SHAHEED BENAZIR BHUTTO WOMEN UNIVERSITY PESHAWAR

CURRICULUM 2023

COMPUTER SCIENCE

BS (4-YEAR PROGRAM)

Main Campus Shaheed Benazir Bhutto Women University, Charsadda Raod, Landay Sarrak Laramma Peshawar. Phone number: 091-9224769

SHAHEED BENAZIR BHUTTO WOMEN UNIVERSITYPESHAWAR

CURRICULUM OF
BS (CS) AND
BS SOFTWARE ENGINEERING

-

SHAHEED BENAZIR BHUTTO WOMEN UNIVERSITY

PESHAWAR

TITLE:

CURRICULUM COMPUTER SCIENCE – BS PROGRAM‖

Approved from Statutory Bodies:

Board of Studies in June 2022 and by Circulation in Sept 2023
In 12th meeting of Board of Faculty held on October 2023
In 19th meeting of Academic Council held on Nov 2023

COMPILED BY:

Dr. Fouzia Jabeen
In charge department of Computer Science

Facilitated by: Dr. Fouzia Idrees, BS Programme Coordinator Deptt CS,

Dr.Tabinda Salam, Advanced Studies Coordinator Deptt CS,

Dr. Brekhna Saif, Associate Degree Programme Coordinator, and

University Curriculum Revamp Committee

OBTAINABLE FROM: Academics Section

Shaheed Benazir Bhutto Women University, Peshawar.

Landay Sarrak Charsadda Road Larammah Peshawar.

Phone Number: 091-9239297.

CURRICULUM

OF

BS – COMPUTER SCIENCE (4-YEAR PROGRAM)

BS – SOFTWARE ENGINEERING (4-YEAR PROGRAM)

ACADEMICS SECTION

SHAHEED BENAZIR BHUTTO WOMEN UNIVERSITY PESHAWAR

CONTENTS

SHAHEED BENAZIR BHUTTO WOMEN UNIVERSITY PESHAWAR

UNIVERSITY AT A GLANCE

Shaheed Benazir Bhutto Women University Peshawar is a premier women university of Khyber Pakhtunkhwa. It has earned this position by virtue of its futuristic outlook towards higher education, strong emphasis on need-based research and focus on innovation and entrepreneurship. Its academic programs are designed to meet the national needs and challenges of the new millennium. While traditional fields of Social, Biological and Physical Sciences have been updated with emerging trends, modern disciplines are being offered to prepare professionals to manage the ever-growing demands of knowledge economy with requisite degree of expertise.

This university is the first ever female university which was established in accordance with the Frontier Women University Act 2004, passed by the Provincial Assembly and assented by the Governor Khyber Pakhtunkhwa on 7th February, 2005. However, according to the revised Act (Khyber Pakhtunkhwa Act No. XI 2010), passedby the provincial assembly Khyber Pakhtunkhwa on September 6, 2010, the University was renamed as Shaheed Benazir Bhutto Women University. The University is destinedto be a leading public sector Women University to impart education to the femalepopulation of this region in order to develop scientific, socio cultural, economic and political stability, through learner centered teaching and research, while strengthening theidentity of the students at national and international level.

Shaheed Benazir Bhutto Women University has come a long way in developing as a global center of excellence for imparting higher education. The universities at large have assumed the role of drivers of knowledge-based regional development. In

contemporary times, the transformation in the world economy is perennial; technologies evolve at neck breaking speeds. These are extra ordinary times requiring extra ordinary preparations and efforts.

VISION OF THE UNIVERSITY

Shaheed Benazir Bhutto Women University aspires for excellence in learning, education, creativity, research and innovation.

MISSION STATEMENT OF THE UNIVERSITY

The mission of Shaheed Benazir Bhutto Women University is to contribute to the society through transformative powers of education, creativity and research with a focus on diversity, linkages, entrepreneurship and innovation. We aim to prepare individuals with problem solving attitude, humanistic outlook, critical thinking and the ability to respond to socio-economic challenges.

SHAHEED BENAZIR BHUTTO WOMEN UNIVERSITY PESHAWAR

DEPARTMENT OF COMPUTER SCIENCE

INTRODUCTION

Department of Computer Science is one of the pioneer department of Shaheed Benazir Bhutto Women University Peshawar. Our goal is the education and training of a diverse body of students who can lead this information technology revolution. To this end, the computing programs orient students toward the pragmatic aspects of computing and provide the learning and practices to make them proficient computing professionals. Computational thinking is rooted in solid mathematics and science, and grounding in these fundamentals is essential. Our laboratory environment exposes students to many commercial software tools and systems, and introduces modern software development techniques. In the context of the practice of computing, this early grounding forms the basis for an education that prepares students for a computing career. The Bachelor of Science in Computer Science program has been offered at SBBWU since 2005. Our goal has been and continues to be a high quality degree program thatensures that students will be able to integrate theory and practice, recognize the importance of abstraction and appreciate the value of efficient design created to meetclearly developed requirements. The program is intended to prepare students for lifelong learning as they undertake professional careers in computing.

VISION OF THE DEPARTMENT

The Department of Computer Science, SBBWUP will be known as an international leader in computing and research, education, and innovation and will generate competent professionals to become part of the industry and research organizations at the national and international levels.

MISSION STATEMENT OF THE DEPARTMENT

To create, share, and apply knowledge in Computer Science, including in interdisciplinary areas that extend the scope of Computer Science and benefit humanity. Imparting the skills necessary to continue education to grow professionally and contribute positively to the global economic well-being and also inculcating in them good ethics, social responsibility and a humanistic approach.

