Institute of Computing

Introduction

The Institute of Computing (IoC), formerly known as Institute of Information Technology (IIT), is one of the pioneer and largest academic institute at Kohat University of Science and Technology (KUST). IoC offers various Degree, certification programmes and provide trainings and services to other departments at the campus. The degree programmes are in par with the HEC and other universities of international repute. Apart from offering regular academic programmes, the institute organizes various seminars, workshops and conferences on the latest topics and issues in computing on regular basis. The institute enjoys the support of an outstanding team of experienced, highly skilled and young faculty members. The existing faculty members are well qualified from reputed universities of Pakistan and abroad. The faculty provides expertise in various specialization areas such as IT Management, Security Technologies, Network Technologies, Software Technologies, Artificial Intelligence, Data Mining, Mobile Technologies, Distributed Systems and Theoretical Computer Science. The department also offers state-of-the art facilities that include auditorium, well equipped lecture rooms, multi-purpose Computing Labs, Final Year Projects Lab, Research & Development Cell, Huawei Lab, Cisco Lab and Web Development Cell. The class rooms and labs are equipped with latest teaching aids such as multimedia projectors, LEDs and OHPs for students and teachers.

Teaching Faculty:

The IoC is known because of its outstanding faculty where the Institute has attracted some high-quality faculty members during the last few years. The list of prominent faculty members is given below. The Institute also gets supports from other departments to teach specialization courses.

• Professor Fida Youns Khattak	Dean of the Faculty
• Dr. Shafiullah Khan	Director/ Associate Professors
• Dr. Hammad Hassan	Assistant Professors
• Dr. Asad Habib	Assistant Professors
• Dr. Jawad Ashraf	Assistant Professors
• Dr. Amjad Mehmood	Assistant Professors
• Dr. Muhammad Zeeshan	Assistant Professors
• Dr. Zeeshan Iqbal	Assistant Professors
• Dr. Abdul Shahid	Assistant Professors
• Dr. Saima Hassan	Assistant Professors
• Dr. Muhammad Irfan ud Din	Assistant Professors
• Dr. Shafaq Mussadiq	Assistant Professors
• Ms. Rabia Khan	Assistant Professors
• Dr. Muhammad Altaf Khan	Lecturer
• Mr. Muhammad Sohail	Lecturer
• Mr. Abdur Rehman	Lecturer
• Mr. Muhammad Abrar Khan	Lecturer
• Mr. Qadeem Khan	Lecturer
• Mr. Faisal Baseer	Lecturer
• Mr. Tahir Naeem	Lecturer
• Mr. Muhammad Ali Zeb	Lecturer
• Mr. Noor Mast	Lecturer

Mr. Muhammad Adnan	Lecturer
Mr. Ilyas Ahmad	Lecturer
Dr. Muhammad Muneer Umar	Lecturer
Mr. Sana Ullah Khan	Lecturer
Mr. Sajid Ullah	Lecturer
Mr. Muhammad Roman	Lecturer

Academic Programmes

The Institute offers the following undergraduate and graduate degree programmes:

- Undergraduate Degree Programmes (4 years)
  - BS in Computer Science (BSCS)
  - BS in Software Engineering (BSSE)
- 16 Year Master Degree (2 years Master after 14 years of education)
  - Master in Computer Science (MCS)
- Graduate Degree Programmes
  - PhD in Computer Science
  - MS in Computer Science (MSCS)
  - MS in Data Science (MSDS)
  - MS in Software Engineering (MSSE)

BS Computer Sciences

BS in Computer Sciences is a four-years bachelor degree programme. This programme is designed as per the national and international standards. Graduates of this programme are pursuing successful careers in academia, industry and R&D organizations in Pakistan and abroad. This programme is accredited by the National Computing Education Accredited Council (NCEAC), HEC.

Area of Specialization

The Institute offers various specializations during the four-years BSCS degree. The students can choose subjects from the specialty of their own interest. However, the Institute offers only those specializations where the total number of students is more than 10.

The curriculum of BS Computer Sciences is divided into various categories as shown in the below table:

Sr #	Category	Description	Credit Hours
1	University Requirements	Courses under this category are compulsory which are completed by all bachelor students at KUST.	28
2	Faculty Common Courses	Support area courses - Only required for students who are registered in Computer Science related degrees.	18
3	Core Course	Core courses related to the degree programme as per the HEC and KUST guidelines.	58

4	Technical Elective	In the senior year students are provided opportunity to choose area of specializations of their interest. The Institute offers various important latest courses for students so they can choose market oriented courses for their electives.	24
5	Senior Design Project/Electives	In order to train students to tackle a project relating the real world problems, senior design project is offered that is spread over the last two semesters.	6
	Total		**134**

BS Computer Science (Semester Plan)

The BSCS degree programme is divided into eight semesters as follows.

Semester wise plan

Code	Course Title	CrHr	Pre-Req	Code	Course Title	CrHr	Pre-Req
Year 1 Semester 1				**Year 1 Semester II**			
XXX***	UR-I	4(3+1)	None	MATH103	Discrete Mathematics	3(3+0)	None
CS102	Programming Fundamentals	4(3+1)	None	CS213	Object Oriented Programming	4(3+1)	CS102
XXX***	FR-I	3(3+0)	None	CS131	Digital Logic and Design	3(3+0)	None
XXX***	UR-II	3(3+0)	None	XXX***	UR-IV	3(3+0)	None
XXX***	UR-III	3(3+0)	None	XXX***	UR-V	3(3+0)	None
				XXX***	UR-VI	3(3+0)	None
Year 2 Semester III				**Year 2 Semester VI**			
CS211	Data Structure and Algorithm	4(3+1)	None	CS251	Software Engineering	3(3+0)	CS101
XXX***	FR-II	3(3+0)	None	CS241	Web Engineering	4(3+1)	CS102
XXX***	FR-III	3(3+0)	None	CS222	Database Management Systems	4(3+1)	CS211
XXX***	FR-IV	3(3+0)	None	CS212	Operating System Concepts	4(3+1)	None
XXX***	UR-VII	3(3+0)	None	XXX***	FR-V	3(3+0)	None
Year 3 Semester V				**Year 3 Semester VI**			
CS233	Computer Organization and Assembly Language	4(3+1)	CS131	CS372	Information Security	3(3+0)	None
CS311	Theory of Automata	3(3+0)	None	CS363	Artificial Intelligence	3(3+0)	None
CS371	Data Communication and Computer Networks	4(3+1)	CS101	CSxxx	CS Elective-II	3(3+0)	None
CSxxx	CS Elective-I	3(3+0)	None	CSxxx	CS Elective-III	3(3+0)	None
XXX***	UR-VIII	3(3+0)	None	XXX***	UR-IX	3(3+0)	None
				XXX***	FR-VI	3(3+0)	
				CS494 (Optional)	Industrial Training Internship	0	None
Year 4 Semester VII				**Year 4 Semester VIII**			
CS411	Design and Analysis of Algorithms	4(3+1)	CS211	CS443	Computer Graphics	4(3+1)	None
CSxxx	CS Elective-IV	3(3+0)	None	CSxxx	CS Elective-VII	3(3+0)	None
CSxxx	CS Elective-V	3(3+0)	None	CSxxx	CS Elective-VIII	3(3+0)	None
CSxxx	CS Elective-VI	3(3+0)	None	CS499	Final Year Project-II	3(0+3)	CS498
CS498	Final Year Project-I	3(0+3)	None				

Note: UR: University Requirement Courses, FR: Faculty Requirement Courses

A. University Requirement Courses (28 CrHr to be completed from this list)

These courses are related to general education category which are offered in various semesters as per the semester plan shown above. Some of the courses such as Islamic studies, Pakistan studies and English are compulsory and must be studied by all students. Students will be required to complete certain courses and credit hours from this list as per the approved plan of the respective programme.

Code	Title	CrHr	Pre-Requisite
ENG112	Study Skills	3(3+0)	None
ENG151	Functional English	2(3+0)	None
ENG253	Communication Skills	3(3+0)	ENG112
ENG334	Technical and Business Writing	3(3+0)	ENG253
RS101	Islamic Studies *	3(3+0)	None
PS101	Pakistan Studies *	3(3+0)	None
CS101	Introduction to Computing *	3(3+0)	None
ECON102	Fundamentals of Economics	3(3+0)	None
MS121	Principles of Management	3(3+0)	None
MS261	Financial Accounting	3(3+0)	None
MS251	Financial Management	3(3+0)	None
MS131	Human Resources Management	3(3+0)	None
MS331	Organizational Theory and Behavior	3(3+0)	None
MS311	Entrepreneurship	3(3+0)	None
SWS101	Introduction to Sociology	3(3+0)	None
SWS231	Social Psychology	3(3+0)	None
SWS215	Human Rights	3(3+0)	None
SWS201	Social Work and Human Behavior	3(3+0)	None
PHI101	Introduction to Logic	3(3+0)	None
BS252	Computer Applications in Finance	3(3+0)	None

*Compulsory (must be offered) and three courses of English language are also compulsory

B. Faculty Requirement Courses (18 CrHr to be completed from this list)

As this programme is offered by the Faculty of Physical and Numerical Sciences, therefore, there are certain courses which are mandatory to be offered to strengthen the fundamental scientific concepts of the students. Students will be required to complete 18 CrHr from the following list of Subjects.

Code	Title	CrHr	Pre-Requisite
STAT102	Probability and Statistics	3(3+0)	None
MATH101	Calculus – I	3(3+0)	None
PHY101	Introductory Mechanics	3(3+0)	None
MATH311	Linear Algebra	3(3+0)	None
MATH471	Mathematical Modeling	3(3+0)	None
MATH271	Ordinary Differential Equations	3(3+0)	None
MATH473	Operation Research	3(3+0)	None

C. Core Courses (58 CrHr):

The following courses are the core courses those are compulsory for every student registered in BSCS degree programme.

Code	Title	CrHr	Pre-Requisite
CS102	Programming Fundamentals	4(3+1)	None
CS213	Object Oriented Programming	4(3+1)	CS102
CS211	Data Structures and Algorithms	4(3+1)	None
MATH103	Discrete Mathematics	3(3+0)	None
CS212	Operating Systems Concepts	4(3+1)	CS211
CS222	Database Management Systems	4(3+1)	CS211
CS251	Software Engineering	3(3+0)	CS101
CS371	Data Communication and Computer Networks	4(3+1)	CS101
CS372	Information Security	3(3+0)	None
CS131	Digital Logic and Design	3(3+0)	None
CS311	Theory of Automata	3(3+0)	None
CS233	Computer Organization and Assembly Language	4(3+1)	CS131
CS411	Design and Analysis of Algorithms	4(3+1)	CS211
CS241	Web Engineering	4(3+1)	CS102
CS363	Artificial Intelligence	3(3+0)	None
CS443	Computer Graphics	4(3+1)	None

D. CS Technical Electives

An important aspect of the CS curriculum is the Technical Elective courses. Students are given choices towards the end of the programme to choose specialization of their own interest. Students are required to complete 24 CrHr from the following list of courses as per the guidance of the Institute.

***Note**: Students will be allowed to choose electives courses from BS in Computer Science, or other BS degree programmes approved by relevant statutory bodies/authorities of KUST. Furthermore, additional courses may be added to the below list after approval from the respective boards/authorities.*

Code	**Title**	**Cr Hrs**	**Pre-Requisite**
General Electives Courses			
CS316	Visual Programming	3(2+1)	CS213
CS413	Parallel and Distributed Computing	3(3+0)	None
CS416	Distributed Systems	3(3+0)	CS371
CS417	Mobile Application Development	3(2+1)	CS213
CS223	Management Information System	3(3+0)	None
CS324	e-Business	3(3+0)	None
CS421	Distributed Database Systems	3(3+0)	CS222
CS422	Big Data Concepts	3(3+0)	None
CS423	Data-warehousing and Data mining	3(3+0)	CS222
CS325	Data Science	3(3+0)	None
CS332	Computer Architecture	3(3+0)	None
CS431	Digital Signal Processing	3(3+0)	None
CS432	Microprocessor and Assembly Language	3(3+0)	None
CS433	Embedded Systems	3(3+0)	None
CS434	System Programming	3(3+0)	None
CS343	Multimedia Technologies	3(3+0)	None
CS344	Internet Concepts	3(3+0)	CS241

CS444	Digital Image Processing	3(3+0)	None
Software Engineering			
CS451	System Analysis and Design	3(3+0)	None
CS453	Object Oriented Analysis & Design	3(3+0)	CS251
CS454	Software Project Management	3(3+0)	CS251
CS455	Software Quality Assurance	3(3+0)	CS251
CS456	Usability Engineering	3(3+0)	CS251
CS457	Simulation and Modeling	3(3+0)	CS251
Artificial Intelligence			
CS364	Human Computer Interaction	3(3+0)	None
CS462	Artificial Neural Networks	3(3+0)	CS363
CS467	Machine Learning	3(3+0)	CS363
CS468	Information Retrieval	3(3+0)	CS363
CS469	Natural Language Engineering	3(3+0)	CS363
Networks & Communication			
CS471	Network Programming	3(2+1)	CS371
CS472	Network Protocols	3(3+0)	CS371
CS473	Wireless Networks	3(3+0)	CS371
CS474	Advanced Computer Networks	3(3+0)	CS371
CS477	Multi-hop Networks	3(3+0)	CS371
CS478	Telecommunication Systems	3(3+0)	CS371
CS476	Network Security	3(3+0)	CS371

E. Senior Design Project: (6 Cr-Hrs)

Senior design project is one of the important aspects the Computer Science programme. The project is spread over two semesters. Students are required to work on a real world problem under the supervision of a senior faculty members. Students can complete this segment in group form as well.

Code	Title	CrHr	Pre-Requisite
CS498	Senior Design Project – I	3(0+3)	None
CS499	Senior Design Project – II	3(0+3)	CS498

BS Software Engineering

Recently, IoC started BS in Software Engineering. It is a four-years bachelor level degree programme designed according to the national and international standards in Software Engineering. The curriculum of BS Software Engineering is divided into various categories as shown in the below table. This programme is registered with the National Computing Education Accredited Council (NCEAC), HEC. The NCEAC will visit soon to the Institute to formally inspect this progarmme for accreditation.

Sr #	Category	Description	Credit Hours
1	University Requirements	Courses under this category are compulsory which are completed by all bachelor students at KUST.	28
2	Faculty Common Courses	Support area courses - Only required for students who are registered in Computer Science related degrees.	12
3	Core Course	Core courses related to the degree programme as per the HEC and KUST guidelines.	67
4	Technical Elective	In the senior year students are provided opportunity to choose area of specializations of their interest. The Institute offers various important latest courses for students so they can choose market oriented courses for their electives.	21
5	Senior Design Project/Electives	In order to train students to tackle a project related real world problems, senior design project is offered that is spread over the last two semesters.	6
	Total		**134**

BS Software Engineering (Semester Plan)

The BSSE degree programme is divided into eight semesters as follows.

Semester wise plan

Code	Course Title	CrHr	Pre-Req	Code	Course Title	CrHr	Pre-Req
Year 1 Semester I				**Year 1 Semester II**			
XXX***	UR-I	4(3+1)	None	MATH103	Discrete Mathematics	3(3+0)	None
CS102	Programming Fundamentals	4(3+1)	None	CS213	Object Oriented Programming	4(3+1)	CS102
XXX***	FR-I	3(3+0)	None	SE101	Software Engineering	3(3+0)	None
XXX***	UR-II	3(3+0)	None	XXX***	UR-IV	3(3+0)	None
XXX***	UR-III	3(3+0)	None	XXX***	UR-V	3(3+0)	None
				XXX***	UR-VI	3(3+0)	None
Year 2 Semester III				**Year 2 Semester IV**			
CS211	Data Structure and Algorithm	4(3+1)	None	SExxx	SE Elective-I	3(3+0)	None
SE202	Software Requirement Engineering	3(3+0)	SE101	CS241	Web Engineering	4(3+1)	CS102
XXX***	FR-II	3(3+0)	None	CS222	Database Management Systems	4(3+1)	CS211

XXX***	FR-III	3(3+0)	None	CS212	Operating System Concepts	4(3+1)	None
XXX***	UR-VII	3(3+0)	None	CS364	Human Computer Interaction	3(3+0)	None
Year 3 Semester V				**Year 3 Semester VI**			
XXX***	FR-IV	3(3+0)	None	CS372	Information Security	3(3+0)	None
SE341	Software Quality Engineering	3(3+0)	SE101	CS315	Software Construction and Development	3(3+0)	None
CS371	Data Communication and Computer Networks	4(3+1)	CS101	SExxx	SE Elective-II	3(3+0)	None
XXX***	UR-VIII	3(3+0)	None	SExxx	SE Elective-II	3(3+0)	None
SE312	Formal Methods in Software Engineering	3(3+0)	None	XXX***	UR-IX	3(3+0)	None
				SE311	Software Design and Architecture	3(3+0)	SE101
				SE494 (Optional)	Industrial Training Internship	0	None
Year 4 Semester VI				**Year 4 Semester VII**			
SE422	Software Project Management	3(3+0)	None	SE411	Software Re-Engineering	3(3+0)	None
SExxx	CS Elective-IV	3(3+0)	None	SExxx	CS Elective-VII	3(3+0)	None
SExxx	CS Elective-V	3(3+0)	None	SExxx	CS Elective-VI	3(3+0)	None
SE441	Software Testing	3(3+0)	None	CS457	Simulation and Modeling	3(3+0)	None
SE498	Final Year Project-I	3(0+3)	None	SE499	Final Year Project-II	3(0+3)	CS498

UR: University Requirement Courses **FR: Faculty Requirement Courses**

A. University Requirement Courses (28 CrHr to be completed from this list)

These courses are related to general education category which are offered in various semesters as per the semester plan shown above. Some of the courses such as Islamic studies, Pakistan studies and English are compulsory and must be studied by all students. Students will be required to complete certain courses and credit hours from this list as per the approved plan of the respective programme.

Code	Title	CrHr	Pre-Requisite
ENG112	Study Skills	3(3+0)	None
ENG151	Functional English	2(3+0)	None
ENG253	Communication Skills	3(3+0)	ENG112
ENG334	Technical and Business Writing	3(3+0)	ENG253
RS101	Islamic Studies *	3(3+0)	None
PS101	Pakistan Studies *	3(3+0)	None
CS101	Introduction to Computing *	3(3+0)	None
ECON102	Fundamentals of Economics	3(3+0)	None
MS121	Principles of Management	3(3+0)	None
MS261	Financial Accounting	3(3+0)	None
MS251	Financial Management	3(3+0)	None
MS131	Human Resources Management	3(3+0)	None
MS331	Organizational Theory and Behavior	3(3+0)	None
MS311	Entrepreneurship	3(3+0)	None
SWS101	Introduction to Sociology	3(3+0)	None
SWS231	Social Psychology	3(3+0)	None

SWS215	Human Rights	3(3+0)	None
SWS201	Social Work and Human Behavior	3(3+0)	None
PH101	Introduction to Logic	3(3+0)	None
BS252	Computer Applications in Finance	3(3+0)	None

*Compulsory (must be offered) and three courses of English language are also compulsory

B. Faculty Requirement Courses

As this programme is offered by the Faculty of Physical and Numerical Sciences, therefore, there are certain courses which are mandatory to be offered to strengthen the fundamental scientific concepts of the students. Students will be required to complete 12 CrHr from the following list of Subjects.

