordinator</b>	
<b>URL (if any)</b>	
<b>Recommended books and Reference Material</b>	<ol style="list-style-type: none"><li>1. Introduction to computer theory by Daniel I.A. Cohen, 2<sup>nd</sup> Edition. 2.</li><li>2. Theory of Automata, Formal Languages and Computation, By S. P. Eugene, Kavier, 2005, New Age Publishers, ISBN (10): 81-224-2334-5, ISBN (13) : 978-81-224-2334-1.</li><li>3. John Hopcroft and Jeffrey Ullman, Introduction to Automata Theory, Languages, and Computation, 2nd edition, 2001, Addison-Wesley.</li></ol>
<b>Course Objectives</b>	The course aims to develop an appreciation of the theoretical foundations of computer science through study of mathematical & abstract models of computers and the theory of formal languages. Theory of formal languages and use of various abstract machines as ‘recognizers’ and parsing will be studied for identifying/validating the synthetic characteristics of



	programming languages. Some of the abstract machines shall also study as 'Transducers'.
<b>Learning Outcomes</b>	<p>At successful completion of the course, students should:</p> <ul style="list-style-type: none"> <li>• Demonstrate advanced knowledge of formal computation and its relationship to languages</li> <li>• Distinguish different computing languages and classify their respective types</li> <li>• Recognize and comprehend formal reasoning about languages</li> <li>• Show a competent understanding of the basic concepts of complexity theory</li> </ul>
<b>Course Contents</b>	Languages, Regular expressions, finite automata, transition graph, Kleene theorem (I, II, III), Machines with output, CFG's and PDA. Turing machine, TM variations, decidability, reducibility, time and space hierarchy, computational complexity i.e. P, NP, NPC problems.
<b>Week wise Topics Covered in the Course</b>	<p><b>[Week 1-2] <u>Regular Languages</u>:</b> Basic Concepts, Types of Languages, Regular Expression, Finite Automata, DFA minimization, NFA, Transition Graph, General Transition Graph.</p> <p><b>[Week 3] <u>Finite Automata with Output</u>:</b> Finite Automaton with output, Moore machine, Mealy machine</p> <p>Equivalent machines, Complementing machine, incrementing machines, decrementing machines, converting Moore machine to mealy machine and vice versa.</p> <p><b>[Week 4] <u>Detection of Regularity of a Language</u>:</b> Regular languages, Complement of a language, No regular languages, Pumping Lemma, Pumping Lemma version II</p>



**[Week 5] Non Regular Languages:** Context Free Grammar (CFG), CFG terminologies, Trees, Total language tree, Chomsky Normal Form (CNF), Push Down Automata (PDA), PDA corresponding to CFG, Conversion form of PDA

**[Week 6] Turing Machine:** Introduction, Turing machine basic concepts, Turing machine variation.

**[Week 7-8] Turing Machine Variations:** Computation with TM, Combining TM, Multi-tape TM, Church TM, Nondeterministic TM, Random Access TM, Universal TM, Hilbert Problem, Enumerators.

**[Week 9] Decidability:** Decidable Languages, Decidable problems concerning regular languages and context-free languages, Halting Problem, Diagonalization method, Halting problem is undecidable, Turing-unrecognizable language.

**[Week 10] Reducibility:** Undecidable Problems from Language Theory, Undecidable Problem, Mapping Reducibility, Computable functions.

**[Week 11] Computability Theory:** Recursion Theorem, Self-reference, Decidability of logical theories, Turing Reducibility, Minimal length descriptions, Optimality of the definition, Incompressible strings and randomness.

**[Week 12] Time Complexity:** Complexity relationships among models, Class P, Class NP, Class NPC, Class NP hard,



	<p>Polynomial time reducibility, Cook-Levin Theorem, Vertex cover problem, Hamiltonian path problem, subset sum problem.</p> <p><b>[Week 13] <u>Space Complexity:</u></b> Savitch's Theorem, Space Class, PSPACE-completeness, TQBF problem, Classes L and NL, NL-completeness, NL equals coNL</p> <p><b>[Week 14] <u>Intractability:</u></b> Hierarchy Theorems, Exponential space completeness, Relativization, Limits of the diagonalization method, Circuit Complexity</p> <p><b>[Week 15] <u>Advanced Topics in Complexity Theory:</u></b> Approximation Algorithms, Probabilistic Algorithms, Alternation, Interactive Proof Systems, Parallel Computation, Graph nonisomorphism, Polynomial time hierarchy</p> <p><b>[Week 16] <u>Write-up Over Advance Topics and Presentations etc:</u></b> Presentations, Assignments and Case studies</p>
--	--