Curriculum has been revised according to HEC curriculum document for BSCS and BSSE by following two committees:

1. Department Curriculum Committee
2. Curriculum Revamp Committee of university

Following are members of BOS

1. Dr. Saeed Mehfooz, Professor, University of Peshawar
2. Dr. Sara Shehzad, Professor, University of Peshawar
3. Dr. Zahoor Jan, Associate Professor, Islamia University, Peshawar.
4. Dr. Brekhna, Assistant Professor, SBBWUP
5. Dr. Sara Mumtaz, Assistant Professor, SBBWUP
6. Dr. Fouzia Jabeen, Lecturer, SBBWUP
7. Dr. Tabinda Salam, Lecturer, SBBWUP

SHAHEED BENAZIR BHUTTO WOMEN UNIVERSITY PESHAWAR
DEPARTMENT OF COMPUTER SCIENCE

Department Curriculum Committee

1. Dr. Fouzia Jabeen
Incharge Deptt CS

2. Dr. Tabinda Salam
Advanced Studies Coordinator

3. Dr. Fouzia Idrees
BS Coordinator

4. Dr. Brekhna
Associate degree Coordinator

Curriculum Revamp Committee

Dr.Farhat Amin. Associate Professor
Department of Bioinformatics, SBBWU
(Convener)

Ms.Sadia Nazeer (Member)
Assistant Professor
Department of English. SBBWU

Dr. Soofia Iftikhar (Member)
Assistant Professor
Department of Statistics, SBBWU

Dr. Samra Kiran (Member)
Assistant Professor
Department of Management Science, SBBWU

Dr. Rehana Masood (Member)
Assistant Professor
Department of Biochemistry, SBBWU

Ms.Mehwish Asmat Ullah (Member)
Deputy Director.
Quality Enhancement Cell, SBBWU

Ms.Tashfeen Zia (Member)
Deputy Director.
Affiliation and Monitoring, SBBWU

Dr.Rubi Bilal (Secretary)
Controller of Examinations, SBBWU

Dr.Safia Ahmed (T.I)
Dean Faculty of Sciences & Social Science, SBBWU

Curriculum for Bachelor Degrees in Computing

Introduction

The objective of the bachelor degrees in computing program is to produce well-rounded graduates, having a strong foundation in theoretical concepts and skills to design and implement complex software using multiple programming technologies under different operating systems and backend technologies. Also, strong academic preparation to pursue careers in local and international IT industry where they can communicate effectively and to continue seeking education through formal or informal methods.

Bachelor Degree Programs in Computing

Computer Science (BS-CS)
Software Engineering (BS-SE)

Eligibility Criteria, Duration of the Program and Award of Degree:

- Minimum 50% marks in Intermediate/12 years schooling/A- Level (HSSC) or Equivalent withMathematics are required for admission in all BS Computing Programs other than BSComputing Engineering.
  **Equivalency certificate by IBCC will be required in case of education from some other country or system.*
- Minimum 60% marks in Intermediate/12 years schooling/A- Level (HSSC) or Equivalent withMathematics are required for admission in BS Computer Engineering Program.
- The students who have not studied Mathematics at intermediate level have to pass deficiencycourses of Mathematics (06 credits) in first two semesters.
- At minimum 130 credit hours are required for award of BS degrees in any computing discipline mentioned in this document.
- The minimum duration for completion of BS Computing degrees is four years. The HEC allows maximum period of seven years to complete BS degree requirements.
- A minimum 2.0 CGPA (Cumulative Grade Point Average) on a scale of 4.0 is required for award of BS Computing Degree.
- The students after successful completion of 04 semesters in BS Computing Programs may exitwith Associate Degree in Computing subject to completion of all requirements for the awardof associate degree, i.e., Credit Hours, CGPA, and compulsory courses.
- Minimum credit hours between 130 to 144 for BS (CS, SE) programs.
- Each program comprises eight semesters spread over four years.

Program Learning Outcomes (PLOs)

Computing programs prepare students to attain educational objectives by ensuring that students demonstrate achievement of the following outcomes (derived from Graduate Attributes define by Seoul Accord www.seoulaccord.org).

Program Learning Outcomes (PLOs)	Computing Professional Graduate

1. **Academic Education**	To prepare graduates as computing professionals
2. **Knowledge for Solving Computing Problems**	Apply knowledge of computing fundamentals, knowledge of a computing specialization, and mathematics, science, and domain knowledge appropriate for the computing specialization to the
	abstraction and conceptualization of computing models from defined problems and requirements
3. **Problem Analysis**	Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines
4. **Design/ Development of Solutions**	Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations
5. **Modern Tool Usage**	Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations
6. **Individual and Team Work**	Function effectively as an individual and as a member or leader in diverse teams and in multi-disciplinary settings
7. **Communication**	Communicate effectively with the computing community and with society at large about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions

8. **Computing Professionalism and Society**	Understand and assess societal, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice
9. **Ethics**	Understand and commit to professional ethics, responsibilities, and norms of professional computing practice
10. **Life-long Learning**	Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional

BS Computer Science

Computer Science Program BS (CS)

A complete detail of BS Program in CS involving Program structure and distribution of credits among various components of Program are discussed in the following pages.

Development in Computer Science

Recent developments in computer hardware, software and communication technologies have offered new exciting opportunities and challenges for creation of innovative learning environments for Computer Science and its curricula design. One of the key elements here is to prepare the graduates for the future. The challenge of getting all newly emerging technologies incorporated in to the curriculum is becoming pivotal for the effectiveness of curricula. There is a need for curricula structures that are really able to grow as we put new demands on them. The curriculum is required to provide integration of all components and the foundations that allow accessing all of the new knowledge and technology to fulfil the vision of future.

The basic intention of an academic Program in Computer Science is to develop the student's critical professional thinking and intuition. The curriculum must be structured to provide a balanced mixture of theory and practical experiences at foundation and advance levels to make the graduate capable of sound professional decisions. As a result the graduate should be able to assume responsible positions in business, government, and education at the research, development, and planning levels. The Program should also provide an excellent foundation for further formal learning and training. The Computer Science curriculum is expected to provide environments to put into practice, the principles and techniques learnt during the course of implementation of academic Program.

The following summarizes some key characteristics for consideration as a basis of a successful academic Program in Computer Science:

1. The Program should provide a broad understanding of the field via introducing concepts, theory, and techniques.
2. Intensive education/training in focused areas of Computer Science is desirable.
3. The Program may encourage students to develop and use abstract models in addition to apply respective technology in practical situations.
4. Computer Science graduates require special communication skills both orally and in writing. They must be able to produce well-organized reports, which clearly delineate objectives, methods of solution, results, and conclusions for a complex task.
5. The Program should provide formal foundations for higher learning.
6. The Program should be dynamic and flexible enough to maintain currency with the latest scientific and technological developments in the field.
7. The Program should provide professional orientation to prepare students for industry.