Code	Title	CrHr	Pre-Requisite
STAT102	Probability and Statistics	3(3+0)	None
MATH101	Calculus – I	3(3+0)	None
PHY101	Introductory Mechanics	3(3+0)	None
MATH311	Linear Algebra	3(3+0)	None
CS401	Operation Research	3(3+0)	None

C. Core Courses (67 CrHr):

The following courses are the core courses. These are compulsory for every student registered in BSSE degree programme.

Code	Title	CrHr	Pre-Requisite
CS102	Programming Fundamentals	(3 3 4)	None
CS213	Object Oriented Programming	(3 3 4)	CS102
CS211	Data Structures and Algorithms	(3 3 4)	None
MATH103	Discrete Mathematics	3(3+0)	None
CS212	Operating Systems Concepts	(3 3 4)	CS211
CS222	Database Management Systems	(3 3 4)	CS211
SE101	Software Engineering	3(3+0)	None
CS371	Data Communication and Computer Networks	(3 3 4)	CS101
CS372	Information Security	3(3+0)	None
CS241	Web Engineering	(3 3 4)	CS102
SE202	Software Requirement Engineering	3(3+0)	SE101
SE341	Software Quality Engineering	3(3+0)	SE101
SE311	Software Design and Architecture	3(3+0)	SE101
SE422	Software Project Management	3(3+0)	None
CS364	Human Computer Interaction	3(3+0)	None
SE411	Software Re-Engineering	3(3+0)	None
SE315	Software Construction & Development	3(3+0)	None
SE312	Formal Methods in Software Engineering	3(3+0)	None
SE441	Software Testing	3(3+0)	None
CS457	Simulation and Modeling	3(3+0)	SE101

D. List of Technical Elective Courses

Technical Elective courses is an important aspect of this programme. Students are given choices towards the end of the programme to select specialization of their own interest. Students are required to complete 24 CrHr from the following list of courses as per the guidance of the Institute.

***Note**: Students will be allowed to choose electives courses from BS in Software Engineering, or other BS degree programmes approved by relevant statutory bodies/authorities of KUST. Furthermore, additional courses may be added to the below list after approval from the respective boards/authorities.*

Code	Title	CrHr	Pre-Requisite
CS417	Mobile Application Development	3(3+0)	CS213
CS311	Theory of Automata	3(3+0)	None
CS431	Digital Signal Processing	3(3+0)	None
CS444	Digital Image Processing	3(3+0)	None
CS421	Distributed Database Systems	3(3+0)	CS222
CS474	Advanced Computer Networks	3(3+0)	CS371
CS463	Artificial Neural Networks	3(3+0)	CS363
CS473	Wireless Networks	3(3+0)	None
CS433	Embedded Systems	3(3+0)	CS363
CS434	System Programming	3(3+0)	None
CS469	Natural Language Engineering	3(3+0)	None
CS343	Multimedia Technologies	3(3+0)	None
CS423	Data-warehousing and data mining	3(3+0)	CS222
CS413	Parallel and Distributed Computing	3(3+0)	None
CS467	Machine Learning	3(3+0)	CS363
CS325	Data Science	3(3+0)	None
CS422	Big Data Concepts	3(3+0)	None
SE201	Object Oriented Software Engineering	3(3+0)	None
SE313	Usability Engineering	3(3+0)	None
SE412	Aspect Oriented Software Engineering	3(3+0)	None
SE314	Software Metrics	3(3+0)	None
SE421	Game Development	3(3+0)	None
SE431	Software Process Engineering	3(3+0)	None
SE442	Software Verification and Validation	3(3+0)	None
CS331	Computer Architecture & Organization	3(3+0)	CS232

E. Senior Design Project: (6 CrHr)

Senior design project is one of the important aspects the Software Engineering degree programme. The project is spread over two semesters. Students are required to work on a real world problem under the supervision of a senior faculty member. Students can complete this segment in groups form as well.

Code	Title	CrHr	Pre-Requisite
SE498	Senior Design Project – I	3(0+3)	None
SE499	Senior Design Project – II	3(0+3)	SE498

Master of Computer Science (MCS) - 16 Years.

The Institute of Computing offers a two years' undergraduate degree programme under the title 'Master in Computer Science (MCS)'. This programme is offered for students who have obtained two years conventional BSc degree in the with relevant courses, i.e., Computer Science, Mathematics and Statistics. This degree programme is spread over four semesters. The year and semester wise details of MCS degree programme are given below.

Sr #	Category	Description	Credit Hours
1	General Education	Courses under this category are compulsory which are completed by all bachelor students at KUST.	9
2	Core Course	Core courses related to the degree programme as per the HEC and KUST guidelines.	51
3	Technical Elective	In the senior year students are provided opportunity to choose area of specializations of their interest. The Institute offers various important latest courses for students so they can choose market oriented courses for their electives.	6
4	Senior Design Project/Electives	In order to train students to tackle a project related real world problems, senior design project is offered in the last semester.	3
	Total		**69**

MCS (Semester Plan)

The whole degree programme is divided into four semesters as follows.

Code	Course Title	CrHr	Pre-Req	Code	Course Title	CrHr	Pre-Req
Year 1 Semester I				**Year 1 Semester II**			
CS102	Programming Fundamentals	4(3+1)	None	CS342	Web Designing	4(3+1)	None
XX***	UR-I	3(3+0)	None	CS113	Object Oriented Programming	4(3+1)	CS102
CS222	Database Management Systems	4(3+1)	None	CS233	Computer Organization & Assembly Language	4(3+1)	None
CS211	Data Structures & Algorithms	4(3+1)	None	CS363	Artificial Intelligence	3(3+0)	CS363
XX***	UR-II	3(3+0)	None	XX***	UR-III	3(3+0)	ENG231
Year 2 Semester III				**Year 2 Semester IV**			
CS212	Operating Systems Concepts	3(3+0)	CS211	CS***	CS Elective-I	3(3+0)	None
CS371	Data Communication & Networks	4(3+1)	None	CS***	CS Elective-II	3(3+0)	None
CS251	Software Engineering	3(3+0)	CS211	CS443	Computer Graphics	4(3+1)	None
CS311	Theory of Automata	3(3+0)	None	CS454	Software Project Management	3(3+0)	None
CS316	Visual Programming	4(3+1)	CS213	CS493	Final Year Project	3(0+3)	None

***UR: University Requirement Courses**

A. List of University Requirements

These courses are related to general education which are offered in various semester as per the semester plan shown below. Students will be required to complete 9 CrHr from this list.

Code	Title	CrHr	Pre-Requisite
ENG112	Study Skills	3(3+0)	None
ENG253	Communication Skills	3(3+0)	None
CS101	Introduction to Computing	(3 3 4)	None
MS121	Principles of Management	3(3+0)	None
MATH103	Discrete Mathematics	3(3+0)	None
MS131	Human Resources Management	3(3+0)	None
MS311	Entrepreneurship	3(3+0)	None
SWS101	Introduction to Sociology	3(3+0)	None
SWS231	Social Psychology	3(3+0)	None

B. List of Core Courses

The following courses are the core courses. These are compulsory for every student registered in MCS degree programme.

Code	**Title**	**CrHr**	**Pre-Requisite**
CS102	Programming Fundamentals	4(3+1)	None
CS211	Data Structures & Algorithms	4(3+1)	None
CS213	Object Oriented Programming	4(3+1)	CS102
CS212	Operating Systems Concepts	3(3+0)	CS111
CS233	Computer Organization & Assembly Language	4(3+1)	None
CS311	Theory of Automata	3(3+0)	CS113
CS342	Web Designing	4(3+1)	None
CS222	Database Management Systems	4(3+1)	None
CS316	Visual Programming	4(3+1)	CS213
CS251	Software Engineering	3(3+0)	None
CS371	Data Communication & Networks	4(3+1)	None
CS454	Software Project Management	3(3+0)	CS251
CS443	Computer Graphics	4(3+1)	None
CS363	Artificial Intelligence	3(3+0)	CS111

C. List of Technical Elective

Technical Elective courses are an important aspect of this programme. Students are given choices towards the end of the programme to select specialization of their own interest. Students are required to complete 6 CrHr from the following list of courses as per the guidance of the Institute. Note: Students can also choose Technical Elective courses from the BS Computer Science and BS Software Engineering degree programmes.

Code	**Title**	**Cr Hrs**	**Pre-Requisite**
CS413	Parallel and Distributed Computing	3(3+0)	None
CS416	Distributed Systems	3(3+0)	CS371
CS417	Mobile Application Development	3(2+1)	CS213
CS223	Management Information System	3(3+0)	None

CS324	e-Business	3(3+0)	None
CS421	Distributed Database Systems	3(3+0)	CS222
CS422	Big Data Concepts	3(3+0)	None
CS423	Data-warehousing and Data mining	3(3+0)	CS222
CS325	Data Science	3(3+0)	None
CS332	Computer Architecture	3(3+0)	None
CS431	Digital Signal Processing	3(3+0)	None
CS432	Microprocessor and Assembly Language	3(3+0)	None
CS433	Embedded Systems	3(3+0)	None
CS434	System Programming	3(3+0)	None
CS343	Multimedia Technologies	3(3+0)	None
CS344	Internet Concepts	3(3+0)	CS241
CS444	Digital Image Processing	3(3+0)	None
CS451	System Analysis and Design	3(3+0)	None
CS453	Object Oriented Analysis & Design	3(3+0)	CS251
CS455	Software Quality Assurance	3(3+0)	CS251
CS456	Usability Engineering	3(3+0)	CS251
CS457	Simulation and Modeling	3(3+0)	CS251
CS364	Human Computer Interaction	3(3+0)	None
CS462	Artificial Neural Networks	3(3+0)	CS363
CS467	Machine Learning	3(3+0)	CS363
CS468	Information Retrieval	3(3+0)	CS363
CS469	Natural Language Engineering	3(3+0)	CS363
CS471	Network Programming	3(2+1)	CS371
CS472	Network Protocols	3(3+0)	CS371
CS473	Wireless Networks	3(3+0)	CS371
CS474	Advanced Computer Networks	3(3+0)	CS371
CS477	Multi-hop Networks	3(3+0)	CS371
CS478	Telecommunication Systems	3(3+0)	CS371
CS476	Network Security	3(3+0)	CS371

D. Senior Design Project: (6 CrHr)

Senior design project is one of the important aspects in this programme. Students are required to work on a real world problem under the supervision of a senior faculty member. Students can complete this segment in groups form as well.

Senior Design Project: (3 CrHr)				
S.No	**Code**	**Title**	**Pre-Requisite**	**CrHr**
1	CS493	Project	None	3(0+3)

Master of Science (MS) Computer Science

MS in Computer Science is a two years graduate level degree programme. IoC is successfully running this degree programme since 2006 The focus of this programme is mainly on research. Students are encouraged to take part in the conferences and seminars arranged by KUST and other universities in the region. The graduates of this programme are well received by the job markets and research organizations.

Programme Structure:

The whole programme can be completed in two years that consists of four semesters with 30 CrHr. The below table shows the overall structure of the programme:

Category	Credit Hours	Description
Core Courses	12	Four core courses are compulsory. A list of core courses is designed based on latest trend in this discipline as per the HEC criteria which will be offered to students in the whole duration. Students will be required to follow the semester plan as given in this prospectus. In case there is any change in the plan, the Institute will inform all students about it.
Elective Courses	12	A number of specialization/major areas have been identified. Relevant courses for each specialization area are listed. Students will be required to complete their credit hours from the chosen area.
Thesis	06	Intensive research to be conducted in this course. The University encourages Master students to publish their research work at international forums.
Paper Publication/ Presentation	0	All students will be required to get published one research paper from their thesis in HEC recognized journal or give presentation in any national conference/workshop during the entire degree program.
Internal Presentation	0	All students will be required to give internal presentation in their research group during the entire degree program.
Total	30	MPhil degree is awarded after successfully completion of all of the above requirements.

MSCS (Semester Plan)

The MSCS degree programme is divided into four semesters as follows.

√	Course Title	CrHr	Pre-Req	Code	Course Title	CrHr	Pre-Req
Year 1 Semester I				**Year 1 Semester II**			
CS515	Theory of Computations	3(3+0)	None	CS516	Advance Analysis of Algorithms	3(3+0)	--
CS517	Advanced Operating Systems	3(3+0)	None	CS531	Advanced Computer Architecture	3(3+0)	--
CS***	CS Elective – I	3(3+0)	None	CS***	CS Elective – II	3(3+0)	--
Year 2 Semester III				**Year 2 Semester IV**			
CS***	CS Elective – III	3(3+0)	--	CS699 Contd:	MS Research	-	6(0+6)
CS***	CS Elective – IV	3(3+0)	--				
CS699	MS Research	6(0+6)	--				

Core Courses (12 CrHr):
The following courses are the core courses which are compulsory for every student to registered in MSCS degree program. Students are required to study all these courses.

Code	Title	Pre-Requisite	CrHr
CS515	Theory of Computations	None	3(3+0)
CS516	Advanced Analysis of Algorithms	None	3(3+0)
CS517	Advanced Operating Systems	None	3(3+0)
CS531	Advanced Computer Architecture	None	3(3+0)

Elective (Student will be required to take – 4 courses) 12 CrHr

Code	Title	Pre-Requisite	CrHr
CS505	Research Methods with an IT Perspective	None	3(3+0)
CS611	Advance Parallel Programming	None	3(3+0)
CS612	Advance Distributed Systems	None	3(3+0)
CS613	Analysis and Design of Parallel Algorithms	None	3(3+0)
CS614	Algorithms for Bio-Informatics	None	3(3+0)
CS622	Advance Data Mining	None	3(3+0)
CS623	Big Data	None	3(3+0)
CS541	Multimedia Computing and Applications	None	3(3+0)
CS542	Computer Vision	None	3(3+0)
CS643	Semantic Web	None	3(3+0)
CS644	Data Compression	None	3(3+0)
CS645	Web Metrics	None	3(3+0)
CS646	Web Usability	None	3(3+0)
CS554	Requirement Engineering	None	3(3+0)
CS555	Software System Architecture	None	3(3+0)
CS556	Advance Software Quality Assurance	None	3(3+0)

CS657	Advance Software Project Management	None	3(3+0)
CS658	Software Testing	None	3(3+0)
CS563	Soft Computing	None	3(3+0)
CS564	Evolutionary Computing	None	3(3+0)
CS565	Advance Natural Language Engineering	None	3(3+0)
CS566	Design and Development of Corpora	None	3(3+0)
CS567	Information Architecture for the World Wide Web	None	3(3+0)
CS568	Advance Human Computer Interaction	None	3(3+0)
CS569	Information Authoring	None	3(3+0)
CS661	Advance Machine Learning	None	3(3+0)
CS662	Advance Artificial Neural Networks	None	3(3+0)
CS665	Development of Natural Language Engineering Resources	None	3(3+0)
CS666	Optimized Input Methods	None	3(3+0)
CS667	Machine Translation	None	3(3+0)
CS668	Advance Information Retrieval	None	3(3+0)
CS571	Advance Computer Communication & Networks	None	3(3+0)
CS576	Advance Network Security	None	3(3+0)
CS573	Wireless and Mobile Networks	None	3(3+0)
CS673	Advance Network Programming	None	3(3+0)
CS676	Network Management	None	3(3+0)
CS677	Advance Information Security	None	3(3+0)
CS678	Networks Middle Ware Design	None	3(3+0)

Thesis/Research: (6 CrHr)

Code	Title	Pre-Req	Cr Hrs
CS699	MS Research Report	None	6(0+6)

Master of Science in Data Science (MSDS)

Big Data Analytics and pervasive computing hinge on the principle axis of data analytics. MSDS programme is going to be relevant in terms of job creation and artisanal smart business generation.

IoC started the graduate level MSDS degree programme in 2018. It is designed to give students the option to be a part of the data science endeavor. We believe our MSDS graduates would benefit from the early-bird advantage in this career oriented and promising field.

The objectives of this programme are enlisted below.

- To equip students to transform data into actionable insights to make complex organizational decisions.
- To enable students, understand and analyze a problem and to arrive at computable solutions.
- To expose students to the set of technologies that match those solutions.
- To gain hands-on experience on data-centric tools for statistical analysis, visualization and big data applications.
- To understand the implications of handling data in terms of data security and business ethics.

Structure of the MSDS program

The program is spread over four semesters according to the following structure. The 6-credit hour thesis is mandatory.

Category	Credit Hours	Description
Core Courses	09	Three core courses are compulsory. A list of core courses is designed based on latest trend in this discipline as per the HEC criteria which will be offered to students in the whole duration. Students will be required to follow the semester plan as given in this prospectus. In case there is any change in the plan, the Institute will inform all students about it.
Elective Courses	15	A number of specialization/major areas have been identified. Relevant courses for each specialization area are listed. Students will be required to complete their credit hours from the chosen area.
Thesis	06	Intensive research to be conducted in this course. The University encourages Master students to publish their research work at international forums.
Paper Publication/ Presentation	0	All students will be required to get published one research paper from their thesis in HEC recognized journal or give presentation in any national conference/workshop during the entire degree program.

Internal Presentation	0	All students will be required to give internal presentation in their research group during the entire degree program.
Total	30	MPhil degree is awarded after successfully completion of all of the above requirements.

MSDS (Semester Plan)

The MSDS degree programme is divided into four semesters as follows.

Code	Course Title	CrHr	Pre-Req	Code	Course Title	CrHr	Pre-Req
Year 1 Semester I				**Year 1 Semester II**			
DS501	Tools and Techniques for Data Science	3(3+0)	None	CS661	Advance Machine Learning	3(3+0)	None
DS502	Statistical and Mathematical Methods for Data Analysis	3(3+0)	3(3+0)	DS***	DS Elective – II	3(3+0)	None
DS***	DS Elective – I	3(3+0)	3(3+0)	DS***	DS Elective – III	3(3+0)	None
Year 2 Semester III				**Year 2 Semester IV**			
CS***	CS Elective – III	3(3+0)	--	DS699 Contd:	MS Research	-	6(0+6)
CS***	CS Elective – IV	3(3+0)	--				
DS699	MS Research	6(0+6)	--				

Core Courses (09 CrHr):
The following courses are the core courses which are compulsory for every student to registered in MSDS degree programme. Students are required to study all these courses.