**Institute:** Department of Computer and Software Technology  
University of Swat

**Program (s):** BS Computer Science/ Software Engineering/ Information Technology

## Course Description

<b>Course Code</b>	MATH-103
<b>Course Title</b>	Mathematics-I
<b>Credit Hours</b>	3
<b>Prerequisites by Course(s) and Topics</b>	
<b>Assessment Instruments</b> (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	Individual Assignments, Group Assignments, Quizzes, Class Tasks, Mid Term Exam, Final Exam
<b>Course Coordinator</b>	
<b>URL (if any)</b>	
<b>Recommended books and Reference Material</b>	1. A Textbook of Mathematics for Grade XI by Prof Dr. Gulzar Ali Khan and Prof Dr. Islam Noor.
<b>Course Objectives</b>	The aim of this course is to enable students coming from pre-medical background to have grasp over basic concepts of intermediate level mathematics.
<b>Learning Outcomes</b>	
<b>Course Contents</b>	Complex Numbers, addition and subtraction of complex number. Multiplication and division of complex number, conjugate of complex number, Introduction to matrices, types of matrices, addition, subtraction, and multiplication of matrices, Determinants, determinant of 2-by-2 matrices, determinant of 3-by-3 matrices, adjoint, and inverse of matrices, solution of system of equation by Cramer's rule, Introduction to vector and scalar, types of vectors, addition, subtraction of vector, unit vector angle between two vector, Sequences, Miscellaneous Series, Permutation, even and odd permutation, Combination, Probability, Mathematical Induction and Binomial Theorem, Introduction to functions, domain and range, one to one function, onto function, many to one function, constant function, identity function even and odd function, Graphs of functions, Linear Programming, Trigonometric Identities, the law of cosines,



	Application of Trigonometry Solutions of Trigonometric Equations.
<b>Week wise Topics Covered in the Course</b>	<p>[Week 1] Complex Numbers, addition and subtraction of complex number.:</p> <p>[Week 2] Multiplication and division of complex number, conjugate of complex number,</p> <p>[Week 3] Introduction to matrices, types of matrices, addition ,subtraction, and multiplication of matrices:</p> <p>[Week 4] Determinants, determinant of 2-by-2 matrices, determinant of 3-by-3 matrices, adjoin, and inverse of matrices, solution of system of equation by Cramer's rule.</p> <p>[Week 5] Introduction to vector and scalar, types of vectors, addition, subtraction of vector, unit vector angel between two vector:</p> <p>[Week 6] Sequences, Miscellaneous Series:</p> <p>[Week 7] Sequences, Miscellaneous Series:</p> <p>[Week 8] Permutation, even and odd permutation:</p> <p>[Week 9] Combination, Probability:</p> <p>[Week 10-11] Mathematical Induction and Binomial Theorem:</p> <p>[Week 12] Introduction to functions, domain and range, one to one function, onto function, many to one function, constant function, identity function even and odd function:</p> <p>[Week 13] Graphs of functions:</p> <p>[Week 13-14] Linear Programming:</p> <p>[Week 15] Trigonometric Identities, the law of cosines, Application of Trigonometry:</p> <p>[Week 16] Solutions of Trigonometric Equations.:</p>