Program Structure - BS Computer Science

Computer science is the study of the theory, experimentation, and engineering that form the basis for the design and use of computers. It is the scientific and practical approach to computation and its applications and the systematic study of the feasibility, structure, expression, and mechanization of the methodical procedures (or algorithms) that underlie the acquisition, representation, processing, storage, communication of, and access to information [ref WordNet Princeton definition].

Computer Science is the application of a systematic, disciplined and quantifiable approach to the design, development, operation, and maintenance of software systems. It is in fact the practice of designing and implementing large, reliable, efficient and economical software by applying the principles and practices of engineering. The program aims to train students in all aspects of software life cycle from specification through analysis and design to testing, maintenance and evaluation of software product.

Coverage of ACM Knowledge Areas

Computer Science curriculum is designed keeping in view following identified knowledge areas of ACM [ref # ACM 2013 curriculum report]. It has been tried to reasonably cover all knowledge areas without compromising the flexibility needed for a national model curriculum.

- AL - Algorithms and Complexity
- AR - Architecture and Organization
- CN - Computational Science
- DS - Discrete Structures
- GV - Graphics and Visual Computing
- HCI - Human-Computer Interaction
- IAS - Information Assurance and Security
- IM - Information Management
- IS - Intelligent Systems
- NC - Networking and Communications
- OS - Operating Systems
- PBD - Platform-based Development
- PD - Parallel and Distributed Computing
- PL - Programming Languages
- SDF - Software Development Fundamentals
- SE - Software Engineering
- SF - Systems Fundamentals
- SP - Social Issues and Professional Issues

Proposed Curriculum for BS-CS

Table 1: Generic Structure for Computing Disciplines

Areas	Credit Hours	Courses
Computing Core	46	14
Domain Core	18	6
Domain Elective	21	7
Mathematics & Supporting Courses	12	4
Elective Supporting Courses	3	1
General Education Requirement	30	12
Totals	**130**	**44**

STRUCTURE

Sr.	Categories	Credit Hours Min – Max
1.	General Education (Gen Edu) Requirements: Mandatory Courses of General Education.	30 – 30
2.	Major (Disciplinary) Requirements: Area of Study in Which the Degree is offered	*72 ≥
3.	Interdisciplinary/Allied Requirements (To Support Horizon of the Major)	**12 ≥
4.	Field Experience/Internship (Practical Work Experience related to a Student's Field of Study or Career interest)	03 – 03
5.	Capstone Project or Capstone Research Project	03 – 06
	Total	**120 – 144**

*****

The Credit Hours for the courses of Major Disciplines may vary but not less than 72 Credit Hours.
****The Credit Hours for Interdisciplinary/Allied Courses may vary but not less than 12 Credit Hours.**

- Total number of Credit hours 120-144
- Duration 4 years
- Semester duration 16-18 weeks
- Semesters 8
- Course Load per Semester 15-18 Cr hr
- Number of courses per semester 4-6 (not more than 3 lab / practical courses)

Scheme of studies for BS (Computer Science)

4-Year Program (8 Regular Semesters of 18 weeks each)

Semester	Category	Course Codes	Course Title	Lectures	Lab	Cr.Hrs
Semester 1	Art & Humanities	AH-301	Art & Humanities(GER)	2	0	2
	Islamic Studies/Religious Studies/Ethics	ISL-301	Islamic Studies (GER)	2	0	2
	Quantitative Reasoning (QR I)	MTH-401	Quantitative Reasoning (QR I)	3	0	3
	Functional English	ENG-303	Functional English (GER)	3	0	3
	Major I	CSC-302	Programming Fundamentals (CC)	3	1	4
	Application of Information and Communication Technologies.	CSC-308	Applications of Information and Communication Technologies (GER)	2	1	3
			Total	**15**	**2**	**17**
Semester 2	Natural Science	**0000**	Natural Science (GER)	2	1	3
	Expository Writing	ENG-304	Expository Writing (GER)	3	0	3
	Interdisciplinary/Allied	MTH-317	Linear Algebra (MSC)	3	0	3
	Ideology and Constitution of Pakistan	PST-313	Ideology and Constitution of Pakistan (GER)	2	0	2
	Major II	CSC-312	Object Oriented Programming (CC)	3	1	4
	Major III	CSC-305	Digital Logic and Design (CC)	2	1	3
			Total	**15**	**3**	**18**
Semester 3	Civics and Community Management	PSC-418	Civics and Community Management (GER)	2	0	2
	Major III	CSC-417	Human Computer Interaction and Computer Graphics (DC)	2	1	3
	Interdisciplinary/Allied	MTH-410	Multivariate Calculus (MSC	3	0	3
	Entrepreneurship	MS-309	Introduction to Entrepreneurship (GER)	2	0	2
	Major IV	CSC-423	Data Structures (CC)	3	1	4
	Major V	CSC-431	Computer Organization and Assembly Language (CC)	2	1	3
			Total	**14**	**3**	**17**
Semester 4	Social Science	000	Social Science (GER)	2	0	2
	Quantitative Reasoning (QR II)	MTH-402	Quantitative Reasoning (QR II)	3	0	3
	Major VI	CSC-428	Computer Architecture (DC)	2	1	3
	Major VII	CSC-432	Operating Systems (CC)	2	1	3
	Major VIII	CSC-426	Domain Elective-I(Web Technologies)	2	1	3
	Major IX	CSC-429	Domain Elective-II(Cyber Security)	2	1	3