Code	Title	Pre-Requisite	CrHr
DS501	Tools and Techniques for Data Science	None	3(3+0)
DS502	Statistical and Mathematical Methods for Data Analysis	None	3(3+0)
CS661	Advance Machine Learning	None	3(3+0)

Elective (Student will be required to take – 5 courses) 15 CrHr

Code	Title	Pre-Requisite	CrHr
CS542	Computer Vision	None	3(3+0)
CS614	Algorithms for Bio-Informatics	None	3(3+0)
CS565	Advance Natural Language Engineering	None	3(3+0)
DS503	Bayesian Data Analysis	None	3(3+0)
DS504	Data Visualization	None	3(3+0)
DS505	Deep Learning	None	3(3+0)
DS506	Optimization Methods for Data Science and Machine Learning	None	3(3+0)
DS604	Deep Reinforcement Learning	None	3(3+0)
DS602	Probabilistic Graphical Models	None	3(3+0)
DS603	Social network analysis	None	3(3+0)
DS604	Time series Analysis and Prediction	None	3(3+0)
DS511	Algorithmic trading	None	3(3+0)
DS512	Cloud computing	None	3(3+0)
DS513	Distributed Data Processing and Machine Learning	None	3(3+0)

DS611	Distributed Machine Learning in Apache Spark	None	3(3+0)
DS612	High performance computing	None	3(3+0)
DS521	Big Data Analytics	None	3(3+0)
DS522	Computational Genomics	None	3(3+0)
DS621	Inference & Representation	None	3(3+0)
DS622	Scientific Computing in Finance	None	3(3+0)

Thesis/Research: (6 CrHr)

Code	Title	Pre-Requisite	CrHr
DS699	MS Research Report	None	6(0+6)

Master of Science in Software Engineering MSSE

The Master of Science in Software Engineering (MSSE) degree programme is designed to equip our students with theoretical as well as applied knowledge of software life cycle for solution of complex problems. It is aimed to prepare the students to learn independently in a constantly changing and challenging discipline.

Program Objectives

The objectives of MS (Software Engineering) degree programme are enlisted below.

1. Prepare students who can critically apply concepts, theories and practices to provide creative solutions of complex computing problems.
2. Prepare students who can define, plan, implement and test a medium-sized software project using appropriate software engineering processes, methods and techniques.
3. Prepare students to effectively communicate their ideas in written and electronic form, and prepare them to work collaboratively in a team environment.
4. Prepare students with a theoretical software engineering background and applied research needed to enter a doctorate program in software engineering.

Structure of the MSSE program

The program is spread over four semesters according to the following structure. The 6-credit hour thesis is mandatory.

Category	Credit Hours	Description
Core Courses	09	Three core courses are compulsory. A list of core courses is designed based on latest trend in this discipline as per the HEC criteria which will be offered to students in the whole duration. Students will be required to follow the semester plan as given in this prospectus. In case there is any change in the plan, the Institute will inform all students about it.
Domain Elective Courses	06	A number of specialization/major areas have been identified. Relevant courses for each specialization area are listed.
General Elective Courses	09	Students will be required to complete their credit hours from the chosen area.
Thesis	06	Intensive research to be conducted in this course. The University encourages Master students to publish their research work at international forums.
Paper Publication/ Presentation	0	All students will be required to get published one research paper from their thesis in HEC recognized journal or give presentation in any national conference/workshop during the entire degree program.

Internal Presentation	0	All students will be required to give internal presentation in their research group during the entire degree program.
Total	30	MPhil degree is awarded after successfully completion of all of the above requirements.

MSSE (Semester Plan)

The MSSE degree programme is divided into four semesters as follows.

Code	Course Title	CrHr	Pre-Req	Code	Course Title	CrHr	Pre-Req
Year 1 Semester I				**Year 1 Semester II**			
SE501	Advanced Requirements Engineering	3(3+0)	None	SE541	Software Testing and Quality Assurance	3(3+0)	--
SE511	Advanced Software System Architecture	3(3+0)	None	SE***	Domain Elective – II	3(3+0)	--
SE***	Domain Elective – I	3(3+0)	None	SE***	General Elective – I	3(3+0)	--
Year 2 Semester III				**Year 2 Semester IV**			
CS***	CS Elective – II	3(3+0)	--	SE699 Contd:	MS Research	-	6(0+6)
CS***	CS Elective – III	3(3+0)	--				
SE699	MS Research	6(0+6)	--				

Core Courses (09 CrHr):

The following courses are the core courses which are compulsory for every student to registered in MSSE degree programme. Students are required to study all these courses.

Code	Title	Pre-Requisite	CrHr
SE501	Advanced Requirements Engineering	SE501	3(3+0)
SE511	Advanced Software System Architecture	SE511	3(3+0)
SE541	Software Testing and Quality Assurance	SE541	3(3+0)

Domain Specific Technical Elective Courses (Student will be required to take –2 courses) 06 CrHr

Code	Title	Pre-Requisite	CrHr
SE512	Software Measurement and Metrics	None	3(3+0)
SE631	Component Based Software Engineering	None	3(3+0)
SE513	Advanced Formal Methods	None	3(3+0)
CS568	Advance Human Computer Interaction	None	3(3+0)
SE514	Agile Software Development Methods	None	3(3+0)
SE515	Empirical Software Engineering	None	3(3+0)
SE521	Advanced Software Project Management	None	3(3+0)

General Technical Elective Courses (Student will be required to take –3 courses) 09 CrHr

Code	Title	Pre-Requisite	CrHr
SE621	Software Risk Management	None	3(3+0)
CS601	Research Methodology	None	3(3+0)
SE622	Software Configuration Management	None	3(3+0)
SE641	Reliability Engineering	None	3(3+0)

SE632	Complex Networks	None	3(3+0)
SE611	Agent Based Modeling	None	3(3+0)

Thesis/Research: (6 CrHr)

Code	Title	Pre-Requisite	CrHr
SE699	MS Research Report	None	6(0+6)

PhD Computer Science programme

IoC offers a very strong PhD programme in various areas of computer sciences. This section briefly discusses the structure and courses of the PhD programme.

The IoC expects that all the students admitted to PhD studies must perform well during their enrollment as PhD students. In order to maintain a good standing, all students are required to perform satisfactorily during various stages such as course work, comprehensive examination and thesis proposal defense. Finally, admission to candidacy requires the students to demonstrate their evidence of research ability in the thesis proposal defense conducted by the university. The key programme activities are expected to be completed as per HEC guidelines for PhD degree programme.

For detailed information, the candidates for admission in PhD programme are required to visit the official website and contact admission office for the latest rules and regulations regarding PhD programme.

Programme Structure

Students will be required to complete 18 credit hours course work as per the HEC guidelines.
The course work will be followed by Research Thesis.

Category	**Credit Hours**	**Description**
Elective Courses	18	A number of specialization/major areas have been identified. Relevant courses for each specialization area are listed. Students will be required to complete their credit hours from the chosen area. 9 credit hours should be from specialization.
Non-Credit Hour	0	A PhD student must pass two non-credit courses (6 Credit hours) selecting from offered courses for under graduate/graduate programme in the university.
Comprehensive Examination	NA	A PhD student will be required to pass comprehensive examination after successful completion of course work and before submitting research proposal.
Paper Publication	NA	A PhD student will be required to get published one research paper from his/her thesis in HEC recognized journal during the entire degree program.
Presentation	NA	A PhD student will be required to publish and present one research paper in national/international conference.
Thesis	06	Intensive research to be conducted in this course. The University encourages PhD students to publish their research work at international forums.
Total	24	PhD degree is awarded after successfully completion of all of the above requirements.

PhD Computer Science (Semester Plan)

The PhD degree programme is divided into six semesters as follows.

Code	Course Title	CrHr	Pre-Req	Code	Course Title	CrHr	Pre-Req
Year 1 Semester I				**Year 1 Semester II**			
CS***	CS Elective – I	3(3+0)	None	CS***	CS Elective – III	3(3+0)	None
CS***	CS Elective – II	3(3+0)	None	CS***	CS Elective – IV	3(3+0)	None
XXX***	Non Credit Course - I	0(3+0)	None	XXX***	Non Credit Course - II	0(3+0)	None
Year 2 Semester III				**Year 2 Semester IV**			
CS***	CS Elective – V	3(3+0)	None	CS991	Thesis	6(0+6)	None
CS***	CS Elective – VI	3(3+0)	None				
				Comprehensive Examination			
Year 3 Semester V				**Year 3 Semester VI**			
CS999	Thesis	6(0+6)	None	CS999 Continued	Thesis	6(0+6)	None

Elective (Student will be required to take – 6 courses) 18 CrHr

Code	Title	Pre-Requisite	CrHr
CS711	Cluster and Grid Computing	None	3(3+0)
CS712	High Performance Computing	None	3(3+0)
CS741	Advance Topics in Semantic Web	None	3(3+0)
CS764	User Interface Design in Global Perspectives	None	3(3+0)
CS766	Knowledge Representation	None	3(3+0)
CS767	Computational Linguistics	None	3(3+0)
CS768	Statistical Natural Language Engineering	None	3(3+0)
CS769	Knowledge Based System Design	None	3(3+0)
CS765	Parsing Technologies	None	3(3+0)
CS862	Information Foraging	None	3(3+0)
CS863	Recommender Systems	None	3(3+0)
CS866	Computational Intelligence	None	3(3+0)
CS867	Speech Processing Techniques	None	3(3+0)
CS868	Advance Information Retrieval	None	3(3+0)
CS771	Mobile Adhoc Networks	None	3(3+0)
CS772	Wireless Mesh Networks	None	3(3+0)
CS776	Wireless Sensor Networks	None	3(3+0)
CS777	Advance Networking	None	3(3+0)
CS778	Advance Wireless Network Security	None	3(3+0)
CS779	Advance Wireless Networks	None	3(3+0)
CS875	Mobile Communication Systems	None	3(3+0)
CS876	Information and Coding Theory	None	3(3+0)
CS877	Traffic Control and Quality of Services	None	3(3+0)

Thesis/Research: (6 CrHr)

Code	Title	Pre-Requisite	CrHr
CS999	Thesis	None	6(0+6)

Undergraduate Core Courses
(Description and Contents)

Course Code	**CS102**
Course Title	Programming Fundamentals
Cr Hrs	4 (3+1)
Pre-requisite	Nil
Recommended Texts	1. Starting Out with C++ from Control Structures to Objects, Tony Gaddis, 9th Edition, 2017, ISBN-13: 978-0134498379, ISBN-10: 0134498372 2. The C++ Programming Language, Bjarne Stroustrup, Edition 4, Addison-Wesley, 2013, ISBN: 0133522857, 9780133522853. 3. C++ programming cookbook Herb Schildt's C++ programming cookbook / C++ (Computer program language) , Schildt, Herbert. To produce programmers equipped with an understanding of fundamental computational concepts underlying most programming languages. 4. The role of programming within the overall software development and appropriate attitudes and working practices for a professional programmer and skills supporting. 5. The solution of small problems using a programming language &clear expression of solutions at different levels of abstraction.McGraw-Hill, New York: c2008. ISBN-9780071488600
Course Description	As part of this course, students will be introduced to programming concepts and techniques. They will analyze and design programs using primitive statements of C/C++ for a wide variety of problems in math, science, financials, and games.
Course Objectives	• To produce programmers equipped with an understanding of fundamental computational concepts underlying most programming languages. • The role of programming within the overall software development and appropriate attitudes and working practices for a professional programmer and skills supporting. • The solution of small problems using a programming language &clear expression of solutions at different levels of abstraction.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction to Programming Concepts Higher-level language Vs low-level language Vs Object code Role of compiler, interpreter and assembler
W2	An example C++ program and its flow of control Preliminary Remarks about Program Style

W3	Understanding Data type and constant Variables and rules of declaration
W4	Working with arithmetic expression Identifiers, Integers, Real numbers, Type Casting, ,Characters
W5	Combing operation with assignment operator Boolean Expressions and Operators
W6	Basic C++ language statements and functions Boolean Expressions and Operators
W7	If and if else statements with examples Multiple Selection and Switch statement
W8	Conditional operator AND, OR etc Loop Statements FOR loop
W9	While loop with examples Do while loop with examples and difference between do and while loop. Arrays and Strings: The Basic Idea and Notation
W10	Declaring an array and initializing array Assignment Statements and Expressions with Array Elements. Arrays as Parameters in Functions.
W11	Sorting Arrays with examples Two-dimensional Arrays Strings, The Sentinel String Character '\0'
W12	Functions and Procedural Abstraction: The Need for Sub-programs, User-defined Functions with examples
W13	Value and Reference Parameters Recursion: The Basic Idea, A Simple Example, The Mechanics of a Recursive Call, Recursion and Iteration
W14	Introducing Pointers, Declaring Pointers, Pointers and memory address, Assignments with Pointers Using the Operators "*" and "&"
W15	Dynamic Arrays, Local Vs global variables ,Automatic and Dynamic Variables
W16	Basic Sorting Algorithms

Course Code	**CS213**
Course Title	**Object Oriented Programming**
Cr Hrs	4 (3+1)
Pre-requisite	CS102 (Programming Fundamentals)
Recommended Texts	1. Object-oriented programming in C++ ,Tatyana Sopronyuk , NonnaShulga , CreateSpace Independent Publishing Platform; 1 edition (September 26, 2014), ISBN: 978-1502520906 2. Microsoft Visual C# 2015: An Introduction to Object-Oriented Programming 6th Edition , Joyce FarrellCengage Learning Publisher ;6 edition (June 3, 2015), , ISBN: 978-1285860237. 3. C++ How To Program: Dietel&Dietel, Harvey & Paul, Prentice Hall, 10th Edition, 2017, ISBN-13: 9780134448848.
Course Description	This course gives you a thorough grounding in the basics of Object Orientation i.e. Abstraction, Polymorphism, Inheritance, Encapsulation and other related concepts. Students will learn about these concepts in a

	C++ development environment. It will also give you knowledge about C++ Class and Template Constructs.
Course Objectives	• The course aims to focus on OOP concepts, analysis and software development.

Week Wise Distribution of the Contents

Lecture No	**Topic**
W1	Basic Concepts Introduction to OOP, History Introduction to basic features (Classes, objects, inheritance, polymorphism)
W2	Overloading, data hiding / encapsulation
W3	Class Example Declaring class, Member functions and data Using class
W4	Constructor, Destructor Overloaded constructor, default copy constructor
W5	Overloading unary operators, Overloading binary operators
W6	Arithmetic operators, concatenating strings Comparison operators, arithmetic assignment operator
W7	Data type conversion Conversion among basic data types and objects to basic data types Derived class and base class
W8	Derived class constructor Overriding member functions
W9	Multiple inheritance
W10	Address of operator Indirection operator Pointers and arrays
W11	Pointers and functions Function Declaration, Definition, Calling
W12	Passing arguments to functions Inline functions Function overloading
W13	Virtual functions Early and late binding
W14	Abstract classes and pure virtual functions Friend functions
W15	Function Overriding
W16	Practical projects including complete oop concept

Course Code	**CS211**
Course Title	**Data Structures & Algorithms**
Cr Hrs	4 (3+1)
Pre-requisite	None

Recommended Texts	1. Data Structures and Algorithms Made Easy: Data Structures and Algorithmic Puzzles, Fifth Edition Paperback, Narasimha Karumanchi, Career Monk Plublications; 5 edition (August 28, 2016), ISBN: 978-8193245279 2. Data Structures and Algorithms in Java 6th Edition by Michael T. Goodrich ,Roberto Tamassia , Michael H. Goldwasser , Wiley 6 edition (January 28, 2014),ISBN: 978-1118771334 3. Data Structures and Algorithms in C++, Adam Drozdek, 2012, ISBN-13: 978-1133608424
Course Description	The course will cover well-known data structures such as dynamic arrays, linked lists, stacks, queues, tree, heap, disjoint sets and table
Course Objectives	Three goals will be accomplished: • Implement these structures in C++ • Determine which structures are appropriate in various situations • Confidently learn new structures beyond what's presented in this class.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Definitions (Data, Entity, information, Data types, Built in Data types, ADT, pointers in C), Concept of Data Structure, Overview of Data Structure Algorithm: Simple and Complex, Components of an Algorithm: introduction, Start/End, Statements: Executable and non-executable
W2	Flowcharts, Control Structure(Sequential, Conditional, Loops)&Linear DS and Non Linear DS
W3	Function Sub Algorithm, Procedure Sub Algorithm **ARRAYS** Definition, One dimensional Array, Memory Allocation for an Array Traversing in a Linear Array, Searching in a Linear Array
W4	Insertion in a Linear Array, Deletion in a Linear Array Two Dimensional Arrays, Sparse Matrices, Pointer Arrays Linear Search, Binary Search
W5	**LINKED LISTS** Definition, Single Linked List, Operation on a Single Linked List Creating a Single Linked List(Algorithm), Accessing a Single Linked List(Algorithm)
W6	Insertion into a Single Linked List(Algorithm) Deletion into a Single Linked List(Algorithm) Searching into a Single Linked List(Algorithm)
W7	Circular Linked List: Insertion into a Circular Linked List(Algorithm), Deletion into a Circular Linked List(Algorithm), Searching into a Circular Linked List(Algorithm) Double Linked List: Accessing a Double Linked List(Algorithm), Insertion into a Double Linked List(Algorithm), Deletion into a Double Linked List(Algorithm)
W8	**STACK** Introduction, Definition, Representation of stack Operation on Stack PUSH and POP (Algorithms) Converting an infix notation to postfix (Algorithm) and example

W9	Evaluation of Arithmetic Expression from postfix notation(Algorithm) and example **SORTING** Bubble Sort (Algorithm) Quick Sort (Algorithm)
W10	Insertion Sort(Algorithm) Selection Sort(Algorithm) **QUEUES** Introduction, Definition, Representation of Queues
W11	Insertion and Deletion in Queue Deque, Priority Queue **TREES** Binary Tree Terminologies
W12	Representation of Binary Tree: Using Linked List, Sequential Representation Operation On Binary Tree, Preorder Inorder Traversing
W13	Post Order Traversing Insertion, Deletion, Searching Constructing binary tree from arithmetic Expressions
W14	**GRAPHS** Graph terminologies and Representation of Graphs Operation On graphs
W15	Adjacency Matrix ,Adjacency List
W16	**RECURSION** Basic of Recursion, Tower of Hanoi, Fibonacci Series.

Course Code	**CS212**
Course Title	**Operating Systems Concepts**
Cr Hrs	4(3+1)
Pre-requisite	CS211 (Data Structures & Algorithms)
Recommended Texts	1. Operating Systems: Internals and Design Principles (8th Edition), William Stallings, Pearson; 8 edition (February 2, 2014), ISBN: 978-0133805918. 2. Operating System Concepts, AviSilberschatz, Peter Baer Galvin, Greg Gagne, John Wiley & Sons, 2004, ISBN-9780471364146 | 978-0-471-36414-6. 3. Operating System Design: The Xinu Approach, Douglas Comer, 2015, ISBN-13: 978-1498712439
Course Description	This course gives you the overview on operating system in general and how different activities and problems are being handled by it. Different algorithms and techniques are introduced to students for handling different operations within OS
Course Objectives	• Build an understanding of the fundamental concepts of operating system. • Letting students understand the main techniques and algorithms being implemented for handling different issues in operating system

Week Wise Distribution of the Contents

Lecture No.	Topic
W1	Introduction: Operating System History & Evolution, Types of operating system (Batch, Real Time, Time Sharing, Single use, Multi-user)
W2	Process Concepts: Definition of Process, Process States, State Transitions, 2-State,3-State & 5-State Models & PCB
W3	Scheduling: Scheduling Definition, Scheduling levels (High Level, Mid-Level, Low Level)&Types of Scheduling (Preemptive, Non Preemptive)
W4	Scheduling Policies (FIFO, Round Robin, SJF, SRT, HRN)
W5	Parallel Processing: Definition, A Control Structure for Indicating Parallelism: Para begin/Para end, Mutual Exclusion & Critical Section
W6	Software Solutions to Mutual Exclusion (Dekker's Algorithm, Peterson's Algorithm) Hardware Solution to Mutual Exclusion (Test and Set Instruction)
W7	Inter process Communication Message Passing (Mailboxes & Ports, Pipes, remote Procedure Calls)
W8	Deadlock: Introduction, Example of Deadlock, Resource Concepts &Necessary Conditions for Deadlock
W9	Deadlock Prevention, Deadlock Avoidance Deadlock Detection, Deadlock Recovery
W10	Real Storage: Storage Organization, Storage Management, Storage Hierarchy,
W11	Storage Management Strategies (Fetch Strategies, Placement Strategies, Replacement Strategies)
W12	Contiguous VS Noncontiguous Storage Allocation, Fixed Partition Multiprogramming (Absolute Translation & Loading, Relocatable Translation & Loading, Protection, Fragmentation)
W13	Variable Partition Multiprogramming (Coalescing Holes, Storage Compaction, Storage Placement Strategies), Swapping
W14	Virtual Storage: Basic Concepts, Block Mapping, Paging Paging Address Translation with Direct Mapping, Paging Address Translation with Associative Mapping, Paging Address Translation with Combined Associative/ Direct Mapping Segmentation: Segmented Address Translation by Direct Mapping, Dynamic Address Translation in Paging/ Segmentation Systems
W15	Virtual Storage Management: Virtual Storage Management Strategies (Fetch Strategies, Placement Strategies, Replacement Strategies), Page Replacement Strategies (Principle of Optimality, Random Page Replacement, First-in-first-out, Least-Recently-used, Least-Frequently-Used, Not-Used-Recently, Demand Paging

W16	File System: File System Basics, File System Functions, Data Hierarchy, File Organization (Sequential, Direct, Indexed Sequential, Partitioned).