--	--



**Program (s):** BS - Computer Science/Software Engineering/Information Technology

## Course Description

<b>Course Code</b>	MATH-154
<b>Course Title</b>	Mathematics-II
<b>Credit Hours</b>	3
<b>Prerequisites by Course(s) and Topics</b>	
<b>Assessment Instruments</b> (homework, quizzes, midterms, final, programming assignments, lab work, etc.)	Individual Assignments, Group Assignments, Quizzes, Class Tasks, Mid Term Exam, Final Exam
<b>Course Coordinator</b>	
<b>URL (if any)</b>	
<b>Recommended books and Reference Material</b>	1. A Textbook of Mathematics for Grade XII by Prof. Mumtaz Khan.
<b>Course Objectives</b>	The aim of this course is to enable students coming from pre-medical background to have grasp over basic concepts of intermediate level mathematics.
<b>Learning Outcomes</b>	
<b>Course Contents</b>	Introduction to function, domain and range of a function types of functions, one to one function, many one function, onto function, constant function, identity function. exponential and logarithmic functions. Introduction to limits, left and right limit, one-sided limits, infinite limit, Differentiation implicate differentiation, logarithmic differentiation, derivatives of exponential and inverse trigonometric function, Higher Order Derivative and Applications, Differentiation of Vector Functions, Integrations, Integrations by parts, Indefinite and definite integral, integration by substitution, Integrations by partial fraction. Plane Analytic Geometry, Equation of line and plane, Distance between two points, angle between two line and angle between two plane. Conics section, parametric equations, tangent lines to parametric curves, polar coordinates, relation between polar and rectangular coordinates, Equation of parabola, hyperbola and ellipse, Introduction to Differential Equations, order and degree of differential equation, solution of differential equation by separable of variable,



	Introduction to Partial Differentiation, degree and order of partial differential equation, Numerical Methods, solution of one variable of equation by Newton's Rapson and Bisection method.
<b>Week wise Topics Covered in the Course</b>	<p>[Week 1] Introduction to function, domain and range of a function types of functions, one to one function, many one function, onto function, constant function, identity function. exponential and logarithmic functions:</p> <p>[Week 2] Introduction to limits, left and right limit, one-sided limits, infinite limit,</p> <p>[Week 3] Differentiation implicate differentiation, logarithmic differentiation:</p> <p>[Week 4] derivatives of exponential and inverse trigonometric function .</p> <p>[Week 5] Higher Order Derivative and Applications:</p> <p>[Week 6] Higher Order Derivative and Applications:</p> <p>[Week 7] Differentiation of Vector Functions:</p> <p>[Week 8] Integrations, Integrations by parts:</p> <p>[Week 9] Indefinite and definite integral, integration by substitution, Integrations by partial fraction.:</p> <p>[Week 10] Plane Analytic Geometry, Equation of line and plane:</p> <p>[Week 11] Distance between two points, angle between two line and angle between two plane:</p> <p>[Week 12] Conics section, parametric equations, tangent lines to parametric curves, polar coordinates, relation between polar and rectangular coordinates:</p> <p>[Week 13] Equation of parabola, hyperbola and ellipse:</p>



	<p>[Week 14] Introduction to Differential Equations, order and degree of differential equation, solution of differential equation by separable of variable:</p> <p>[Week 15] Introduction to Partial Differentiation, degree and order of partial differential equation:</p> <p>[Week 16] Numerical Methods, solution of one variable of equation by Newton's Rapson and Bisection method:</p>
--	--



## List of Examiners for MS and PhD

Examiner Name	Designation	Address
Dr. Azhar Rauf	Professor	University of Peshawar, Peshawar
Dr. Ibrar Ali Shah	Professor	Department of Computer Software Engineering, UET Mardan
Dr. Jamil Ahmad	Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Laiq Hassan	Professor	Department of Computer Systems Engineering, UET Peshawar
Dr. M. Daud Awan	Professor	Department of Computer Science, Preston University Islamabad
Dr. M. Waqar Aziz	Professor	Department of Computer Science, CECOS Peshawar
Dr. Saeed Mahfooz	Professor	University of Peshawar, Peshawar
Dr. Shah Khusro	Professor	University of Peshawar, Peshawar
Dr. Arif Ur Rahman	Associate Professor	Department of Computer Science, Bahria University Islamabad
Dr. Ayyaz Hussain	Associate Professor	Department of Computer Science, Quaid I Azam University, Islamabad
Dr. Faisal Bukhari	Associate Professor	University of Punjab, Lahore
Dr. M. Asif	Associate Professor	National Textile University Faisalabad
Dr. M. Muzammal	Associate Professor	Department of Computer Science, Bahria University Islamabad
Dr. M. Sajjad	Associate Professor	Department of Computer Science, Islamia College University Peshawar
Dr. M. Usman	Associate Professor	Department of Computer Software Engineering, UET Mardan
Dr. Muazzam A Khan Khattak	Associate Professor	Department of Computer Science, Quaid I Azam University, Islamabad
Dr. Muhammad Arshad	Associate Professor	City University of Science and Information Technology, Peshawar
Dr. Muhammad Sajjad	Associate Professor	Islamia College University Peshawar
Dr. Munib Gohar	Associate Professor	Department of Computer Science, Bahria University Islamabad
Dr. Onaiza Maqbool	Associate Professor	Department of Computer Science, Quaid I Azam University, Islamabad
Dr. Sara Shahzad	Associate Professor	University of Peshawar, Peshawar
Dr. Sehatullah	Associate Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Siffat Ullah Khan	Associate Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Waheed Iqbal	Associate Professor	University of Punjab, Lahore
Dr. Nadeem Iqbal	Associate Professor	Abdul Wali Khan University Mardan
Dr. Nadir Shah	Associate Professor	COMSATS Wah Cantt
Dr. Sajid Anwar	Associate Professor	IM Sciences, Peshawar