			Total	**13**	**4**	**17**
Semester 5	Interdisciplinary/Allied Course	STATS-402	Probability and Statistics (MSC)	3	0	3
	Major X	CSC-502	Theory of Automata (DC)	3	0	3
	Major XI	CSC-524	Database Systems (CC)	3	1	4
	Major XII	CSC-515	Computer Networks (CC)	2	1	3
	Major XIII	CSC-533	Software Engineering (CC)	3	0	3
			Total	**14**	**2**	**16**
Semester 6	Internship (Mandatory)	CSC-698	Internship (Mandatory in summer break)	3	0	3
	Interdisciplinary/Allied Course	MS-402/MS-305	Basics of Marketing OR/Financial Accounting 1(ESC)	3	0	3
	Interdisciplinary/Allied Course	ENG-402	Business Communication (MSC)	3	0	3
	Major XIV	CSC-517	Parallel and Distributed Computing (DC)	2	1	3
	Major XV	CSC-521	Information Security (CC)	2	1	3
	Major XVI	CSC-543	Artificial Intelligence (CC)	2	1	3
			Total	**15**	**3**	**18**
Semester 7	Major XVII (Elective)		Domain Elective III			3
	Major XVIII (Elective)		Domain Elective IV			3
	Major XIX	CSC-629	Compiler Construction (DC)	2	1	3
	Major XXX	CSC-672	Analysis of Algorithm (CC)	3	0	3
	Capstone Research Project	CSC-699	Capstone Research Project (CC)	0	3	3
			Total			**15**
(TWO COURSES HAVE TO BE SELECTED FROM LIST OF COMPUTER SCIENCE DOMAIN ELECTIVE COURSES)						
Semester 8	Major XXI	CSC-628	Advance Database Management System(DC)	2	1	3
	Major XXII (Elective)		Domain Elective V			3
	Major XXIII (Elective)		Domain Elective VI			3
	Major XXIV(Elective)		Domain Elective VII			3
	Capstone Research Project	CSC-699	Capstone Research Project (CC)	0	3	3
			Total			**15**
(THREE COURSES HAVE TO BE SELECTED FROM LIST OF COMPUTER SCIENCE DOMAIN ELECTIVE COURSES)						
Total credit hours 133, Computing Core (CC)=14,Domain Core (DC)=6,Domain Electives(DE)=7, General Education (GER)=12,Maths and Supporting Courses (MSC)=4, Elective Supporting Course ESC=1						

Computer Science Domain Electives

No	**Course Title**	**Course Code**	**Credit hours**
1.	Agent Based Software Engineering	CSC-660	3 (3,0)
2.	Big Data Analytics	CSC-643	3 (2,1)
3.	Cloud Computing	CSC-537	3 (3,0)
4.	Web Technologies	CSC-426	3(2,1)
5.	Software Testing and Quality Assurance	CSC-684	3(2,1)
6.	Cyber Security	CSC-429	3(2,1)

7	Web Engineering	CSC-430	3(2,1)
8.	Computer Graphics	CSC-525	3(2,1)
9	Professional Practices	CSC-530	3(3,0)
10.	Data Encryption and Security	CSC-659	3(3,0)
11.	E-Commerce	CSC-615	3(3,0)
12.	Game Application Development	CSC-662	3(2,1)
13.	Global Software Development	CSC-661	3(3,0)
14.	Information Systems Audit	CSC-657	3(3,0)
15.	Management Information Systems	CSC-656	3(3,0)
16.	Mobile Application Development	CSC-624	3(2,1)
17.	Multimedia Communication	CSC-654	3(3,0)
18.	Real Time Systems	CSC-540	3 (3,0)
19.	Semantic Web	CSC-544	3 (3,0)
20.	Software Engineering Economics	CSC-545	3 (3,0)
21.	Software Metrics	CSC-529	3 (3,0)
22.	Systems Programming	CSC-546	3 (3,0)
23.	Topics in Software Engineering	CSC-547	3 (3,0)
24.	Visual Programming	CSC-531	3 (2,1)
25.	Advance Statistics		3(3,0)
26.	Data Mining	CSC-634	3 (2,1)
27.	Deep Learning	CSC-663	3 (2,1)
28.	Speech Processing	CSC-665	3 (3,0)
29.	Reinforcements Learning	CSC-664	3 (3,0)
30.	Fuzzy Systems	CSC-666	3 (3,0)
31.	Evolutionary Computing	CSC-667	3 (3,0)
32.	Swarm Intelligence	CSC-668	3 (3,0)
33.	Agent Based Modeling	CSC-669	3 (3,0)
34.	Knowledge Based Systems	CSC-670	3 (3,0)
35.	Programming for Artificial Intelligence	CSC-539	3 (2,1)
36.	Machine Learning	CSC-528	3 (2,1)
37.	Artificial Neural Networks	CSC-646	3 (2,1)
38.	Knowledge Representation & Reasoning	CSC-529	3 (3,0)
39.	Computing Vision	CSC-622	3 (2,1)
40.	Natural Language Processing	CSC-649	3 (2,1)
41.	Introduction to Data Science	CSC-536	3 (2,1)
42.	Data Visualization	CSC-644	3 (2,1)
43.	Data Warehousing & Business Intelligence	CSC-641	3 (2,1)
44.	Privacy Preservation	CSC-671	3 (3,0)
45.	Speech Processing	CSC-672	3 (3,0)
46.	Text Mining	CSC-673	3 (3,0)
47.	Topics in Data Science	CSC-674	3 (3,0)
48.	Advance Database Management Systems	CSC-528	3 (2,1)
49.	Introduction to Cyber Security	CSC-534	3 (2,1)
50.	Digital Forensics	CSC-651	3 (2,1)
51.	Information Assurance	CSC-535	3 (2,1)
52.	Network Security	CSC-650	3 (2,1)
53.	Secure Software Design and Development	CSC-652	3 (2,1)
54.	Vulnerability Assessment & Reverse Engineering	CSC-653	3 (2,1)

55.	Hardware Security	CSC-675	3 (3,0)
56.	Malware Analysis	CSC-676	3 (3,0)
57.	Wireless and Mobile Security	CSC-677	3 (3,0)
58.	Penetration Testing	CSC-678	3 (2,1)
59.	Cyber Law & Cyber Crime (Cyber Warfare)	CSC-679	3 (3,0)
60.	Digital Image Processing	CSC-632	3 (2,1)
61.	Simulation and Modeling	CSC-655	3 (3,0)
62.	Graph theory	CSC-658	3 (3,0)

BS Software Engineering

BS Software Engineering

Software plays a central and underpinning role in almost all aspects of daily life: communications, government, manufacturing, banking and finance, education, transportation, entertainment, medicine, agriculture, and law. The number, size, and application domains of computer programs have grown dramatically; as a result, huge sums are being spent on software development. Most people's lives and livelihoods depend on this development's effectiveness. Software products help us to be more efficient and productive. They provide information, make us more effective problem solvers, and provide us with safer, more flexible, and less confining work, entertainment, and recreation environments.