Course Code	**CS222**
Course Title	**Database Management Systems**
Cr Hrs	4 (3+1)
Pre-requisite	CS211 (Data Structures & Algorithms)
Recommended Texts	1. Fundamentals of Database Systems, Ramez Elmasri, 2015, 7th Edition, Pearson publishers, ISBN-10: 0133970779, ISBN-13: 978-0133970777 2. Database Systems: Design, Implementation, & Management, Carlos Coronel; Steven Morris, 2014, 11th Edition, Course Technology, ISBN-10: 1285196147, ISBN-13: 978-1285196145. 3. Database Systems: A Practical Approach to Design, Implementation, and Management, Thomas Connolly, Carolyn Begg, 2014, 6th Edition, ISBN-10: 0132943263, ISBN-13: 978-0132943260
Course Description	Investigates how database management system techniques are used to design, develop, implement and maintain modern database applications in organizations.
Course Objectives	The main objective of this course is to • Introduce students to fundamentals of database technology by studying databases from three viewpoints: those of the database user, the database designer, and the database administrator. • It teaches the use of a database management system (DBMS) by treating it as a black box, focusing only on its functionality and its interfaces.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction to Database Systems The relational model: Relational algebra and relational calculus, The SQL language
W2	Database application development Conceptual database modeling using the entity-relationship model
W3	Schema quality through the study of functional dependencies and normalization
W4	Database transactions, Physical database design and tuning
W5	Concurrency and Serializability Database security
W6	Analytics and data warehousing
W7	Map Reduce
W8	NoSQL systems
W9	Database Security / Distributed Database
W10	Transaction & Concurrency
W11	Normalization. 1NF, 2NF, 3NF, BCNF

W12	SQL DDL Statements, SQL DML Statements, SQL DCL Statements
W13	Introducing and Working on SQL Server, SQL Form Designing
W14	SQL Server Report Generation
W15	Connecting SQL Work Bench with NetBeans, Working with SQL workbench queries
W16	SQLite for Mobile Systems

Course Code	**CS251**
Course Title	**Software Engineering**
Cr Hrs	3 (3+0)
Pre-requisite	CS101 (Introduction to Computing)
Recommended Texts	1. Software Engineering: A Practitioner's Approach, Roger s. Pressman, Bruce Maxim, 8th Edition, 2014, McGraw-Hill Education, ISBN-10: 0078022126, ISBN-13: 978-0078022128. 2. Software Engineering, Ian Sommerville, 10th Edition, 2015, Addison Wesley, ISBN-10: 0321210263, ISBN-13: 978-0321210265. 3. Beginning Software Engineering, Rod Stephens, 1st Edition, 2015, ISBN-10: 8126555378, ISBN-13: 978-8126555376. 4. The Complete Software Developer's Career Guide: How to Learn Programming Languages Quickly, Ace Your Programming Interview, and Land Your Software Developer Dream Job, John Sonmez, 2017, Simple Programmer, LLC. 5. Essentials of Software Engineering, Frank Tsui and Orlando Karam, 2016, 4th Edition, Jones & Bartlett Learning, ISBN-10: 1284106004, ISBN-13: 978-1284106008.
Course Description	Software development is the practice of organizing the design and construction of software, the beating heart of much technology fundamental to our personal and professional life. This free introductory course, An introduction to software development, discusses the engineering nature of software development, its challenges and some fundamental practices which have developed to meet them. Software development is a fast-moving discipline and as a software development professional you must be able to track its leading edge. The course also teaches you some fundamental skills to help you interact with the growing published academic and professional literature on the subject.
Course Objectives	appreciate the engineering nature of software development To describe key activities in software development and the role of modelling explain key concepts in software development such as risk and quality explain the basics of an object-oriented approach to software development To describe a simple workflow for interacting with the published literature on software development.

Week Wise Distribution of the Contents

Lecture Number	Topics

W1	Introduction to software development, the discipline concerned with the methods, techniques and processes of building software artefacts.
W2	Software development as engineering
W3	Software Development Process: Analysis, Design
W4	Implementation, Testing, Deployment.
W5	Software development difficulties: A philosophical perspective, A historical perspective
W6	Risk: Risk management, Risk Analysis.
W7	Quality: Functional fitness, Usability
W8	Flexibility, Testability, Reusability,
W9	Modeling and the UML: Domain modeling.
W10	Specification modeling
W11	Design modeling
W12	Object orientation: Modeling with objects
W13	Programming with objects
W14	Finding and reading academic articles.
W15	A workflow for reading the academic literature: Preparation, Discovery, Assimilation, Recording, Relating.

Course Code	**CS371**
Course Title	**Data Communication and Computer Networks**
Cr Hrs	4 (3+1)
Pre-requisite	CS101(Intro. to Computing)
Recommended Texts	1. Data Communications and Computer Networks: A Business User's Approach, Curt White, Edition 8, Cengage Learning, 2015, ISBN: 1305465245, 9781305465244 2. Data Communication and Networking, Behrouz A. Ferouzan, 5th Edition, McGRAW-HILL, 2012, ISBN-10: 0073376221 ISBN-13: 978-0073376226. 3. Data and Computer Communication, William Stallings, 10th Edition, Prentice Hall, 2013, ISBN-10: 0133506487, ISBN-13: 978-0133506488.
Course Description	This course gives you the overview about, What Data Communication actually is? What tools and techniques you will use to send your data from one place to another. Similarly, all about the techniques, devices and concepts of data sending and receiving activities will be covered in this course
Course Objectives	• Build an understanding of the fundamental concepts of computer networking. • Familiarize the student with the basic taxonomy and terminology of the computer networking area &Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking. • Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction to Data comm. Peer to Peer vs. Client-Server Based Networking, Networking Topologies (Bus, Ring, Star and Mesh Topologies),
W2	Sizes of Networks (LANs, MANs and WANs)
W3	Network Models, Signals Digital and Analog Data Transmission, Transmitter, Receiver, Medium and Transceiver
W4	Broadcast vs. Point to Point Communication, Directional and Omni Directional Communication, Modes of Communication (Simplex, Half Duplex and Full Duplex) Types of Transmission (Asynchronous and Synchronous Transmission), Transmission Impairments (Attenuation, Distortion, Noise)
W5	Media: Bounded Media, Coaxial Cable, Twisted Pair (Shielded and Unshielded), Optical Fiber, Unbounded Media, Radio Waves, Microwave, Infrared, Satellites
W6	**OSI Reference Model:** Seven Layers of OSI model
W7	**TCP/IP Layers:** Five Layers of TCP/IP **Multiplexing:** Frequency Division Multiplexing, Time Division Multiplexing, Synchronous TDM & Statistical TDM
W8	**Switching:** Circuit switching, Space Division Circuit Switching, Time Division Circuit Switching, Packet Switching, Datagram Packet Switching & Virtual Circuit Packet Switching
W9	**Data Encoding:** Digital Encoding, NRZL, NRZI, Manchester and Differential Manchester, Analog Encoding, Amplitude Frequency and Phase Shift Keying **Direct Link Networks:** Error Detection and Correction, Two Dimensional Parity, Cyclic Redundancy Check& Hamming Codes
W10	Flow Control, Stop & Wait ARQ, Sliding Window Algorithm Error Control, Stop & Wait ARQ, Go Back N ARQ & Selective Reject
W11	Ethernet (IEEE 802.3), Physical Properties, Frame Format & Addressing Token Ring (IEEE 802.5): Physical Properties, Frame Format, Token Algorithm, Monitor Election and Responsibilities of Monitor
W12	FDDI: Physical Properties, Frame Format, Token Algorithm Wireless LAN (802.11): Physical Properties, Distribution System, Hidden Nodes Problem & Exposed Nodes Problem.
W13	**Internetworking:** Internet Protocol, IPV4, Frame Format, Segmentation and Reassembly, IPv4, Sub netting and CIDR&IPv6
W14	*End to End Issues*: Transmission Control Protocol, Connection Establishment Connection Termination, TCP State Transition Diagram &User Datagram Protocol
W15	Connecting LANs
W16	Network Programming

Course Code	**CS372**
Course Title	**Information Security**
Cr Hrs	3 (3+0)
Pre-requisite	CS371**(**Data Communication and Computer Networks)

Recommended Texts	1. Principles of Information Security, 2015 4th Edition by Michael E. Whitman and Herbert J. Mattord. 2. Computer Security: Art and Science, Matthew Bishop, 2018, 2nd Edition, ISBN-10: 0321712331, ISBN-13: 978-0321712332 3. Cryptography and Network Security by William Stalling 7th Edition, 2015
Course Description	This course is intended to help students gain fundamental and comprehensive understanding of information security. We will focus on an overview of major information security issues, technologies, and approaches. Students who successfully complete this course will have a concept and knowledge of security properties, concerns, policies, models, cryptography, PKI, firewalls, security evaluation, and real-life security cases. Students will also have hands-on experience in selected information security technologies through lab sessions.
Course Objectives	• To provide an understanding of principal concepts, major issues, technologies, and basic approaches in information security. • To provide concept-level hands-on experience in specific topic area. • To provide the ability to examine and analyze real-life security cases.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction and Overview of Information Security Understanding the Threats: Malicious software, Viruses, trojans, rootkits, worms, botnets.
W2	Memory exploits (buffer overflow, heap overflow, integer overflow, format string)
W3	Formalisms: Access control theory, access control matrix, Information flow), Policies: Security policies, Confidentiality policies (BLP model),
W4	Integrity policies (Biba, and Clark-Wilson model), Hybrid policies (Chinese Wall model, role-based access control)
W5	Cryptography Block and stream ciphers, Cryptographic hash functions., Message Authentication Codes (MAC), Public and private key systems, Message digests. Approximate strength of ciphers, Authentication, Password system.
W6	Systems Secure design principles (Least-privilege, fail safe defaults, complete mediation, separation of privilege.
W7	TCB and security kernel construction, System defense against memory exploits
W8	UNIX security and Security-Enhanced Linux (SELinux),Windows security.
W9	Network Security: TCP/IP security issues, DNS security issues and defenses, TLS/SSL. Network Intrusion detection and prevention systems ,Firewalls.
W10	Software Security: Vulnerability auditing, penetration testing, Sandboxing, Control flow integrity.
W11	Web Security: User authentication, authentication-via-secret and session management.

W12	Cross Site Scripting, Cross Site Request Forgery, SQL Injection.
W13	database security
W14	host- based and network-based security issues, operational security issues, physical security issues; personnel security electronic voting
W15	classification and trust modeling; risk assessment, IDS
W16	Legal and Ethical Issues: Cybercrime and computer crime, Intellectual property, copyright, patent, trade secret, Hacking and intrusion, Privacy, identity theft.

Course Code	**CS131**
Course Title	**DIGITAL LOGIC AND DESIGN**
Cr Hrs	3 (3+0)
Pre-requisite	Nil
Recommended Texts	1. Digital Logic and Computer Design, M. Morris Mano, prentice hall publisher, 2015, ISBN-13: 978-0132145107 2. Digital Computer Electronics, Albert P. Malvino, Jerald A Brown, Career Education publisher, 2015, ISBN-13: 978-0028005942 3. Digital Computer Electronics, Malvino Brown, 4th Ed, Career Education Inc, 2010, ISBN-10: 0028005945, ISBN-13: 978-0028005942. 4. Digital Logic Design: A Rigorous Approach, Guy Even, Ist Ed, Cambridge University Press, 2012, ISBN: 978-1107027534.
Course Description	This course provides a modern introduction to logic design and the basic building blocks used in digital systems, in particular digital computers. It demonstrates knowledge of practical aspects of digital components including setup and hold time in flip-flops and fan-in, fan-out, and noise margin in logic gates. Create minimal realizations of single and multiple output Boolean functions. Design and analyze combinational circuits using medium scale integrated (MSI) components, including arithmetic logic units. Demonstrate knowledge of clocking issues within synchronous systems; Demonstrate knowledge of hazards and race conditions generated within asynchronous FSMs
Course Objectives	A student who successfully fulfills the course requirements will have demonstrated: An ability to define different number systems, binary addition and subtraction, 2's complement representation and operations with this representation. • An ability to understand the different switching algebra theorems and apply them for logic functions. • An ability to define the Karnaugh map for a few variables and perform an algorithmic reduction of logic functions. • An ability to define the following combinational circuits: buses, encoders/decoders, (de)multiplexers, exclusive-ORs, comparators, arithmetic-logic units; and to be able to build simple applications. • An ability to understand the bits table element and the different latches and flip-flops.

Week Wise Distribution of the Contents

Lecture Number	Topics
W1	Digital concepts: Number Systems
W2	binary arithmetic binary codes error detection and correction
W3	parity check methods Combinational logic: Pulsed operation and troubleshooting of gate networks
W4	Applications of combinational logic in different logic functions and their troubleshooting
W5	Flip-flops and related devices
W6	Operating characteristics, basic applications and troubleshooting
W7	Counters: Different types of counters design of sequential circuits, applications and troubleshooting of counter circuits, symbols and dependency notations
W8	Shift Registers: Different types of shift register, their application and trouble shooting symbols and dependency notations
W9	Memories and Programmable Logic Devices: Memory concepts, types of memory
W10	Programmable Logic Devices, testing and troubleshooting
W11	Integrated Logic Circuit Technologies: Basic operational circuit parameters
W12	data sheet interpretation TTL circuits, practical considerations in the uses of TTL
W13	CMOS circuits, comparison of CMOS and TTL characteristics, interfacing logic families
W14	ECL circuits, I2L circuits, PMOS and NMOS
W15-W16	Laboratory: Construction of logic gates using discrete components, study of characteristics of digital integrated circuits, design and construction of different types of combinational circuits like half adders, full adders, parallel binary adders, look ahead carry adders, comparators, decoders, encoders, multiplexers, de-multiplexers, Parity generators/checkers, different type of sequential circuits and their troubleshooting. Design, construction and study of different types of flip-flops and their troubleshooting. Comparatives study of digital ICs of different technologies and their mutual interfacing

Course Code	**CS311**
Course Title	**Theory of Automata**
Cr Hrs	3 (3+0)
Pre-requisite	None
Recommended Texts	1. An Introduction to Formal Languages and Automata 6th Edition, Peter Linz Jones & Bartlett Learning publisher (January 26, 2016), ISBN: 978-1284077247 2. Compiler Construction Using Java, Javacc And Yacc, Anthony J Dos Reis WIEEECS Publisher, 1 edition (2015), ISBN: 978-8126556182 3. Theory of Automata, Alban Andahi, 2018, Createspace Independent Publishing Platform, ISBN-10: 1985742446 ISBN-13: 978-5742444 *4.* Compilers: Principles, Techniques, and Tools, Alfred V. AhoMonica S. LamRaviSethiJeffrey D. Ullman, Addison Wesley, 2006 ISBN-978-0321486813
Course Description	This is an introductory course on Theory of Automata and Compiler Construction. Students are introduced to the concept of Formal Language and Automata. In Automata they learn about finite automata (deterministic; non-deterministic), transition graphs and pushdown automata (deterministic; non-deterministic). The course also introduces the students to the operation of a modern compiler that translates code in a programming language into machine code. However, due to the introductory nature of the course, we will spend majority of the time on syntax analysis and code generation.
Course Objectives	• To introduce concepts in automata theory and identify different formal language classes and their relationships • Design grammars and recognizers for different formal languages • To prove and disprove theorems in automata theory using its properties. • To know the working of compiler and to enable the students to learn different programming techniques in constructing a compiler.

Week Wise Distribution of the Contents

Lecture No	Topic
W1	Introduction Terminologies of Languages Operations on languages
W2	Descriptive definition Examples using Descriptive definition Recursive Definition
W3	Regular Expressions(RE)
W4	Finite Automaton(FA)

W5	Non- Deterministic Finite Automata- NFA Conversion of NFA to DFA Union of Two Fas
W6	Concatenation of Two Fas Transition Table Transition Graph-TG
W7	Generalize Transition Graph- GTG Context Free Grammar-CFG
W8	Designing Context Free Grammar ,Chomsky Normal Form
W9	Context Sensitive Languages ,Context Sensitive Grammars
W10	Tree Ambiguity Pushdown Automata
W11	Introduction to Passes of a Compiler Text Processing Parsing
W12	Specification of Tokens, Recognition of Tokens Top-down Parsing Predictive Parsing
W13	Lexical Analysis Specification of Token Recognition of Token
W14	Lexical Generator Syntax Analysis Intermediate Generation of Codes
W15	Translation of Expressions Type Checking
W16	Code Generation

Course Code	**CS233**
Course Title	**Computer Organization and Assembly Language**
Cr Hrs	4 (3+1)
Pre-requisite	CS131 (Digital Logic And Design)
Recommended Texts	1. Computer Organization and Design, David A. Patterson, John L. Hennessy, Morgan Kaufmann publisher, 2016, ISBN-13: 978-0124077263 2. Principles of Computer Organization and Assembly Language, Patrick Juola, Pearson publisher, 2015, ISBN-13: 978-0131486836 3. Assembly Language for Intel-based Computers, Kip R. Irvine, 5th Ed, Prentice Hall Publishing, 2006, ISBN: 978-0130910134 4. Computer Organization Assembly Language, Micheal Thorne, 2nd Ed, Addison Wesley, 1991, ISBN: 978-0805368796. 5. Professional Assembly Language, Richard Blum, 1st Edition, Wrox Publisher, 2005, ISBN: 978-0764579011
Course Description	Computer Organization and Assembly Language is aimed to enable students to study and explore in detail the machine representation of instructions and data using a modern digital computer. This course enables students to study microprocessor addressing and the mechanism behind data movement between memory and microprocessor. Student will also experiment to program interrupts and to perform interrupt driven I/O. Basic machine organization are studied. The course will focus on the most popular Intel 80x86 microprocessor. It is suggested for the benefits of students to use emu8086 emulator for the entire course in order to avoid any compatibility issues that may arise due to the recent advancements in contemporary processors and Operating Systems.
Course Objectives	The main objectives of this course is to • Introduce the organization of computer systems and usage of assembly language for optimization and control. • Emphasis should be given to expose the low-level logic employed for problem solving while using assembly language as a tool. At the end of the course the students should be capable of writing moderately complex assembly language subroutines and interfacing them to any high level language.