Dr. Farhan Riaz	Associate Professor	National University of Science and Technology, Islamabad
Dr. Muhammad Sadiq Khan	Assistant Professor	University of Chitral
Dr. Aftab Alam	Assistant Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Arif Shah	Assistant Professor	Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur
Dr. Arshad Ahmad	Assistant Professor	Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur
Dr. Bashir Hayat	Assistant Professor	Department of Computer Sciences, IM Sciences, Peshawar
Dr. Engr. Sami ur Rahman	Assistant Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Fakhre Alam	Assistant Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Fakhrud Din	Assistant Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Ghufuran Ullah	Assistant Professor	Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur
Dr. Iftikhar Alam	Assistant Professor	City University of Science and Information Technology, Peshawar
Dr. Khalid Haseeb	Assistant Professor	Department of Computer Science, Islamia College University Peshawar
Dr. M. Sohail Khan	Assistant Professor	Department of Computer Software Engineering, UET Mardan
Dr. Muhammad Faisal	Assistant Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Muhammad Zahid Khan	Assistant Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Mushtaq Ali	Assistant Professor	Hazara University, Mansehra
Dr. Nasir Rashid	Assistant Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Nizam Uddin	Assistant Professor	University of Chitral
Dr. Pervez Khan	Assistant Professor	Department of Computer Science & IT, University of Malakand, Chakdara Dir (L)
Dr. Rashid Naseem	Assistant Professor	Pak-Austria Fachhochschule: Institute of Applied Sciences and Technology, Haripur
Dr. Shahab Haider	Assistant Professor	City University of Science and Information Technology, Peshawar
Dr. Shams Ur Rahman	Assistant Professor	Department of Computer Science, UET Mardan



**Higher Education Commission  
National Computing Education Accreditation Council  
(NCEAC)**



NCEAC/HEC/General/3-20  
March 20, 2020

Vice Chancellor/Rector

**Subject: Decisions of NCEAC 36<sup>th</sup> General Council meeting**

Accreditation by NCEAC is mandatory for all universities and institutes offering Bachelor degree program(s) in computing or a related discipline. NCEAC accreditation is mandatory for all affiliated institutions of universities as well. NCEAC accredits Bachelor degree programs in the following areas:

1. Artificial Intelligence.
2. Bioinformatics
3. Computer Science
4. Cyber Security
5. Data Science
6. Information Systems
7. Information Technology, and
8. Software Engineering

The General Council of NCEAC, in its 36<sup>th</sup> meeting held on 13<sup>th</sup> March 2020, decided the following:

- ☒ *With immediate effect, NCEAC allows admission to students of Intermediate (Pre-Medical) in all Bachelor computing programs (CS, SE, IT, AI, DS, CySec). All such students must pass deficiency courses of Mathematics of 6 credit hours within one year of their regular studies. The deficiency courses should cover most of the relevant topics to bachelor degree in computing education from intermediate level mathematics.*
- ☒ *NCEAC will ONLY visit for accreditation those institutes, who are adhering to the approved admission criteria mentioned in revised HEC computing curriculum 2017 and with the recent approval it makes it mandatory for institutions to only grant admission to applicants who have 50% or above marks in the intermediate or equivalence exam with either mathematics as a subject or pre-medical as a discipline.*

NCEAC requires following information from Universities using this URL <https://nceac.org/Affiliation>

1. Names of all computing programs offered by the University (campus wise)
2. Total number of affiliated institutes (Name, Address, contact number, programs offered)

For guidance please contact NCEAC IT team on [amansha@hec.gov.pk](mailto:amansha@hec.gov.pk) or [hshafeez@hec.gov.pk](mailto:hshafeez@hec.gov.pk) contact No. (051) 9040 2772.

  
Dr Shoab A Khan