Software Engineering is the application of a systematic, disciplined and quantifiable approach to the design, development, operation, and maintenance of software systems. It is in fact the practice of designing and implementing large, reliable, efficient and economical software by applying the principles and practices of engineering. The department aims to train students in all aspects of software life cycle from specification through analysis and design to testing, maintenance and evolution of software product.

Program Learning Outcomes (PLOs)

Program learning outcomes are the narrower statements that describe what students are expected to know and be able to do by the time of graduation. These relate to the knowledge, skills and attitude that the students acquire while progressing through the program.

The program must demonstrate that by the time of graduation the students have attained a certain set of knowledge, skills and behavioral traits, at least to some acceptable minimum level. Specifically, it is to be demonstrated that the students have acquired the following graduate attributes (GAs)

GA1 Engineering Knowledge: An ability to apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

GA2 Problem Analysis: An ability to identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusionsusing first principles of mathematics, natural sciences and engineering sciences.

GA3 Design/Development of Solutions: An ability to design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

GA4 Investigation: An ability to investigate complex engineering problems in a methodical way including literature survey, design and conduct of experiments, analysis and interpretation of experimental data, and synthesis of information to derive valid conclusions.

GA5 Modern Tool Usage: An ability to create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering activities, with an understanding of the limitations.

GA6 The Engineer and Society: An ability to apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issuesand the consequent responsibilities relevant to professional engineering practice and solution to complex engineering problems.

GA7 Environment and Sustainability: An ability to understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of and need for sustainable development.

GA8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

GA9 Individual and Team Work: An ability to work effectively, as an individual or in a team, on multifaceted and /or multidisciplinary settings.

GA10 Communication: An ability to communicate effectively, orally as well as in writing, on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

GA11 Project Management: An ability to demonstrate management skills and apply engineering principles to one's own work, as a member and/or leader in a team, to manage projects in a multidisciplinary environment.

GA12 Lifelong Learning: An ability to recognize importance of, and pursue lifelong learning in the broader context of innovation and technological developments.

Proposed Curriculum for BS-SE

Curriculum Model for BS Software Engineering

The generic structure for computing degree program given before is mapped with the BSSEprogram in the following tables.

Generic Structure for Computing Disciplines:

Areas	Credit Hours	Courses
Computing Core	46	14
Domain Core	18	6
Domain Elective	21	7
Mathematics & Supporting Courses	12	4
Elective Supporting Courses	3	1
General Education Requirement	30	12
Totals	**130**	**44**

SCHEME OF STUDIES OF BS SOFTWARE ENGINEERING (SESSION- 2023 ONWARDS)

Semester	Category	Course Codes	Course Title	Lectures	Lab	Cr.Hrs
Semester 1	Art & Humanities	AH-301	Art & Humanities(GER)	2	0	2
	Islamic Studies/Religious Studies/Ethics	ISL-301	Islamic Studies (GER)	2	0	2
	Quantitative Reasoning (QR I)	MTH-401	Quantitative Reasoning (QR I)	3	0	3
	Functional English	ENG-303	Functional English (GER)	3	0	3
	Major I	CSC-302	Programming Fundamentals (CC)	3	1	4
	Application of Information and Communication Technologies.	CSC-308	Applications of Information and Communication Technologies (GER)	2	1	3
			Total	**15**	**2**	**17**
Semester 2	Natural Science	**0000**	Natural Science (Applied Physics) (GER)	2	1	3
	Expository Writing	ENG-304	Expository Writing (GER)	3	0	3
	Interdisciplinary/Allied	MTH-317	Linear Algebra (MSC)	3	0	3
	Ideology and Constitution of Pakistan	PST-313	Ideology and Constitution of Pakistan (GER)	2	0	2
	Major II	CSC-312	Object Oriented Programming (CC)	3	1	4
	Major III	CSC-305	Digital Logic and Design (CC)	2	1	3
			Total	**15**	**3**	**18**

Semester 3	Civics and Community Management	PSC-418	Civics and Community Management (GER)	2	0	2
	Major III	CSC-533	Software Engineering (CC)	3	0	3
	Interdisciplinary/Allied	MTH-410	Multivariate Calculus (MSC	3	0	3
	Entrepreneurship	MS-309	Introduction to Entrepreneurship (GER)	2	0	2
	Major IV	CSC-423	Data Structures (CC)	3	1	4
	Major V	CSC431	Computer Organization and Assembly Language (CC)	2	1	3
			Total	**15**	**2**	**17**
Semester 4	Social Science	000	Social Science (GER)	2	0	2
	Quantitative Reasoning (QR II)	MTH-402	Quantitative Reasoning (QR II)	3	0	3
	Major VI	CSC-504	Software Design and Architecture (DC)	3	0	3
	Major VII	CSC-432	Operating Systems (CC)	2	1	3
	Major VIII	CSC-426	Domain Elective-I(Web Technologies)	2	1	3
	Major IX	CSC-429	Domain Elective-II(Cyber Security)	2	1	3
			Total	**14**	**3**	**17**
Semester 5	Interdisciplinary/Allied Course	STATS-402	Probability and Statistics (MSC)	3	0	3
	Major X	CSC-631	Software Construction and Development (DC)	2	1	3
	Major XI	CSC-524	Database Systems (CC)	3	1	4
	Major XII	CSC-515	Computer Networks (CC)	2	1	3
	Major XIII	CSC-543	Artificial Intelligence (CC)	2	1	3
			Total	**12**	**4**	**16**
Semester 6	Internship (Mandatory)	CSC-698	Internship (Mandatory in summer break)	3	0	3
	Interdisciplinary/Allied Course	MS-402/MS-305	Basics of Marketing /OR/Financial Accounting I(ESC)	3	0	3
	Interdisciplinary/Allied Course	ENG-402	Business Communication (MSC)	3	0	3
	Major XIV	CSC-516	Software Quality Engineering (DC)	2	1	3
	Major XV	CSC-521	Information Security (CC)	2	1	3
	Major XVI	CSC-580	Software Requirement Engineering (DC)	2	1	3
			Total	**15**	**3**	**18**
Semester 7	Major XVII (Elective)		Domain Elective III			3
	Major XVIII (Elective)		Domain Elective IV			3
	Major XIX	CSC-538	Software Project Management (DC)	2	1	3
	Major XXX	CSC-672	Analysis of Algorithm (CC)	3	0	3
	Capstone Research Project	CSC-699	Capstone Research Project (CC)	0	3	3
			Total			**15**