Week Wise Distribution of the Contents

Lecture Number	Topics
W1	Introduction to the course. The need to study assembly language Resources (books, assemblers, emulators etc) • General concepts about microcomputer, microprocessors • objectives and perspective of assembly language • Instruction execution cycle • Reading/writing to memory • programmable CPU registers and their categories 16-bit, 32-bit and 64-bit registers
W2	Bus and bus types Microprocessors bus Structure Address, data and control lines of a bus

W3	Memory Organization and Structure Linear and segmented memory models
W4	Addressing modes
W5	Introduction to the Assembler and Debugger
W6	Register programming: Data movement, arithmetic Addressing: Indirect addressing, arrays, Indexed operands, Pointers Flags register: flags description
W7	Programming various flags
W8	Program Control Instructions: jump and loop instructions
W9	Subroutines: run time stack (32-bit)
W10	Stack operations (push, pop)
W11	Defining and Using procedure: Call, RET and Proc directives Nested procedure call, passing register arguments to procedures
W12	64-bit programming
W13	64-bit addition and subtraction
W14	Linking to an external library Peripherals Control Interrupts
W15	Interfacing with High Level Languages
W16	Course Review, guidelines for final term exam

Course Code	**CS411**
Course Title	**Design and Analysis of Algorithms**
Cr Hrs	4 (3+1)
Pre-requisite	CS211 (Data Structures & Algorithms)
Recommended Texts	1. An Introduction to the Analysis of Algorithms (2nd Edition) , Robert Sedgewick , Philippe Flajolet, Addison-Wesley Professional Publisher (January 28, 2013), ISBN: 978-0321905758 2. Practical Analysis of Algorithms (Undergraduate Topics in Computer Science), 4th Edition , Dana Vrajitoru , William Knight , Publisher: Springer (September 15, 2014), ISBN: 978-3319098876 3. Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, 2011, Pearson Publishers, ISBN-10: 0132316811 ISBN-13: 978-0132316811. 4. Analysis Of Algorithms, Jeffrey McConnell, 2007, ISBN-13: 978-0763707828
Course Description	This course gives student the understanding of the designing aspects of an algorithm. It will also make them grasp the concept of analysis of an algorithm. Student will be provided with real life problems and then will get a walkthrough from designing and analysis phase of its algorithm to better understand. This will help student understand the core concepts and notions.
Course Objectives	• Provide students with an understanding of the principles and techniques used in the design and analysis of algorithms. • Provide the understanding of the notion of a mathematical proof and some knowledge of elementary discrete mathematics.

	- Student will analyze a variety of data structures and algorithms chosen for their importance and their illustration of fundamental concepts. - Making student understand the analysis of the worst-case running time of an algorithm as a function of input size.

Week Wise Distribution of the Contents

Lecture Number	Topics
W1	Introduction, Analyzing the worst-case running time of an algorithm as a function of input size
W2	Asymptotic notations solve recurrence relations
W3	Divide-and conquer technique
W4-W6	Understand the concepts of Dynamic programming
W7-W9	Understand the concepts of Greedy Algorithm
W10-W13	Understand the concepts of Graph traversing
W14	Understand the concepts of MST and their algorithms
W15	Understand the algorithms for computing shortest path
W16	Understand the algorithms for computing shortest path, Basics of Complexity theory

Course Code	**CS241**
Course Title	**Web Engineering**
Cr Hrs	4 (3+1)
Pre-requisite	CS102 (Programming Fundamentals)
Recommended Texts	1. Web Design with HTML, CSS, JavaScript and jQuery Set 1st Edition, Jon Duckett, Wiley, 2014, ISBN-13: 978-1118907443 2. Web Design All-in-One For Dummies 2nd Edition, Sue Jenkings, For Dummies, 2013, ISBN-13: 978-1118404102 3. Head First HTML5 Programming, Eric Freeman and Elisabeth Robson, 2011, ISBN: 978-1-449-39054-9 4. How to Do Everything with JavaScript, Scott Duffy, 2013, ISBN-0-07-222887-3. 5. A Practical Guide to Designing for the Web, Mark Boulton Design, Limited, 2009, ISBN-13: 978-0956174017
Course Description	This course introduces students to basic web design using HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). The course does not require any prior knowledge of HTML or web design. Throughout the course students are introduced to planning and designing effective web pages; implementing web pages by writing HTML and CSS code; enhancing web pages with the use of page layout techniques, text formatting, graphics, images, and multimedia; and producing a functional, multi-page website.
Course Objectives	- Recognize and understand HTML web page elements - Know how to write HTML code - Understand and apply effective web design principles - Enhance web pages using text formatting, color, graphics, images, and multimedia

	• Incorporate forms into web pages • Understand and apply CSS to format web page elements • Adding scripting language to add action to HTML. • To make the design and contents of a website optimized for search engines called Search Engine Optimization (SEO)

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction to the Course, Hyper Text Markup Language (HTML), Setting Environment for HTML (Software) Getting started with tags, HTML Elements ,Paired and Singular Tags, HTML, Head, Title, Body tags, Writing simplest web page, How to save webpage, How to view webpage
W2	HTML Paragraphs, Headings, HTML Attributes, HTML Comments HTML Formatting, Bold, Emphasize, Italic, Underline, Marked, Small, Deleted, Inserted, Subscript, SuperScript, Code, Preformatted, Abbreviation, Quotations, Cite, Address etc
W3	HTML Colors, Standard Colors, RGB Colors, HexaDecimal Colors, Linking Documents Images, Linked Images, HTML Lists, Ordered Lists, Un-Ordered Lists HTML Table
W4	HTML Forms, Form Methods, Form Elements
W5	HTML5 Form Elements Meta Tag, Inline Frames (IFrame)
W6	Div and Span, HTML Media, Video, Audio, Plug-Ins
W7	HTML Graphics, HTML Convas HTML API's, Geolocation
W8	HTML Drag-Drop HTML Local Storage, App Cache
W9	App Cache, HTML Web Workers
W10	Cascaded StyelSheet (CSS), The Style Attribute, CSS Syntax Inlined CSS, Internal CSS, External CSS
W11	CSS Selectors, Tags, Class, Id, Pseudo Selectors, Attribute Selectors, CSS3 Selectors CSS Properties, Font CSS Properties, Text CSS Properties, Setting Size (various units of text size) Background CSS Properties, Border CSS Properties, Links CSS Properties Lists CSS Properties
W12	Box Model in detail, Setting Layout, Using Display, Position, Float, Width, Height CSS properties for setting layout, using Div for Layout Setting Introduction to Responsive Design Responsive Design explained with an example
W13	Introduction to Scripting Languages, Introduction to Javascript, How to add Javascript to HTML InlineJavascript, Internal Javascript, External Javascript Javascript Syntax, Data Types, Variables, Complex Data Types (Arrays, Objects), Operators

W14	Defining Methods in Javascript, Selection Statements in Javascript, Looping in Javascript Event-driven HTML and Javascript RegEx, Javascript Form Validation
W15	JavascriptDebuggingm, Understanding Browser support JS HTML Document Object Model (DOM) JQuery Basics, JQuery in Action, JSON Basics Communicating with Web Server using Javascript
W16	Search Engine Optimization (SEO) Getting website on Internet, understanding how HTTP Works and Request are processed, Understanding the overall cycle of how web works What is Domain Name, Selecting Domain Names, Purchasing Domain Names, Purchasing Hosting Space on Shared Web-Server Uploading files on rented space using FTP, How to bind your domain name with hosting server, Visiting Website Live, How to make money from your website by advertisement

Course Code	**CS363**
Course Title	**Artificial Intelligence**
Cr Hrs	3 (3+0)
Pre-requisite	None
Recommended Texts	1. Artificial Intelligence: What Everyone Needs to Know 1st Edition: Jerry Kaplan ,Oxford University Press, 2016, ISBN: 978-1493682225 2. The Cambridge Handbook of Artificial Intelligence, Keith Frankish, William M. Ramsey, Cambridge University Press, 2014, ISBN: 978-0521691918 3. Artificial Intelligence: A Modern Approach, James Barrat, 2015, 2nd Edition, ISBN-10: 1250058783, ISBN-13: 978-1250058782 4. Introducing Artificial Intelligence (Introducing...(Totem)) by Henry Brighton (2004), ISBN-13: 978-1840468410 5. Artificial Intelligence: A Modern Approach (3rd Ed.), by S. Russell and P. Norvig, Prentice Hall, 2010, ISBN-13: 978-0136042594
Course Description	This course will introduce the basic principles in artificial intelligence research. It will cover simple representation schemes, problem solving paradigms, constraint propagation, and search strategies. Areas of application such as knowledge representation, natural language processing, expert systems, vision and robotics will be explored. The LISP programming language will also be introduced.
Course Objectives	• To have an appreciation for and understanding of both the achievements of AIand the theory underlying those achievements. • To have an appreciation for the engineering issues underlying the design of AIsystems. • To have a basic proficiency in a traditional AI language including an ability to write simple to intermediate programs and an ability to understand code written inthat language.

	- To have an understanding of the basic issues of knowledge representation and blind and heuristic search, as well as an understanding of other topics such as minimax, resolution, etc. that play an important role in AI programs.

Week Wise Distribution of the Contents

Lecture Number	Topics
W1	INTRODUCTION, Intelligent Agents, Problem Solving Agents Introduction: What is Artificial Intelligence, Definition and Applications, Thinking Humanly, Thinking Rationally, Acting Humanly, Acting Rationally, Rational Agents, A Brief History Intelligent Agents: Agents, Agents and Environment, Rational Agents, PEAS (Performance Measure, Environment, Actuators, Sensors), Environment Types
W2	Agent Types and Basic Architectures: Simple Reflex Agents, Model-Based Reflex Agents, Goal-Based Agents, Utility Based Agents, Learning Agents, Problem Solving Agents, Goal Formulation, Problem Formulation Search Tree Search Algorithms: General Tree Search, Search Strategies, Uninformed Search Strategies Uninformed Search Strategies, Breadth – First Search, Uniform – Cost Search, Depth – First Search
W3	Informed (Heuristic) Search Strategies, Graph Search, A* Search Probability in Artificial Intelligence, Bayes Rule, Examples and Applications, Probabilistic Inferences
W4	Machine Learning: Types of Machine Learning, Supervised , learning, Unsupervised Learning, Reinforcement Learning, Statistical Learning Methods
W5	Supervised Learning, Bayesian Learning, Application, Bayesian Learning for Classification, Implementation.
W6	Unsupervised Learning – Clustering, , k-Nearest Neighbors
	Artificial Neural Networks, Fuzzy Logic
W7	Knowledge and Reasoning: Logical Agents, Knowledge Based Agents, Logic, Propositional Logic, First Order Logic
W8	Programming with Prolog Case Based Reasoning Semantic Networks
W9	Planning: Planning, Planning Under Uncertainty, Analysis of Planning Approaches
W10	Advanced Planning Games: Markov Decision Processes – I
W11	Markov Decision Processes – II, Evaluation Functions
W12	Game Theory – I Game Theory – II Hidden Markov Models and Filters: Markov Networks
W13	Gibbs Sampling, Particle Filtering Hidden Markov Models – I Hidden Markov Models – II

Lecture Number	Topics
W14	Applications: Computer Vision – I. Computer Vision – II Robotics – I,
W15	Robotics – II Grammar of English Natural Language Processing – I,
W16	Natural Language Processing – II

Course Code	**CS443**
Course Title	**Computer Graphics**
Cr Hrs	4(3+1)
Pre-requisite	None
Recommended Texts	1. Computer Graphics: Principles and Practice (3rd Edition) 3rd Edition, John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, Kurt Akeley, Addison-Wesley Professional, 2013, ISBN-13: 978-0321399526 2. Fundamentals of Computer Graphics, Fourth Edition 4th Edition, Steve Marschner, Peter Shirley, A K Peters/CRC Press, 2015, ISBN-13: 978-1482229394 3. Computer Graphics Through OpenGL: From Theory to Experiments, Second Edition, SumantaGuha, A K Peters/CRC Press, 2014, ISBN-13: 978-1482258394 4. Computer Graphics using OpenGL, F.S. Hill Jr. & Stephen M. Kelley Jr, Pearson Books, 2016, ISBN-0023548568. 5. Computer Graphics with OpenGL, Donald Hearn and M. Pauline Baker, Prentice Hall, 2013, ISBN-0130153907
Course Description	Important topics in computer graphics include sprite graphics, vector graphics, 3D modeling, shaders, GPUdesign, and computer vision, among others. The overall methodology depends heavily on the underlying sciences of geometry, optics, and physics. Computer graphics is responsible for displaying art and image data effectively and beautifully to the user, and processing image data received from the physical world. The interaction and understanding of computers and interpretation of data has been made easier because of computer graphics. Computer graphic development has a significant impact on many types of media and has evolutionized animation, movies, advertising, video games, and graphic design generally.
Course Objectives	• Students will use a standard computer graphics API (OpenGL) to reinforce concepts and study fundamental computer graphics algorithms. • Computer Graphics is the study of basic concepts and principles of graphics. • Topics include an introduction to the basic concepts, 2-D and 3-D modeling and transformations, viewing transformations, projections, rendering techniques, graphical software packages and graphics systems.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction to Computer Graphics: Course Introduction, Course Objectives, Course Contents, Quality expectations , An Initial Definition of Computer Graphics, Standard Definitions and Terms, Computer Graphics Research and Development Scope, Levels of Computer Graphics, Pattern, Pattern recognition , Pattern Classification, Brightness Adaptation And Discrimination, Contrast Sensitivity, Mach Bands, Image Formation Model, Historical Milestones
W2	Vector Math: Vector Definition, Vector Components , Vector Algebra, Vector Addition, Vector Addition with Numbers, component-wise operation, Vector Subtraction, Vector Multiplication, Unit Vectors and Normalization, Image Components in Matlab, Image Vector Operation in Matlab
W3	Open GL in Visual C++ .NET: Low-level API, cross-language ,cross-platform, 2D, 3D computer graphics, Why Learn About OpenGL, Graphics Platform, GLUT - The OpenGL Utility Toolkit, Configure and open window (GLUT) , Initialize OpenGL (Optional), Register input callback functions (GLUT) , Render, Resize ,Input: keyboard, mouse ,Enter event processing loop (GLUT)
W4	OpenGL Programming : The Header File Of OpenGL #Include <GL/Glut.H>,A Simple Open GL Program To Create A Window, Clearing The Window, Windows Back Color, Closing The Window With Esc Key, Plotting Pixels , Plotting A Triangle
W5	OpenGL Programming: Filling The Triangle, Printing Cubes , Drawing A Complex Geometrical Shape , Plotting A Filled Square, Display A Filled Circle , Rotating Cubes, Clock Wise And Anti Clock Wise Rotation, 3 D Object Rotation
W6	Geometric Primitives in Open GL : Geometry Elements , A Triangle as a Geometry Object, A Simple Example of Geometry Elements , A Complex Example of Geometry Elements, Double Buffering In Animation , Animation Example in OpenGL.
W7	OpenGL for Interactivity: Image Processing, Image Synthesis, Photo real Rendering, Non-Photo real Rendering, Animation, Physics Simulation, Character Animation, Modeling, Raster and Vector Graphics, Graphics Interlacing, Frame buffer in Animation, 3D Models, Handling Keyboard Inputs with GLUT, Handling Mouse Inputs with GLUT
W8	Traditional Graphics Pipeline: Transformation, Lighting, Clipping, Scan conversion, Pixel processing.
W9	Color Models: Types of Color model, RGB Color Model, CMYK Color Model, Additive vs. Subtractive Color Models , HSI or HSV Color Model.
W10	Transformations: Translation, Scaling, Rotation, OpenGL Examples
W11	Lighting: Blinn-Phong Lighting Model, Ambient Lighting Model, Diffuse Lighting Model, Specular Lighting Model, Light source, Material Properties, Light Example in OpenGL.
W12	Shading: Shading model, some real word examples, Flat shading, Smooth Shading, OpenGL Examples of Flat and Smooth Shading. Triangulation: Problem of Triangulation, Triabulation, Calculation, Trifocal Tensor (Tritensor), Photogrammetry.
W13	Texture: Texture, Texture Applications, Textures Examples, Texture Synthesis, Texture Representing Methods , Statistical Methods, The Choice of Statistic, The Choce of Scale , Representing Texture Using the Statistics of Filter Outputs, Histogram based texture description, Grey Level Co-occurrence Matrix (GLCM).

Lecture Number	Topic
W14	Radiosity: Cornell Box, Lighting Effects, Phong Shading, Ray Tracing, Planar piecewise constancy assumption, Diffuse Interreflections – Radiosity, Conservation of Energy. Counting the Object In the 3D Image: Step 1: Read the Image, Step 2: Convert the Image to Grayscale, Step 3: Threshold the image, Step 4: Complement the image, Step 5: Find the Boundaries of the Objects, Step6: Results, Matlab Code.
W15	Image Segmentation: Segmentation, Segmentation Applications, Human Vision: Grouping and Gestalt, Gestalt Factors, Parallel curves, symmetric groups, continuous curves, Closure Curves, Visual illusions (examples), Simple Segmentation Techniques: Background Subtraction, Shot Boundary Detection, Segmentation by Clustering, displaying objects in the Segmented Image, Detecting a Cell Using Image Segmentation.
W16	Unity 3D: Introduction to Unity 3D, Scripting in Unity 3D via Visual C#.NET

Course Code	SE202
Course Title	Software Requirement Engineering
Cr Hrs	3 (3+0)
Pre-requisite	SE101 (Software Engineering)
Recommended Texts	1. Requirements Engineering for Software and Systems, Phillip A. Laplante, Second Edition (Applied Software Engineering Series) 2nd Edition, CRC Press, 2015, ISBN: 978-1466560819 2. Requirements Engineering: Foundation for Software Quality, Maya Daneva Oscar Pastor, Springer, 2016, ISBN: 978-3-319-30282-9 3. Requirements Engineering: Fundamentals, Principles, and Techniques 2010th Edition, Klaus Pohl, Springer; 2010 edition, ISBN-10: 3642125778, ISBN-13: 978-3642125775 4. Software Requirements (3rd Edition) (Developer Best Practices), Karl Wiegers & Joy Beatty, 2016, ISBN-10: 0735679665, ISBN-13: 978-0735679665 5. Requirements Engineering for Software and Systems, Third Edition (Applied Software Engineering Series) 3rd Edition, Phillip A. Laplante, 3rd Edition, Auerbach Publications, ISBN-10: 1138196118, ISBN-13: 978-1138196117
Course Description	This course enables students to understand basics of requirement engineering, its importance and various models used for gathering it.
Course Objectives	• Definition of requirements and its role in development • Fundamental concepts and activities of engineering, Information election techniques, Modeling scenarios • Fundamental of goal, Goal modeling heuristics, Object modeling for requirement engineering • Deriving operational requirements from goals, requirements specification, requirements verification and validation

Week Wise Distribution of the Contents

Lecture Number	Topic

W1	Introduction to Software Requirements Engineering, Introduction of Software Requirements Engineering, IEEE Definition of Requirements Engineering. Types of Requirements
W2	Software Quality Requirements, The Four Worlds Model in Requirement Engineering. Software Requirement Traceability: The V Model and Requirements
W3	Concern of RE in each layer of V Model, Requirements and Communication, Tractability and Change Management. Tractability Analysis, Impact Analysis, Derivation Analysis, Coverage Analysis, Requirements and Testing
W4	Software Requirements Engineering Process Models: The Processes, Software Process Models, Process Difficulties, Focus of the Process, Process Decomposition (ISO 12207), Process Architecture
W5	RE Process Activities, RE process - inputs and outputs, RE process variability and Models. Linear RE process Model, Linear Iterative RE Process Model, Iterative RE Process Model, Spiral model of the RE process, Diagrams Represent Process Disciplines
W6	RE process problems, Process improvement, Planning process improvement, Good practice for RE process improvement Examples of good practice guidelines Requirements Elicitation: Requirements Elicitation, Reasons of elicitation, How to do requirements elicitation? Requirements Elicitation Process, Components of requirements elicitation, Elicitation activities, The requirements elicitation process
W7	Elicitation stages, Focus of Elicitation, Outcomes of Requirement Elicitation Requirements Engineering Techniques: Interviews, Types of interviews, Surveys, Questionnaires Task Analysis, Domain Analysis
W8	Introspection, Repertory Grids, Card Sorting, Class Responsibility Collaboration, Laddering, Group Work Brainstorming, Joint Application Development (JAD Requirements Workshops, Ethnography, Observation, Protocol Analysis
W9	Prototyping, Ethical and Legal Considerations in RE: Ethical Considerations, Some Of The Key Ethical Considerations in RE. The Right To Be Informed, The consent form, Permission to Record, Create a Comfortable Experience, Appropriate Language, Anonymity, The Right To Withdraw
W10	Appropriate Incentives, Valid and Reliable Data, Acknowledge the True Capabilities, Data Retention and Documentation, Debrief, Confidential Disclosure Agreement (CDA), Preparing for the User Requirements Activity: Creating a Proposal, Why Create a Proposal?, Sections of the Proposal,

	Programming practice, Top down & bottom up ,Structure Programming, Information hiding
W11	Internal documentation, conducting a Wants and Needs Analysis. Moderating Requirements Engineering Activity, Key Guidelines for Moderator. Inviting Observers, Advantages of Observers, Guidelines for Observers, Recording and Note-taking, Dealing with Awkward Situations
W12	Using Company's Existing Facilities, renting a Marketing or Hotel Facility, Building a Permanent Facility. Devoted User Requirements Facility and Digital versus Analog Labs, Learning about the Product & User. Networking, Customer Support Comments, Log Files, Marketing Department, Early Adopter or Partner Feedback
W13	Creating a User Profile, Persona, Benefits of personas, Things To Be Aware of When Creating Personas, Scenarios, Benefits of a Scenario, Components of Scenario, Template of Scenarios. Requirements Validation: Validation Objectives, Analysis & Validation
W14	Requirements Review Process and its Activities, Problem Actions, Pre-review Checking. Review Team Membership, Review Checklist
W15	Validation Inputs and Outputs, Prototyping for validation, model validation requirements testing, Test case definition, Requirements Test Form
W16	Standards and Requirements Validation EIA-632 Standards

Course Code	SE341
Course Title	Software Quality Engineering
Cr Hrs	3(3+0)
Pre-requisite	SE101 (Software Engineering)
Recommended Texts	1. Software Quality Assurance, Ivan Mistri Richard M Soley John Grundy Bedir Tekinerdogan and Nour Ali, Elsevier Inc, 2016, ISBN: 978-0-12-802301 2. Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement", by Jeff Tian 3. Software Quality Assurance: Principles and Practices 2nd Revised edition Edition, Nina S. Godbole, 2016, Alpha Science International Ltd, ISBN-10: 184265702X, ISBN-13: 978-1842657027
Course Description	This course gives you the overview of the concepts, philosophies, tools, techniques, guidelines, and framework for software quality assurance, verification and validation of software, measurement of quality, quality factors, quality costs at different stages of software

	development lifecycle, Software testing and usability evaluation methods Software quality standards.
Course Objectives	• This course concentrates on the rigorous development of high quality software systems • It includes the processes, methods and techniques for developing quality software, for assessing software quality, and for maintaining the quality of software.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction to Software Quality Assurance: Course Introduction Course Objectives, Course Contents, Quality expectations, An Initial Definition of (Software) Quality, Standard Definitions of (Software) Quality, Quality Problems in Large Software
W2	SQA Expectation : Turning Failure to Success: SQA, SQE for Meeting Quality Expectations, The SQE Process, Scope of Major SQE Activities
W3	Software Quality Frameworks : Quality: Perspectives and Expectations, Quality Frameworks and ISO-9126, Correctness and Defects: Definitions, Properties, and Measurements , A Historical Perspective of Quality
W4	Software Quality Assurance Activities: Quality Assurance (QA), Classification Scheme for QA as Dealing with Defects, Defect Prevention, Defect Reduction (Defect Detection and Removal), Defect Containment
W5	Software Quality through Verification and Validation: QA in context, QA as Defect Handling, QA in Software Process Context, QA activities in V&V Context, V&V in Software Process, QA: Defect View vs. V&V View
W6	SQE embedded in Software Process: Quality Engineering, Key SQE Activities, SQE in Software Process
W7	Testing Overview: Mission of test groups, Generic Test Process, Test Concepts, What to Test: Black Box vs. White Box, Stopping Criteria: Coverage Based vs. Usage Based, Lifecycle Testing
W8	Testing Activities: Detailed Look at Major Testing Activities, Test Management: Responsibilities and Roles, Testing Automation, Test Documentation
W9	Boundary Testing : Checklist-Based Testing, Partitions and Partition Testing, Usage-Based Testing with Musa's Ops, OP Development, Procedure/Examples
W10	Coverage and Usage Testing Based on FSMs and Markov Chains: Finite-State Machines (FSMs), FSM-Based Testing, Markov Chains as Enhanced FSMs, Unified Markov Models (UMM) for Testing
W11	Software Inspection: Basic Concept and Generic Process, Fagan Inspection, Other Inspection and Related Activities, Other Issues.
W12	Defect Prevention and Process Improvement: Basic Concepts and Generic Approaches, Root Cause Analysis for Defect Prevention, Education and Training for Defect Prevention, Other Techniques for Defect Prevention, Focusing on Software Processes

W13	Fault Tolerance and Safety Assurance: Basic Concepts, Fault Tolerance via RB and NVP, Safety Assurance Techniques/Strategies, Summary and Perspectives
W14	Quality Through Automated Tools: UML generation, UML conversion into Coding Coding Standards: Coding Standards Rules, Best Practices
W15	Comparing QA Alternatives: General Areas/Questions for Comparison, Applicability, Effectiveness, and Cost, Summary and Recommendations
W16	Quality in GUI: Quality in user behavior Pattern, Guidelines for Quality interface designing.

Course Code	SE311
Course Title	Software Design and Architecture
Cr Hrs	3(3+0)
Pre-requisite	SE101 (Software Engineering)
Recommended Texts	1. Software Architecture and Design Illuminated, 2015 byJones and Bartlett Publishers, LLC 2. Clean Architecture: A Craftsman's Guide to Software Structure and Design (Robert C. Martin Series) 1st Edition by Robert C. Martin (Author), 2017, Prentice Hall, ISBN-10: 0134494164, ISBN-13: 978-0134494166 3. Design It!: From Programmer to Software Architect (The Pragmatic Programmers) 1st Edition by Michael Keeling, 2017, Pragmatic Bookshelf, ISBN-10: 1680502093, ISBN-13: 978-1680502091
Course Description	Software Architecture and Design will provide students with the principles and concepts involved in the analysis and design of large software systems and how to apply design principles, patterns, and architectures to create reusable and flexible software applications and systems.
Course Objectives	The learning objectives for the course are the following: Understand and apply object-oriented design techniques Develop and evaluate software architectures Select and use appropriate architectural styles Select and use appropriate software design patterns Express the specifications and design of an application using UML Specify parts of the design using a formal design language (OCL)

Week Wise Distribution of the Contents

Lecture Number	**Topic**

W1	The Role of Software Design Design as a problem-solving process Transferring design knowledge
W2	Constraints upon the design process and product Design in the Software Development Process A context for design
W3	Design Qualities The Architecture Business Cycle
W4	Architectures are influenced by system stakeholders Amplification of influences on an architecture
W5	Software Processes and the Architecture Business Cycle Understanding the requirements
W6	Creating or selecting the architecture
W7	Implementing the system based on the architecture, Ensuring that the implementation conforms to the architecture
W8	ARCHITECTURE ACTIVITIES what Makes a "Good" Architecture
W9	ARCHITECTURE AS A TRANSFERABLE, RE-USABLE MODEL
W10	Why Is Software Architecture Important? less Is More
W11	architectural Structures and Views Component-Based Software Architecture
W12	Architecture of User Interfaces Evolution of User Interfaces Look and Feel (Syntax) of User Interfaces
W13	1D layout , 2D layout, 3D layout: 4D layout: Dynamic Style of User Interfaces Usability (Semantics) of User Interfaces
W14	Design Considerations of User Interfaces Enabling Technology
W15	History of UML UML Class Diagrams Object-Oriented Concepts
W16	UML ─ Building Blocks UML ─ Basic Notations UML ─ Modeling Types

Course Code	SE422
Course Title	Software Project Management
Cr Hrs	3(3+0)
Pre-requisite	None
Recommended Texts	1. Information Technology Project Management, Jack T Marchewka, 5th Edition, WILEY, 2016 ISBN: 978-1118911013

	2. Agile Project Management with Scrum, Ken Schwaber, Microsoft Press, 2004, ISBN-978-0735619937. 3. Software Project Management for Dummies, Teresa Luckey and Joseph Phillips, For Dummies, 2016, ISBN-13: 978-0471749349. 4. The Complete Software Project Manager: Mastering Technology from Planning to Launch and Beyond (Wiley CIO), Anna P. Murray, 2016, 1st Edition, Publisher: Wiley, ISBN-10: 1119161835, ISBN-13: 978-1119161837
Course Description	This course gives you the overview about, what project management in general and software project management in particular. What tools and techniques you are used for the proper handling and management of a software project. The basic composition of the course is based upon routine activities, SOPs and norms of the market and how to go about traditional to latest trends
Course Objectives	This course aims to cover the basics • Deliver successful software projects that support organization's strategic goals • Match organizational needs to the most effective software development model • Plan and manage projects at each stage of the software development life cycle (SDLC) • Create project plans that address real-world management challenges • Develop the skills for tracking and controlling software deliverables

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	What is Software Project Management? Management, Project Management, Software Project Management Scope & Limitations
W2	The WaterFall Model, In theory, In Practice Conventional Software Management Performance Software Economics
W3	Pragmatic Software Cost Estimation Reducing Software Product Size Improving Software Processes
W4	Improving Team Effectiveness Improving Automation through Software Environments Achieving Required Quality
W5	Peer Inspection: A Pragmatic Approach Engineering and Production Phases Inception Phase
W6	Elaboration phase Construction Phase Transition phase
W7	Work Breakdown Structure Project Team Structure, Team Structure & responsibilities, Advantages of Team Structure
W8	Organizational Structure, Functional, Matrix, Projectized Project Manager
W9	Deputy Project Manager

	Project Engineer
W10	Line of Business Organizations, SEPA, PRA, SEEA
W11	Project Organizations: Software Management Team, Software Architecture Team, Software Development Team, Software Assessment Team
W12	Evolution of Organization Request for Proposal (RFP)
W13	Contracts, Types of Contracts
W14	Feasibility Report Project Report
W15	Configuration Management
W16	Gantt Chart Pert Chart, COCOMO Model Critical Path Method (CPM)

Course Code	CS364
Course Title	**Human Computer Interaction**
Cr Hrs	3(3+0)
Pre-requisite	None
Recommended Texts	1. Human-Computer Interaction: Fundamentals and Practice, Gerard Jounghyun Kim, Auerbach Publications, 2015, ISBN: 9781482233896 2. Human Computer Interaction (3rd Ed) By Alan Dix.Janet Finlay, ISBN: 9780130461094. 3. Usability Engineering: Scenario-Based Development of Human Computer Interaction by Mary Rosson, John Carroll, Mary Beth Rosson ISBN : 978-1558607125

Course Description

Human-Computer Interaction (HCI) is the study of the principles and methods with which one builds effective interfaces for users. A basic precept of HCI is that users should be able to focus on solving problems, rather than dealing with the intricacies of complex software. Interfaces must be accessible, meaningful, visually consistent, comprehensive, accurate, and oriented around the tasks that users tend to perform. The course will provide a balance of practical and theoretical knowledge, giving students experience ordinarily not provided by other courses in computer science.

Course Objectives

To evaluate software user interfaces using heuristic evaluation and user observation techniques. Conduct simple formal experiments to evaluate usability hypotheses.

- Apply user centered design and usability engineering principles as they design a wide variety of software user interfaces.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction to HCI, The Goals of HCI Usability, HCI & its Evolution Input / Output Channel
W2	Human Senses and Characteristics Eyesight, Hearing, Touch, Human Memory, Emotions etc. The Computer, Text Entry Devices
W3	Positioning, Pointing & Drawing Devices, Display, Devices

W4	Physical Controls, sensors & special devices The Interaction, What are Goals, Domain, Intention
W5	Task Analysis, User& Task Language, and Intention Models of Interaction, The Execution-evaluation cycle, The interaction
W6	Frame Work, Ergonomics, Interaction Styles, Command line, Menus, WIMP, Natural Language, Point & Click, and Forms Fill-Ins etc
W7	Elements of WIMP, Paradigm & Interaction Design Basics, Paradigm Introduction, Paradigm for Interaction, Time Sharing, VDUs
W8	Programming Toolkit, Personal Language Vs Actions, Hypertexts, Multimodality, Computing, Direct Manipulations
W9	Computer supported Cooperative Works, The WWW, Agent Based Interfaces, Ubiquitous Computing
W10	Sensor Based and Context aware Interaction, Interaction Design Basics, The Process of design, Design Goals
W11	Constraints and Trade-Offs, Golden Rule of Design, User Focus, Scenario
W12	Navigation Design, Screen design and layout Iteration & Prototype, Design Rules, Principles to support usability, Standards
W13	Guidelines, Golden Rules & Heuristics, HCI Patterns
W14	Evaluation Techniques, What is Evaluation?, Goals of Evaluation, Evaluation through expert analysis
W15	Evaluation through user participation, Observational Techniques, Choosing an Evaluation method
W16	Universal Design, Universal Design Principles, Multi-Modal Interaction, Design for diversity

Course Code Course	SE411
Title	Software Re-Engineering
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Object Oriented Reengineering Patterns (Serge Demeyer, Stéphane Ducasse, Oscar Marius Nierstrasz, 2016) 2. Re-Engineering Legacy Software (Chris Birchall, 2016) 3. Software Evolution and Maintenance (Kshirasagar Naik and Priyadarshi Tripathy, 2016)
Course Description:	This course explains the 'state-of-the-art' reengineering existing software systems. This includes an introduction to the recent research, as well as an overview of the principles techniques and skills applied in practice today. Students will acquire a range of principles, techniques and skills that are currently being used for reengineering existing software systems. Consequently, the course has a practical ring to it with a minimal theoretical content (taught as reengineering patterns), several lab-sessions (experimenting with a suite of reengineering tools) and one project (restructuring an existing large software system).
Course Objectives	• Assess which parts should be reengineered first, identify the risks and opportunities for a given re-engineering project and extract coarse-grained and fine-grained design models;

	- Exploit tests during re-engineering and select the most appropriate migration strategy; - Solve the typical problems of an object-oriented re-engineering project
Course Outline	- Software life-cycle and examination of Software Process Models - Decision making and economics of software re-engineering - Program analysis and program understanding - Source code representation at higher level of abstraction - Discussion on CASE tools - Software metrics and their use in software evolution - Evaluation of the re-engineering product - Re-engineering economics - Software integration issues

Week Wise Distribution of the Contents

Lecture Number	**Topic**
W1	Software life-cycle and examination of Software Process Models Role of software maintenance and evolution in a product's life cycle
W2	Decision making and economics of software re-engineering Program analysis and program understanding
W3	Source code representation at higher level of abstraction Parsers and Abstract Syntax Trees, Control and Data Flow Graphs, Program Dependency Graphs
W4	Discussion on CASE tools Data flow analysis techniques Data Flow Analysis applications to slicing, change/impact analysis, re-documentation and, application partitioning
W5	Software metrics and their use in software evolution Software migration techniques Evaluation of the re-engineering product
W6	Quality metrics, metrics for progress and productivity Program Comprehension
W7	Re-engineering economics Software integration issues Software Cost Modeling
W8	Software Cost Computing System Evolution
W9	Software Re- engineering in Maintenance Data re-engineering in Maintenance Software Reuse and Reuse Landscape
W10	Software Reusability Definition, Problems Benefits Approaches To Reuse
W11	Software Reuse and Maintainability Issues Design Patterns, Frameworks, Program Generators, COTS, Reuse
W12	Aspect-Oriented Development Product Lines, Web-Services Software Quality Measures
W13	Types of measures: Size –Oriented Metrics, Metrics For Source Code, Metrics For Testing, Metrics For Maintenance, Metrics For Design, Metrics For Specification
W14	Pitfall and tricks of code analysis Identifying calling conversions

W15	Software Types: S-Type, P-Type
W16	Lehman's laws

Course Code	**SE315**
Course Title	**Software Construction & Development**
Cr Hrs	3 (3+0)
Pre-requisite	Nil
Recommended Texts	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 Description	This course gives students experience designing, implementing, testing, and debugging large programs. Students will also get advanced Java programming experience; covering topics such as inheritance, multithreading, networking, database programming, and web development.
Course Objectives	At the end of the course the students will be able to: • Understand the role of design and its major activities within • the OO software development process, with focus on the Unified process. • Develop Object-oriented design models and refine them to reflect implementation details • Evaluate different architectures for a medium size software. • Implement design model using an object-oriented programming language.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Software development process, Software engineering process infrastructure
W2	Software engineering process improvement, Systems engineering life cycle model
W3	Process implementation, Levels of process definition
W4	Life cycle model characteristics, Individual and team software process
W5	Lehman’s Laws, code salvaging, and configuration management
W6	Martin Fowler’s refactoring concepts and their application to small projects
W7	Apply Michael Feathers’ “legacy code” concepts
W8	Exception handling, making methods robust by having them check their inputs sent from calling objects
W9	Software configuration management, Release management,
W10	Software configuration management processes
W11	Software deployment processes, Distribution and backup, Evolution processes and activities

W12	Basic concepts of evolution and maintenance, Working with legacy systems
W13	Refactoring, Error handling
W14	exception handling, and fault tolerance
W15	Personal reviews (design, code, etc.)
W16	Peer reviews (inspections, walkthroughs, etc.)