(TWO COURSES HAVE TO BE SELECTED FROM LIST OF COMPUTER SCIENCE DOMAIN ELECTIVE COURSES)						
Semester 8	Major XXI	CSC-517	Parallel and Distributed Computing (DC)	2	1	3
	Major XXII (Elective)		Domain Elective V			3
	Major XXIII (Elective)		Domain Elective VI			3
	Major XXIV(Elective)		Domain Elective VII			3
	Capstone Research Project	CSC-699	Capstone Research Project (CC)	0	3	3
			Total			**15**
(THREE COURSES HAVE TO BE SELECTED FROM LIST OF COMPUTER SCIENCE DOMAIN ELECTIVE COURSES)						
Total credit hours 133, Computing Core (CC)=14,Domain Core (DC)=6,Domain Electives(DE)=7, General Education (GER)=12,Maths and Supporting Courses (MSC)=4, Elective Supporting Course ESC=1						
GER, QR1 and QR2 Courses will be taken from booklet of General and Mandatory Courses Booklet						

SE Domain Electives (21/130) 7 course			
S.No	**Course Title**	**Course Code**	**Credit hrs**
1	Software Verification and Validation (Testing & QA)	CSC-460	3(2,1)
2	Object Oriented Analysis & Design	CSC-440	3(2,1)
3	Computer Architecture	CSC-428	3(2,1)
4	Theory of Automata	CSC-502	3(3,0)
5	HCI & Computer Graphics	CSC-417	3(2,1)
6	Web Technologies	CSC-426	3(2,1)
7	Advanced Database Management	CSC-628	3(2,1)
8	Web Engineering	CSC-430	3(2,1)
9	Introduction to Data Science	CSC-536	3(2,1)
10	Software Re-Engineering	CSC-620	3(2,1)
11	Mobile Application Development	CSC-624	3(2,1)

BS Course Outlines

DETAIL OF COURSES

1 Course Contents

There are about 29 courses which are common to all degree offeredunder the computing discipline. These common courses are divided into three different categories which are given in the tables below. The course contents of these courses are given on next few pages.The Bloom's Taxonomy is also identified as C = Cognitive domain, P = Psychomotor domain, A = Affective domain with each course. CLOs are also defined with each course.

Table List of Computing Core Courses (Both for BS(CS) and BSSE)

#	Course Title	Domain	Cr Hr
1	Programming Fundamentals	Computing Core	4 (3-3)
2	Object Oriented Programming	Computing Core	4 (3-3)
3	Database Systems	Computing Core	4 (3-3)
4	Digital Logic Design	Computing Core	3 (2-3)
5	Data Structures	Computing Core	4 (3-3)
6	Information Security	Computing Core	3 (2-3)
7	Artificial Intelligence	Computing Core	3 (2-3)
8	Computer Networks	Computing Core	3 (2-3)
9	Computer Organization & Assembly Language	Computing Core	3 (2-3)
10	Software Engineering	Computing Core	3 (3-0)
11	Operating Systems	Computing Core	3 (2-3)
12	Analysis of Algorithms	Computing Core	3 (3-0)
13	Final Year Project – I*	Computing Core	3 (0-6)
14	Final Year Project – II*	Computing Core	3(0-12)

***course contents are not applicable.**

Mathematics & Supporting Courses (Both for BS(CS) and BSSE

#	Course	Domain	Cr Hr
1	Multivariate Calculus	Maths	3 (3-0)
2	Probability & Statistics	Maths	3 (3-0)
3	Linear Algebra	Maths	3 (3-0)
4	Technical & Business Writing/Business Communication	EW	3 (3-0)

General Education Requirement as per HEC UG Education Policy (Both for BS(CS) and BSSE

#	Course	Domain	Cr Hr
1	Application of Information & Communication Technologies	GER	3 (2-3)
2	Functional English	GER	3 (3-0)
3	Expository Writing	GER	3 (3-0)
4	Quantitative Reasoning – 1 (Discrete Structures)	GER	3 (3-0)
5	Quantitative Reasoning – 2 (Calculus and Analytic Geometry)	GER	3 (3-0)
6	Islamic Studies	GER	2 (2-0)
7	Ideology and Constitution of Pakistan	GER	2 (2-0)
8	Social Sciences (Example: Introduction to Management	GER	2 (2-0)
9	Natural Sciences (Applied Physics)	GER	3 (2-3)
10	Arts & Humanities (Professional Practices)	GER	2 (2-0)
11	Civics and Community Engagement	GER	2 (2-0)
12	Entrepreneurship	GER	2 (2-0)

#University shall prepare course contents for these courses or the university may offer some other course instead of these courses from the same area.