Course Code	SE312
Course Title	Formal Methods in Software Engineering
Cr Hrs	3(3+0)
Pre-requisite	None
Recommended Texts	1. Software Engineering Mathematics: Formal Methods Demystified, by Jim Woodcock, Martin LoomesISBN-13:978-0748408139 2. Concise Guide to Formal Methods: Theory, Fundamentals and Industry Applications, Gerard O'Regan, 2017, 1st edition, ISBN-10: 3319640208, ISBN-13: 978-3319640204 3. Formal Methods in Software Engineering (2011), Publish by FTMS Consultants (M) SdnBhd Kuala Lumpur, Malays 4. Designing Reliable Distributed Systems: A Formal Methods Approach Based on Executable Modeling in Maude (Undergraduate Topics in Computer Science) 1st ed. 2017 Edition by Peter Csaba Ölveczky, 2017, 4th edition, springer, ISBN-10: 1447166868, ISBN-13: 978-1447166863
Course Description:	The applied mathematics of computer system engineering used to specify and model the behavior of a system and to mathematically verify that the system design and implementation satisfy system functional and safety properties.
Course Objectives	Students will learn to understand how formal methods (FM) help produce high-quality software by learning about formal modeling and specification languages. They would write and understand formal requirement specifications, Learn about main approaches in formal software verification, know which formal methods to use and when to use automated and interactive tools to validate models and code.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	What are Formal Methods? Why Consider Formal Methods? Formal methods in software engineering Scope and method, Formal Methods Concepts
W2	Classification of formal methods Specify and analyses Specify and prove, Specify and derive
W3	Specify and transform Architecture based Systems classification

W4	Asynchronous or synchronous hardware, analogue or digital hardware Mono- or multi-processor systems, Imperative/functional/logic-based/object-oriented software
W5	Multi-threaded or sequential software, Conventional or real-time operating systems
W6	Embedded systems or local systems or distributed systems
W7	Interaction based Systems classification Transformational systems Interactive systems
W8	Reactive systems System properties Limitations of formal methods
W9	Proposition Introduction to Proposition, Proposition Operators Introduction to Truth Table, Truth Table and Proposition
W10	Introduction to Formal Methods Critical Software, Integrity Level Stages in Formal Methods Predicates, Introduction Existential
W11	Universal Sets Universe, Elements
W12	Cardinality Sets Relationship
W13	Sets Operation UML and Formal Methods
W14	The Object Constraint Language (OCL) Algebraic Specifications, Specifications of abstract data types Completeness, Axioms and term rewriting
W15	Modularity and re-usability Model-based specifications
W16	The Z (Zed) specification Language Z Schemas, Seven Myths of Formal Methods

Course Code	SE441
Course Title	**Software Testing**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Software Testing: A Practical Approach, SANDEEP DESAI ABHISHEK SRIVASTAVA, 2nd edition, PHI Learning Pvt LTD, 2016, ISBN: 978-81-203-5226-1 2. Foundations of Software Testing by by Cem Kaner, Rebecca L Fiedler, 2013 by Context-Driven Press, 2013, ISBN-13: 978-0989811927

	3. Software Testing: Concepts and Operations,Ali Mili, Fairouz Tchier,ISBN: 978-1-118-66287-8,July 2015, ISBN: 978-1-118-66287-8
Course Description	The course covers the important software testing by studying various software testing techniques in details.
Course Objectives	- To find defects which may get created by the programmer while developing the software. - To gain confidence in and providing information about the level of quality. - To prevent defects, to make sure that the end result meets the business and user requirements and also to gain the confidence of the customers by providing them a quality product.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Software Quality Attributes Boolean Attributes, Statistical Attributes
W2	Operational Attributes Usability Attributes Business Attributes
W3	Structural Attributes Why is testing necessary? What is testing? Testing Principles
W4	Testing Objectives Fundamental test Process-I Fundamental test process-II
W5	Psychology of testing Testers and code of ethics Paradoxes and main principles Types of tests Test and maintenance.
W6	Test organization Test planning and estimation Test progress monitoring and control Who does Testing?, When to Start Testing?
W7	When to Stop Testing? Verification & Validation MYTHS: Myth 1: Testing is Too Expensive, Myth 2: Testing is Time Consuming
W8	Only Fully Developed Products are Tested, Myth 4: Complete Testing is Possible
W9	Myth 5: A Tested Software is Bug, Myth 6: Missed Defects are due to Testers
W10	Myth 7: Testers are Responsible for Quality of Product Myth 8: Test Automation should be used Wherever Possible to Reduce Time Myth 9: Anyone can Test a Software Application Myth 10: A Tester's Only Task is to Find Bugs
W11	QA, QC, AND TESTING Testing, Quality Assurance, and Quality Control
W12	Audit and Inspection Testing and Debugging ISO STANDARDS, ISO/IEC 9126

W13	ISO/IEC 9241 ISO/IEC 25000:2005 TYPES OF TESTING, Manual Testing
W14	Automation Testing, What to Automate? When to Automate?, How to Automate? Software Testing Tools,
W15	Testing Methods, Black Box Testing White Box Testing, Grey Box Testing White Box Testing, Grey Box Testing
W16	Integration Testing, System Testing Regression Testing, Acceptance Testing

Course Code	**CS457**
Course Title	**Simulation and Modeling**
Cr Hrs	3(3+0)
Pre-requisite	CS251 (Software Engineering)
Recommended Texts	1. Simulation Modeling and Arena, Manuel D. Rossetti, 2nd Edition, Wiley, 2015, ISBN: 978-1-118-60791-6 2. Simulation Modeling and Analysis, Averill Law, 5th Edition, Mc-Graw Hill EDUCATION, 2015, ISBN: 978-0071255196 3. B. P. Zeigler, Theory of Modelling and Simulation, Academic Press, second edition, 2000, ISBN 9780127784557 4. Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, Bernard P. Zeigler, Herbert Praehofer, Tag Gon Kim, Academic Press, 2000, ISBN-13: 978-0127784557
Course Description	By the end of this course, student should have a deep understanding of the concepts of modeling and simulation of dynamic systems using a variety of formalisms. Student should be able to build modeling and simulation systems. This will give student ample background to understand and use existing modeling and simulation systems
Course Objectives	The purpose of this course is to provide • Students with an opportunity to develop skills in modeling and simulating a variety of management-related problems. • After learning the simulation techniques, the students are expected to be able to solve real world problems which cannot be solved strictly by mathematical approaches.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Basic Simulation Modeling, The Nature of Simulation Systems, Models, and Simulation Discrete-Event Simulation

W2	Simulation of a Single-Server Queueing System, Simulation of an Inventory System Parallel/Distributed Simulation and the High Level Architecture Steps in a Sound Simulation Study, Other Types of Simulation , Advantages, Disadvantages, and Pitfalls of Simulation
W3	Modeling Complex Systems List Processing in Simulation Assessing the Homogeneity of Different Data Sets
W4	Single-Server Queueing Simulation with simlib, Time-Shared Computer Model Multiteller Bank with Jockeying, Job-Shop Model Simple Simulation Language: simlib
W5	Efficient Event-List Manipulation Simulation Software, Comparison of Simulation Packages with Programming Languages Classification of Simulation Software, Desirable Software Features
W6	General-Purpose Simulation Packages Object-Oriented Simulation Review of Basic Probability and Statistic
W7	Random Variables and Their Properties Simulation Output Data and Stochastic Processes Estimation of Means, Variances, and Correlations
W8	Confidence Intervals and Hypothesis Tests for the Mean The Strong Law of Large Numbers The Danger of Replacing a Probability Distribution by its Mean
W9	Comments on Covariance-Stationary Processes Building Valid, Credible, and Appropriately Detailed Simulation Model
W10	Guidelines for Determining the Level of Model Detail , Verification of Simulation Computer Program
W11	Techniques for Increasing Model Validity and Credibility Management’s Role in the Simulation Process
W12	Statistical Procedures for Comparing Real-World, Observations and Simulation Output Data Random-Number Generator
W13	Selecting Input Probability Distribution Useful Probability Distributions, Techniques for Assessing Sample Independence
W14	Activity I: Hypothesizing Families of Distributions, Activity II: Estimation of Parameters
W15	Activity III: Determining How Representative the Fitted Distributions The ExpertFit Software and an Extended Example
W16	Shifted and Truncated Distribution Bézier Distributions Specifying Multivariate Distributions, Correlations, and Stochastic Processes

Course Code	**CS342**
Course Title	**Web Designing**
Cr Hrs	4(3+1)
Pre-requisite	CS113 (Object Oriented Programming)
Recommended Texts	1. Web Design with HTML, CSS, JavaScript and jQuery Set 1st Edition, Jon Duckett, Wiley, 2014, ISBN-13: 978-1118907443 2. Web Design All-in-One For Dummies 2nd Edition, Sue Jenkings, For Dummies, 2013, ISBN-13: 978-1118404102 3. Head First HTML5 Programming, Eric Freeman and Elisabeth Robson, 2011, ISBN: 978-1-449-39054-9 4. How to Do Everything with JavaScript, Scott Duffy, 2013, ISBN-0-07-222887-3. 5. A Practical Guide to Designing for the Web, Mark Boulton Design, Limited, 2009, ISBN-13: 978-0956174017

Course Description

This course introduces students to basic web design using HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets). The course does not require any prior knowledge of HTML or web design. Throughout the course students are introduced to planning and designing effective web pages; implementing web pages by writing HTML and CSS code; enhancing web pages with the use of page layout techniques, text formatting, graphics, images, and multimedia; and producing a functional, multi-page website.

Course Objectives

- Recognize and understand HTML web page elements
- Know how to write HTML code
- Understand and apply effective web design principles
- Enhance web pages using text formatting, color, graphics, images, and multimedia
- Incorporate forms into web pages
- Understand and apply CSS to format web page elements
- Adding scripting language to add action to HTML.
- To make the design and contents of a website optimized for search engines called Search Engine Optimization (SEO)

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction to the Course Hyper Text Markup Language (HTML) Setting Environment for HTML (Software) Getting started with tags HTML Elements Paired and Singular Tags HTML, Head, Title, Body tags Writing simplest web page How to save webpage How to view webpage HTML Paragraphs, Headings HTML Attributes HTML Comments

W2	HTML Formatting Bold, Emphasize, Italic, Underline, Marked, Small, Deleted, Inserted, Subscript SuperScript, Code, Preformatted, Abbreviation, Quotations, Cite, Address etc HTML Colors Standard Colors, RGB Colors, HexaDecimal Colors Linking Documents Images Linked Images HTML Lists Ordered Lists Un-Ordered Lists
W3	HTML Table HTML Forms Form Methods Form Elements HTML5 Form Elements
W4	Meta Tag Inline Frames (IFrame) Div and Span HTML Media Video Audio Plug-Ins HTML Graphics HTML Convas
W5	HTML API's Geolocation HTML Drag-Drop HTML Local Storage App Cache
W6	App Cache HTML Web Workers Cascaded StyelSheet (CSS) The Style Attribute CSS Syntax Inlined CSS Internal CSS External CSS
W7	CSS Selectors Tags, Class, Id, Pseudo Selectors, Attribute Selectors CSS3 Selectors CSS Properties Font CSS Properties Text CSS Properties Setting Size (various units of text size) Background CSS Properties Border CSS Properties Links CSS Properties Lists CSS Properties
W8	Box Model in detail Setting Layout Using Display, Position, Float, Width, Height CSS properties for setting layout Using Div for Layout Setting Introduction to Responsive Design Responsive Design explained with an example

W9	Introduction to Scripting Languages Introduction to Javascript How to add Javascript to HTML Inlined Javascript Internal Javascript External Javascript
W10	Javascript Syntax Data Types Variables Complex Data Types (Arrays, Objects) Operators
W11	Defining Methods in Javascript Selection Statements in Javascript Looping in Javascript Event-driven HTML and Javascript
W12	RegEx Javascript Form Validation
W13	Javascript Debugging Understanding Browser support JS HTML Document Object Model (DOM)
W14	JQuery Basics JQuery in Action JSON Basics
W15	Communicating with Web Server using Javascript Search Engine Optimization (SEO)
W16	Getting website on Internet Understanding how HTTP Works and Request are processed Understanding the overall cycle of how web works What is Domain Name Selecting Domain Names Purchasing Domain Names Purchasing Hosting Space on Shared Web-Server Uploading files on rented space using FTP How to bind your domain name with hosting server Visiting Website Live How to make money from your website by advertisement

Course Code	**CS316**
Course Title	**Visual Programming**
Cr Hrs	3(2+1)
Pre-requisite	CS213 (Object Oriented Programming)
Recommended Texts	1. Introduction to Programming Using Visual Basic (10th Edition), David I. Schneider, Pearson, publisher (April 28, 2016), ISBN: 978-0134542782 2. Beginning Visual C# 2012 Programming by Karli Watson, Jacob Vibe Hammer, Jon Reid and Morgan Skinner, Wrox, 2012, ISBN-13: 978-1118314418. 3. C# For Beginners: The tactical guidebook - Learn CSharp by coding, Jonas Fagerberg, Kindle Edition, 2015, ASIN: B017OAFR8I.

Course Description	Visual programming is a special area in which you will develop programs for MS windows. This course will discuss Microsoft based Windows platform. We will be using C# language as programming tool as it is the top choice for desktop programming and also because the most of Windows Application Programming Interfaces (API)are developed in C# language; furthermore, the documentation available for API by Microsoft also contains C# syntax. We will explore main features of Windows Programming like interface designing, event-based programming, multithreading, handling exceptions, and using GDI etc.
Course Objectives	Describe and explain the functional mechanisms of Windows platform i.e. how it works and the concepts related to basic Windows architecture Develop fine-tuned and robust applications for MS Windows Get hands in specialized and more complex areas of windows programming like network programming, graphics programming etc.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction: Introduction to Visual Studio(VS) .NET IDE, VS installation and setting environment for C# development, Menu Bar and Toolbar
W2	Solution Explorer: Toolbox, Properties Window, Using Help, Simple Program: Displaying Text and an Image Introduction to C# programming: Simple Program: Printing a line of text, Another Simple Program: Adding Integers
W3	Memory Concepts: Arithmetic, Decision Making: Equality and Relational Operators Control Structures: Looping, Making Decision
W4	Control Structures: Arrays in C#, Array Initialization in different ways Different Types of Arrays in C#, Jagged Arrays, Foreach Loop
W5	Object Oriented Programming with C#, Structure of .Net classes Defining Namespaces, Classes, Abstract Classes
W6	Methods: Properties, Defining Access Level, Assemblies, Class View and Object Browser Static Classes, Static Constructors, Static Members, Static vs Instance Members
W7	Exception Handling, Try, Catch, and Finally blocks, C# Exception Classes, Throwing Exception
W8	Graphical User Interface, Windows Forms, Event Handling Model, Delegates Basic Controls of Windows Forms Advance Controls of Windows Forms, Listbox, CheckboxList, RadioList
W9	Multiple Document Interfaces (MDI), MenuStrip, ToolStrip, StatusStrip TreeView Data Access with ADO.Net, Namespaces and Classes for Database access Connecting to database
W10	SqlConnection, SqlCommand, Insert, Update, Delete commands ExecuteNonQuery Fetching Data from database, DataReader, Going through fetched data Displaying data

W11	Binding controls to database, Gridview control, DropDownList, RadioList, CheckBoxList, Editing data in Gridview control Layering and its important, Data Link Layer (DLL), Business Login Layer (BLL) Presentation Layer
W12	Introduction to LINQ and Object Relational Mapping (ORM) Files and Streams, Static Classes FileInfo (Create, Delete) File, Append Text to File
W13	DirectoryInfo(Create, Delete) Directory, Look for files, File Explorer Working with Streams Threads, Thread States: Life Cycle of a Thread, Thread Priorities
W14	Multi-Threading, Delegates, Multi-Threaded Applications Thread Synchronization, ThreadPool
W15	Graphical Device Interface (GDI) and GDI+Drawing basic graphic elements Working with Convas, Complex graphics objects manipulation
W16	Windows Based Services, Files and Registry Operations Windows Services, Windows Service Architecture Creating Windows Services

Course Code	**CS454**
Course Title	**Software Project Management**
Cr Hrs	3(3+0)
Pre-requisite	CS251 (Software Engineering)
Recommended Texts	1. Information Technology Project Management, Jack T Marchewka, 5th Edition, WILEY, 2016 ISBN: 978-1118911013 2. Agile Project Management with Scrum, Ken Schwaber, Microsoft Press, 2004, ISBN-978-0735619937. 3. Software Project Management for Dummies, Teresa Luckey and Joseph Phillips, For Dummies, 2016, ISBN-13: 978-0471749349. 4. The Complete Software Project Manager: Mastering Technology from Planning to Launch and Beyond (Wiley CIO), Anna P. Murray, 2016, 1st Edition, Publisher: Wiley, ISBN-10: 1119161835, ISBN-13: 978-1119161837
Course Description	This course gives you the overview about, what project management in general and software project management in particular. What tools and techniques you are used for the proper handling and management of a software project. The basic composition of the course is based upon routine activities, SOPs and norms of the market and how to go about traditional to latest trends
Course Objectives	This course aims to cover the basics • Deliver successful software projects that support organization's strategic goals • Match organizational needs to the most effective software development model

	• Plan and manage projects at each stage of the software development life cycle (SDLC) • Create project plans that address real-world management challenges • Develop the skills for tracking and controlling software deliverables

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	What is Software Project Management? Management, Project Management, Software Project Management Scope & Limitations
W2	The WaterFall Model: In theory, In Practice Conventional Software Management Performance Software Economics
W3	Pragmatic Software Cost Estimation Reducing Software Product Size Improving Software Processes
W4	Improving Team Effectiveness Improving Automation through Software Environments Achieving Required Quality
W5	Peer Inspection: A Pragmatic Approach Engineering and Production Phases Inception Phase
W6	Elaboration phase Construction Phase Transition phase
W7	Work Breakdown Structure Project Team Structure: Team Structure & responsibilities, Advantages of Team Structure
W8	Organizational Structure: Functional, Matrix, Projectized Project Manager
W9	Deputy Project Manager Project Engineer
W10	Line of Business Organizations: SEPA, PRA, SEEA
W11	Project Organizations: Software Management Team, Software Architecture Team, Software Development Team, Software Assessment Team
W12	Evolution of Organization
W13	Request for Proposal (RFP) Contracts: Types of Contracts
W14	Feasibility Report Project Report
W15	Configuration Management Gantt Chart
W16	Pert Chart Critical Path Method (CPM) COCOMO Model

Graduate Core Courses
(Description and Contents)

Course Code	**CS515**
Course Title	**Theory of Computation**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Introduction to the Theory of Computation 3rd edition, Michael Sipser, Cengage Learning Publisher (2014), ISBN: 978-8131525296 2. Introduction to the theory of Computation, by Michael Sipser, ISBN-13: 978-1-133-18779-0, published by Cengage Learning, 2012 3. Introduction to languages and the theory of computation, by John Martin, ISBN: 0073191469, 2010 4. Introducing The Theory Of Computation, Wayne Goddard, 2008, ISBN-13: 978-0763741259

Course Description

Introduces the foundations of formal language theory, computability, and complexity. Shows relationship between automata and various classes of languages. Addresses the issue of which problems can be solved by computational means, and studies complexity of solutions.