Domain Core (BSCS) (18/130) 6 Courses				
1		Theory of Automata	Domain Core	3 (3-0)
2	DB	Advance Database Management Systems	Domain Core	3 (2-3)
3		HCI & Computer Graphics	Domain Core	3 (2-3)
4	COAL	Computer Architecture	Domain Core	3 (2-3)
5	TA	Compiler Construction	Domain Core	3 (2-3)
6	OS	Parallel & Distributed Computing	Domain Core	3 (2-3)

Domain Core (BS (s/w Engineering)) (18/130) 6 Courses				
1		Software Design and Architecture	Domain Core	3 (3-0)
2	DB	Software Construction and Development	Domain Core	3 (2-3)
3		Software Project Management	Domain Core	3 (2-3)
4	COAL	Software Quality Engineering	Domain Core	3 (2-3)
5	TA	Software Requirement Engineering	Domain Core	3 (2-3)
6	OS	Parallel & Distributed Computing	Domain Core	3 (2-3)

Domain Electives BSCS (21/130) 7 courses

No	Course Title	Course Code	Credit hours
1.	Agent Based Software Engineering	CSC660	3 (3,0)
2.	Big Data Analytics	CSC643	3 (2,3)
3.	Cloud Computing	CSC537	3 (3,0)
4.	Web Technologies	CSC426	3(2,1)
5.	Software Testing and Quality Assurance	CSC684	3(2,1)
6.	Cyber Security	CSC429	3(2,1)
7	Web Engineering	CSC430	3(2,1)
8.	Computer Graphics	CSC525	3(2,1)
9.	Professional Practices	CSC530	3(3,0)
10.	Data Encryption and Security	CSC659	3(3,0)
11.	E-Commerce	CSC615	3(3,0)
12.	Game Application Development	CSC662	3(2,1)
13.	Global Software Development	CSC661	3(3,0)
14.	Information Systems Audit	CSC657	3(3,0)
15.	Management Information Systems	CSC656	3(3,0)
16.	Mobile Application Development	CSC624	3(2,1)
17.	Multimedia Communication	CSC654	3(3,0)
18.	Real Time Systems	CSC540	3 (3,0)
19.	Semantic Web	CSC544	3 (3,0)
20.	Software Engineering Economics	CSC545	3 (3,0)
21.	Software Metrics	CSC529	3 (3,0)
22.	Systems Programming	CSC546	3 (3,0)
23.	Topics in Software Engineering	CSC547	3 (3,0)

24.	Visual Programming	CSC531	3 (2,1)
25.	Advance Statistics		**3**,
26.	Data Mining	CSC634	3 (2-3)
27.	Deep Learning	CSC663	3 (2-3)
28.	Speech Processing	CSC665	3 (3-0)
29.	Reinforcements Learning	CSC664	3 (3-0)
30.	Fuzzy Systems	CSC666	3 (3-0)
31.	Evolutionary Computing	CSC667	3 (3-0)
32.	Swarm Intelligence	CSC668	3 (3-0)
33.	Agent Based Modeling	CSC669	3 (3-0)
34.	Knowledge Based Systems	CSC670	3 (3-0)
35.	Programming for Artificial Intelligence	CSC539	3 (2-3)
36.	Machine Learning	CSC528	3 (2-3)
37.	Artificial Neural Networks	CSC646	3 (2-3)
38.	Knowledge Representation & Reasoning	CSC529	3 (3-0)
39.	Computing Vision	CSC531	3 (2,1)
40.	Natural Language Processing	CSC649	3 (2,1)
41.	Introduction to Data Science	CSC536	3 (2,1)
42.	Data Visualization	CSC644	3 (2,1)

43.	Data Warehousing & Business Intelligence	CSC641	3 (2,1)
44.	Privacy Preservation	CSC671	3 (3,0)
45.	Speech Processing	CSC672	3 (3,0)
46.	Text Mining	CSC673	3 (3,0)
47.	Topics in Data Science	CSC674	3 (3,0)
48.	Advance Database Management Systems	CSC528	3 (2,1)
49.	Introduction to Cyber Security	CSC534	3 (2,1)
50.	Digital Forensics	CSC651	3 (2,1)
51.	Information Assurance	CSC535	3 (2,1)
52.	Network Security	CSC650	3 (2,1)
53.	Secure Software Design and Development	CSC652	3 (2,1)
54.	Vulnerability Assessment & Reverse Engineering	CSC653	3 (2,1)
55.	Hardware Security	CSC675	3 (3,0)
56.	Malware Analysis	CSC676	3 (3,0)
57.	Wireless and Mobile Security	CSC677	3 (3,0)
58.	Penetration Testing	CSC678	3 (2,1)
59.	Cyber Law & Cyber Crime (Cyber Warfare)	CSC679	3 (3,0)
60.	Digital Image Processing	CSC632	3 (2,1)
61.	Simulation and Modeling	CSC655	3 (0)
62.	Graph theory	CSC658	3 (3,0)

SE Domain Electives (21/130) 7 course			
No	Course Title	**Course Code**	**Credit hrs**
1	Software Verification and Validation (Testing & QA)	**CSC460**	**3(2,1)**
2	Object Oriented Analysis & Design	**CSC440**	**3(2,1)**
3	Computer Architecture	**CSC428**	**3(2,1)**
4	Theory of Automata	**CSC502**	**3(3,0)**
5	HCI & Computer Graphics	**CSC417**	**3(2,3)**
6	Web Technologies	**CSC426**	**3(2,3)**
7	Advanced Database Management	**CSC628**	**3(2,1)**
8	Web Engineering	**CSC430**	**3(2,3,**
9	Introduction to Data Science	**CS536**	**3(2,3)**
10	Software Re-Engineering	**CSC620**	**3(2,3)**
11	Mobile Application Development	**CSC624**	**3(2,3)**

Course Outlines Courses of BSCS

Note: Course Contents of General Courses (GER) , QR1 and QR2 will be taken from the General and Mandatory Courses Booklet.

Semester 1st

Course Title: Applications of Information and Communication Technologies	**Course Code: CSC-308**
Course Structure: Lectures: 2 Lab:1	**Credit Hours: 3**
Prerequisites: None	
Course Objective: • This course is designed (o provide students with an exploration of the practical applications of Information and Communication Technologies (ICT) and software tools in various domains. • Students will gain hands-on experience with a range of software applications, learning how to leverage ICT to solve daily life problems, enhance productivity and innovate in different fields. • Through individual and interactive exercises and discussions, students will develop proficiency in utilizing software for communication, creativity, and more.	