Course Objectives

- To introduce the students to the mathematical foundations of computation including automata theory, the notations of algorithm, decidability, complexity and computability.
- To enhance students ability to understand and conduct mathematical proofs for computation and algorithms

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction Terminologies of Languages Descriptive definition
W2	Examples using Descriptive definition
W3	Recursive Definition Examples using Recursive definition
W4	Regular Expressions(RE) Examples of Regular Expressions(RE)
W5	Finite Automaton(FA) Examples of Finite Automaton(FA)
W6	Martin Technique Non- Deterministic Finite Automata- NFA Conversion of NFA to DFA
W7	Union of Two FAs Concatenation of Two FAs
W8	Transition Graph-TG Generalize Transition Graph- GTG
W9	Context Free Grammar-CFG Examples of CFG
W10	Tree Ambiguity Parsing

W11	Languages that are not context-free: pumping lemma for CFLs Pushdown Automata
W12	Recursively Enumerable Languages Turing machines Examples
W13	Computability Theory Transformations Decidability
W14	Complexity Theory Time and space complexity
W15	P, NP NP-Completeness, reductions
W16	Other complexity classes

Course Code	**CS516**
Course Title	**Advance Analysis of Algorithms**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Predictive Analytics For Dummies , Dr. Anasse Bari , Mohamed Chaouchi ,Tommy Jung , Publisher For Dummies (March 24, 2014), ISBN: 978-1118728963 2. Practical Analysis of Algorithms (Undergraduate Topics in Computer Science) 2014th Edition , Dana Vrajitoru , William Knight, Springer Publisher (September 15, 2014), ISBN: 978-3319098876 3. T. H. Corman, R. L. Rivest, Introduction to Algorithms, MIT Press, 2001, ISBN 978-0262033848, 2009 4. R.Sadgewick and P.Flajolet, Introduction to the Analysis of Algorithms, Addison-Wesley Publishing, 1996, ISBN 978-0321905758 5. Analysis Of Algorithms, Jeffrey McConnell, 2007, ISBN-13: 978-0763707828

Course Description

Algorithms are essentially required for solving a wide range of computational problems from a number of different fields such as Artificial Intelligence, Data Mining, Distributed Systems and Computer Networks. Course contents include the study of algorithm analysis techniques and algorithm design strategies that are currently used in a variety of application domains.

Course Objectives

At the end of this course, students should be able to

- Design and develop new and efficient algorithms
- Analyze existing algorithms and select suitable algorithms for complex problem solving.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction Mathematical Preliminaries, Computational Complexity, Calculating Asymptotic Complexity
W2	Sorting Algorithms Solve recurrence relations
W3, W4	P, NP and NP-Completeness

W5, W6	Understand the concepts of Dynamic programming
W7, W8	Understand the concepts of Greedy Algorithm
W9, W10	Understand the concepts of Graph traversing
W11	Satisfiability Problems Evolutionary Algorithms
W12	Understand the algorithms for computing shortest path, Basics of Complexity theory
W13	Linear Programming
W14	Genetic Algorithms
W15	Approximation Algorithms
W16	Theory of Randomization

Course Code	**CS517**
Course Title	**Advanced Operating Systems**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Andrew S. Tanenbaum and Maarten van Steen. “Distributed Systems: Principles and Paradigms”, Prentice Hall, 2nd Edition, 2007. (Required) 2. Modern Operating Systems, Tanenbaum, Andrew S. Tanenbaum, 2nd Edition, ISBN: 978-0136006633, Prentice Hall (2001)

Course Description

The course will focus on the engineering and performance trade-offs in the design of operating systems. The purpose will be to teach not only what operating systems are and how they work today, but also why they are designed the way they are and how they are likely to evolve in the future. The emphasize will be on the practical aspects of the topics through the case study of Linux kernel as an example of a commercial operating system.

Course Objectives

Upon successful completion of the course the students would be able to:

- Understand in great detail how and why different parts of an operating system work.
- Understand the engineering tradeoffs involved in the design of various sub-modules of an operating system.
- Understand how operating systems are structured, what are alternative OS architectures and how different modules interact together to form a cohesive and complex system.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Review of Operating Systems concepts
W2	Hardware concepts of distributed systems
W3	Software concepts and design issues
W4	Communication in distributed systems
W5	Threads and thread usage Multithreading operating system
W6	Client – server model Implementation of Client-server model

W7	Remote procedure call
W8	Implementation of remote procedure call
W9	Synchronization in distributed systems
W10	Clock synchronization
W11	Mutual exclusion; Election algorithms
W12	Transaction and concurrent control
W13	Deadlock in distributed systems
W14	Processor Allocation
W15	Real – time distributed systems
W16	Distributed file systems

Course Code	**CS531**
Course Title	**Advance Computer Architecture**
Cr Hrs	3(3+0)
Pre-requisite	None
Recommended Texts	1. Computer Organization and Architecture – Designing or Performance, 10th Edition, by Willian Stallings 2. Computer Architecture – A Quantitative Approach, 5th Edition, by John L. Hennessy and David A. Patterson 3. Structured Computer Organization, 5th Edition by Andrew S. Tanenbaum
Course Description	The course covers the topics of organization and architecture of computer systems hardware; instruction set architectures; addressing modes; register transfer notation; processor design and memory systems.
Course Objectives	• To enable students to understand the basic components of a computer system • To enable students to understand the issues that are limiting the performance of processor, caches and memories • To enable students to highlight architecture and organization parameters for processor performance • To enable students to understand the cycle of instruction execution in the pipeline of a processor

Week Wise Distribution of the Contents

Lecture No.	Contents
W1	Organization and Architecture, Structure and Functions
W2	History of Computers, Evolution of Intel x86 Architecture, Embedded Systems, ARM Architecture
W3	Designing for Performance, Multicore, MICs and GPGPUs, Amdahl’s Law, Little Law, Basic measure of computer performance
W4	Computer components, Computer function, Interconnection Structures, Bus Interconnection, Point-to-Point Interconnect, PCI Express

W5	Computer Memory System, Cache Memory Principles, Elements of cache Dsign
W6	Mapping functions, Replacement algorithms, Pentium 4 cache organization
W7	Semiconductor main memory, Error Correction
W8	DDR DRAM, Flash Memory, Newer Nonvolatile Solid-State Memory Technologies
W9	Magnetic Disk, RAID, Solid State Drives
W10	Optical Memory, Magnetic Tape, Machine instruction characteristics
W11	Types of operands, Types of operations, Addressing modes, Instruction format
W12	Processor organization, Register organization, Instruction cycle, Instruction pipelining
W13	Instruction execution characteristics, The use of large register file,
W14	Compiler-based register optimization, Reduced instruction set architecture, RISC pipelining
W15	MIPS R4000, SPARC, RISC vs CISC controversy
W16	Instruction level parallelism, Design issues, ,Intel core microarchitecture, ARM Cortex-A8, Coretex-M3

Course Code	SE501
Course Title	**Advanced Requirements Engineering**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Software Engineering: A Practitioner's Approach, Roger S. Pressman, Bruce R. 2. Maxim, 8th Ed, McGraw-Hill Education, 2015. 3. Object-Oriented Analysis, Design and Implementation, Brahma Dathan, Sarnath 4. Ramnath, 2nd Ed, Universities Press, India, 2014. 5. Software Modeling and Design: UML, Use Cases, Patterns, and Software 6. Architectures, Hassan Gomaa, Cambridge University Press, 2011. 7. Applying UML & Patterns: An Introduction to Object-Oriented Analysis & Design and Iterative Development, Craig Larmen, 3rd Edition. (2004)
Course Description The course will discuss concepts for systematically establishing, defining and managing the requirements for a large, complex, changing and software-intensive systems, from technical, organizational and management perspectives. The course will consider the past, present and future paradigms and methodologies in requirements engineering. The course will cover informal, semi-formal and formal approaches, while striking a balance between theory and practice. The course will involve building models of both requirements engineering process and requirements engineering product, concerning both functional and non-functional goals/requirements/specifications, using a systematic decision-making process.	
Course Objectives	

The main objectives are to:

1. Understand the need for requirements for large-scale systems.
2. Understand the stakeholders involved in requirements engineering.
3. Understand requirements engineering processes.
4. Understand models of requirements.
5. Understand functional requirements.
6. Understand non-functional requirements.
7. Understand scenario analysis
8. Understand object-oriented and goal-oriented requirements engineering

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Software Requirements Fundamentals: Product and process requirements
W2	Functional and non-functional requirements
W3	Emergent properties, Quantifiable requirements
W4	System and software requirements
W5	Requirements Process: Process models, Process actors
W6	Process support and management, Process quality and improvement
W7	Requirements Analysis: Requirements sources, Elicitation techniques
W8	Requirements Analysis: Requirements classification, Conceptual modeling
W9	Architectural design and requirements allocation, Requirements negotiation, Formal analysis
W10	Requirements Specification: System definition document, System requirements document
W11	Software requirements specification
W12	Requirements Validation*:* Requirements reviews, Prototyping, Model validation, Acceptance tests
W13	Practical Considerations: Iterative nature of the requirements process, Change management
W14	Requirements attributes, Requirements tracing, Measuring requirements
W15	Software Requirements Tools
W16	Current research topics in requirement engineering.

Course Code	SE511
Course Title	**Advanced Software System Architecture**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Designing Software Architectures: A Practical Approach (SEI Series in Software Engineering), Humberto Cervantes, Rick Kazman, 1st Edition, Addison-Wesley Professional. (2016) 2. Software Product Lines: Practices and Patterns, P. Clements and L. Northrup, Addison-Wesley. (2001) 3. Software Architecture: Foundations, Theory, and Practice, R. Taylor, N. Medvidović and E.M. Dashofy, John Wiley. (2010)

Course Description

This course introduces students to the architectural patterns and abstractions used in the design of software systems. It covers principles and practices for structuring software—how to model and organize large systems so that they are understandable, efficient, and most importantly extensible. Students analyze, implement, and document software structures utilizing common object-oriented and module-level design patterns. They also consider how these software structures shapes the effectiveness and use of different libraries, frameworks, and APIs, so that they are better able to choose and integrate these tools when developing information systems. Concepts are applied through programming interactive applications using TypeScript. In the end, students will understand the ways that large software systems are structured, and how to consider these structures when making decisions about the development and management of information technology.

Course Objectives

The main objective of this course is to enable students for:

- Model software systems and design their architecture following engineering best practices.
- Recognize common architectural styles in software, and analyze their use in information technologies and for particular problem domains.
- Describe and diagram common software design patterns used in software projects.
- Apply architectural patterns in the development of complex software systems, including through the use of existing frameworks and APIs.
- Assess the utility of a software system's architecture and its feasibility for achieving particular goals

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Quality attributes in the context of architecting.
W2	Qualitative and quantitative assessment of architectures.
W3	Architectural modeling through Architecture Description Languages
W4, W5	System modeling its relation to software architecting
W6	Architecting for evolution and variability
W7	Partitioned and layered architectures
W8	System-of-Systems and Ultra-Large Scale Systems
W9-W10	Software Product Lines and Configurable Software
W11	Self-Adaptive Software
W12	Architectural Description Languages
W13	Feature Modeling
W14	Architecture and Model-Based Testing
W15-W16	Current research topics in software system architecture

Course Code	SE541
Course Title	**Software Testing and Quality Assurance**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Software Quality Assurance: Integrating Testing, Security, and Audit (Internal 2. Audit and IT Audit), Abu Sayed Mahfuz, Auerbach Publications, 2016. 3. Practical Model-Based Testing: A Tools Approach, Mark Utting and Bruno Legeard, 4. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2006.

	5. Software Quality Engineering, Testing, Quality Assurance, and Quantifiable improvements, Jeff Tian, IEEE Computer Society, 2005. 6. Introduction to Software Engineering, P Ammann and J Offutt, Cambridge University Press, 2008.

Course Description

This course explores the goals of quality assurance and quality control activities performed during the life cycle of a software product. It focuses on integrating test processes with agile software development methodologies. Practical exercises give experience of design, specification, execution of tests plus test automation using tools through a mixture of instructor-directed exercises and student research leading to knowledge sharing.

Course Objectives

The main objective of this course is to expose the student to the key concepts and practices in software testing and quality assurance. All the topics are designed to understand the key aspects of software quality assurance relevant to all phases of the software life cycle, alternative approaches to software testing, application of current automated tools, standards, and emerging trends

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Testing techniques
W2, W3	Black Box testing, White Box and Grey Box testing techniques
W4, W5	Quality Assurance planning and execution
W6, W7	Automated testing topics include constructing a framework
W8	scripting techniques
W9	generating a test data
W10, W11	generating test architecture
W12	pre/post-processing
W13	test maintenance
W14	job specific metrics
W15, W16	Current research topics in Software Testing and Quality Assurance.

Course Code	DS501
Course Title	**Tools and Techniques in Data science:**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Python for Data Analysis, 2nd Edition, William McKinney (2018) 2. An Introduction to Statistical Learning with Applications in R, 1st Edition, G. James, D. Witten, T. Hastie and R. Tibshirani (2013) 3. Computational and Inferential Thinking: The Foundations of Data Science, 1st Edition, A. Adhikari and J. DeNero (2017) 4. Data Mining and Analysis: Fundamental Concepts and Algorithms, 1st Edition, M. 5. Zaki & W. Meira, (2013) 6. Data Science from Scratch, 1st Edition, Joel Grus (2015) 7. Doing Data Science, 1st Edition, Cathy O'Neil and Rachel Schutt (2013) 8. Introduction to Data Science. A Python Approach to Concepts, Techniques and

	9. Applications, 1st Edition, Laura Igual. (2017)

Course Description

Basic concepts and programming tools for handling and processing data. Includes data acquisition, cleaning data sources, application of machine learning techniques and data analysis techniques, large-scale computation on a computing cluster.

Course Objectives

The main objective of this course is to introduce the tools and techniques in data science and explore common challenges and techniques used in analysis of data.

Week Wise Distribution of the Contents

Lecture Number	Topic
W1, W2	Introduction to Data Science, Data Science Life cycle & Process (Asking Right Questions, Obtaining Data, Understanding Data, Building Predictive Models, Generating Visualizations)
W3	For Building Data Products, Introduction to Data (Types of Data and Datasets),
W4	Data Quality (Measurement and Data Collection Issues),
W5	Data pre-processing Stages, Aggregation, Sampling,
W6	Dimensionality Reduction
W7	Feature subset selection, Feature creation
W8	Algebraic & Probabilistic View of Data
W9	Introduction to Python Data Science Stack (Python, Numpy, Pandas, Matplotlib)
W10	Relational Algebra & SQL, Scraping
W11	Data Wrangling assessing, structuring, cleaning & munging of data
W12	Basic Descriptive & Exploratory Data Analysis, Introduction to Text Analysis (Stemming, Lemmatization, Bag of Words, TF-IDF),
W13	Introduction to Prediction and Inference (Supervised & Unsupervised) Algorithms
W14	Introduction to Scikit Learn
W15, W16	Bias-Variance Tradeoff, Model Evaluation & Performance Metrics (Accuracy, Contingency Matrix, Precision-Recall, F-1 Score, Lift, etc.), Introduction to Map-Reduce paradigm

Course Code	DS502
Course Title	**Statistical and Mathematical Methods for Data Science**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. 1. Probability and Statistics for Computer Scientists, 2nd Edition, Michael Baron (2013) 2. 2. Linear Algebra and Its Applications, 5th Edition, David C. Lay and Steven R. Lay (2016) 3. 3. Introduction to Linear Algebra, 5th Edition, Gilbert Strang (2016) 4. 4. Probability for Computer Scientists, online Edition, David Forsyth. (2018) 5. Probability And Statistics For Computer Science by David Forsyth

Course Description

This course will introduce students to the statistical and mathematical methods needed in the practice of data analytics. Students will learn the basics of statistics, probability, linear algebra, calculus and optimization techniques relevant to data analytics

Course Objectives

The main objective of this course is to improve students' with their ability to work individually and in team for estimating and testing parameters of appropriate model for the data at hand, and getting results also using the software R for statistical computing, 3. interpreting the final results

Week Wise Distribution of the Contents

Lecture Number	Topic
W1, W2	Probability: Probability basics (axioms of probability, conditional probability, random variables, expectation, independence, etc.),
W3	Multivariate distributions, Maximum a posteriori and maximum likelihood estimation
W4, W5	Statistics: introduction to concentration bounds,
W6	Laws of large numbers
W7	Central limit theorem, minimum mean-squared error estimation, confidence intervals
W8	Linear algebra: Vector spaces, Projections (will also cover the least regression),
W9, W10	Linear transformations
W11, W12	singular value decomposition (this substitute for PCA), eigen decomposition, power method
W13	Optimization: Matrix calculus with Lagrange Multipliers
W14, W15	Gradient descent, coordinate descent
W16	introduction to convex optimization

Course Code	CS661
Course Title	**Advance Machine Learning**
Cr Hrs	3(3+0)
Pre-requisite	Nil
Recommended Texts	1. Elements of Statistical Learning (2001) 2. Pattern Recognition & Machine Learning, 1st Edition, Chris Bishop (2011) 3. Machine Learning: A Probabilistic Perspective, 1st Edition, Kevin R Murphy (2012) 4. Applied Machine Learning, online Edition, David Forsyth, http://luthuli.cs.uiuc.edu/~daf/courses/LearningCourse17/learning-book-6-April-nn- revision.pdf

Course Description

This course introduces several fundamental concepts and methods for machine learning. The objective is to familiarize the audience with basic and advanced learning algorithms and techniques and their applications, as well as general questions related to analyzing and handling large data sets. Several software libraries and data sets publicly available will be used to illustrate the application of these algorithms.

Course Objectives

The main goal of this course is to help students learn, understand, and practice advanced machine learning approaches, which include the study of modern computing scaling up with machine learning techniques focusing on industry applications. Mainly the course objectives are: conceptualization and

summarization of machine learning, machine learning techniques, and scaling up machine learning approaches

Week Wise Distribution of the Contents

Lecture Number	Topic
W1	Introduction to machine learning and statistical pattern recognition
W2	Supervised learning: Part I (Graphical models (full Bayes, Naïve Bayes),
W3	Decision trees for classification & regression for both categorical & numerical data
W4	Ensemble methods, Random forests, Boosting (Adaboost and Xgboost), Stacking;
W5	Part II (Four Components of Machine Learning Algorithm (Hypothesis, Loss Functions, Derivatives and Optimization Algorithms)
W6	Gradient Descent, Stochastic Gradient Descent
W7	Linear Regression, Nonlinear Regression, Perceptron
W8	Support vector machines, Kernel Methods
W9	Logistic Regression, Softmax, Neural networks
W10, W11	Unsupervised learning: K-means, Density Based Clustering Methods (DBSCAN, etc.)
W12	Gaussian mixture models, EM algorithm,
W13	Reinforcement learning
W14	Tuning model complexity, Bias-Variance Tradeoff
W15	Grid Search, Random Search
W16	Evaluation Metrics; Reporting predictive performance