Course Outline:

Introduction to Information and Communication Technologies: Components of Information and Communication Technologies (basics of hardware, software, ICT platforms, networks, local and cloud data storage etc.). Scope of Information and Communication Technologies (use of ICT in education, business, governance, healthcare, digital media and entertainment, etc.). Emerging technologies and future trends. **Basic ICT Productivity Tools:** Effective use of popular search engines (e.g., Google, Bing, etc.) to explore World Wide Web. Formal communication tools and etiquette (Gmail, Microsoft Outlook, etc.). Microsoft Office Suites (Word, Excel, PowerPoint).Google Workspace (Google Docs, Sheets, Slides).Dropbox (Cloud storage and file sharing), Google Drive (Cloud storage with Google Docs integration) and Microsoft OneDrive (Cloud storage with Microsoft Office integration). Evernote (Note-taking and organization applications) and OneNote (Microsoft's digital notebook for capturing and organizing ideas). Video conferencing (Google Meet, Microsoft Teams, Zoom, etc.). Social Media applications (Linkedln, Facebook, Instagram, etc.). **ICT in Education:** Working with learning management systems (Moodle, Canvas, Google Classrooms, etc.).Sources of online education courses (Coursera, edX, Udemy, Khan Academy, etc.).Interactive multimedia and virtual classrooms. **ICT in Health and Well-being:** Health and fitness tracking devices and applications (Google Fit, Samsung Health, Apple Health, Xiaomi Mi Band, Runkccpcr, etc.).Telemedicine and online health consultations (OLADOC, Sehat Kahani, Marham, etc.).**ICT in Personal Finance and Shopping:** Online banking and financial management tools (JazzCash, Easypaisa, Zong PayMax, Il LINK and MNET, Keenu Wallet, etc.).E-commerce platforms (Daraz.pk, Telcnoart, Shophivc, etc.)Digital Citizenship and Online Etiquette. Digital identity and online reputation. Netiquette and respectful online communication. Cyberbullying and online harassment. **Ethical Considerations in Use of ICT Platforms and Tools:** Intellectual property and copyright issues. Ensuring originality in content creation by avoiding plagiarism and unauthorized use of in fornication sources. Content accuracy and integrity (ensuring that the content shared through ICT platforms is free from misinformation, fake news, and manipulation).

Practical Requirements

As part of the overall learning requirements, the course will include guided tutorials and exercises to ensure that students are proficient in commonly used software applications such as worcl processing software (e.g., Microsoft Word), presentation software (e.g., Microsoft PowerPoint), spreadsheet software (e.g., Microsoft Excel) among such other tools. Students may be assigned practical tasks that require them to create documents, presentations, and spreadsheets etc. Assigning tasks that involve creating, managing, and organizing files and folders on both local and cloud storage systems. Students will practice file naming conventions, creating directories, and using cloud storage solutions (e.g., Google Drive, OneDrive). The use of online learning management systems (LMS) where students can access course materials, submit assignments, participate in discussion forums, and take quizzes or tests. This will provide students with the practical experience with online platforms commonly used in education and the workplace.

Course Outcomes:

- By the end of this course, students will be able to Explain the fundamental concepts, components, and scope of Information and Communication Technologies (ICT).
- Identify uses of various ICT platforms and tools for different purposes.
- Apply ICT platforms and tools for different purposes to address basic needs in different domains of daily, academic, and professional life.
- Understand the ethical and legal considerations in use of ICT platforms and tools.

Recommended Books: (Min5-8Max) Latest Edition of the Following Books.

1.Vermaat, Shaffer, and Freund, Discovering Computers, 2017, Cengage Learning.
2.Gaskin, Vargas, and McLellan, GO! with Microsoft Office,2013, Pearson.
3.Grauer and Poatsy. Exploring Microsoft Office,2016, Pearson.
4.Morley and Parker Computing Essentials, 2023, McGraw Hill.
5. Evans, Martin, and Poatsy. Technology in Action,2021, Pearson.

Course Name: Programming Fundamentals	**Course Code: CSC-302**
Course Structure: Lectures: 3, Labs: 1	**Credit Hours: 4**
Prerequisites: None	
Course Objective: • This course provides fundamental concepts of programming to freshmen. The courses is pre-requisite to many other courses, therefore, students are strongly advised to cover all contentsand try to achieve CLOs to the maximum possible level. • The course may be taught as language independent. Further, it is up to the university to choose any language for the practical/Lab purpose but that must be latest and market oriented.	
Course Outline: Introduction to problem solving, a brief review of Von-Neumann architecture, Introductionto programming, role of compiler and linker, introduction to algorithms, basic data types andvariables, input/output constructs, arithmetic, comparison and logical operators, conditional statements and execution flow for conditional statements, repetitive statements and execution flow for repetitive statements, lists and their memory organization, multi- dimensional lists, introduction to modular programming, function definition and calling, stack rolling and unrolling, string and string operations, pointers/references, static and dynamic memory allocation, File I/O operations.	
Course Outcomes: After completion of the course students will be able to: • Understand basic problem-solving steps and logicconstructs • Apply basic programing concepts • Design and implement algorithms to solve real worldproblems	
Recommended Books: Latest Edition of the Following Books 1. Tony Gaddis,, Starting out with Programming Logic & Degins, 4th Edition, 2018,Peearson. 2. Brian W. Kernighan, Dennis M. Ritchie, The C Programming Language, 2nd Edition, 1988,Pearson 3. Robert Lafore, Object Oriented Programming in C++,2001,Sams. 4. Paul Deitel & Harvey Deitel , C How to Program, 7th Edition, 2015,Pearson 5. Jeri R. Hanly & Elliot B.Koffman, Problem Solving and Program Design in C++, 7th Edition,2006,Edison Wesley	

Course Outlines of Courses of Semester 2nd

Course Title: Linear Algebra	**Course Code: MTH-405**
Course Structure: Lectures: 3	**Credit Hours:** 3
Prerequisites: Calculus	
Course Objective: • To provide fundamentals of solutions for systems of linear equations, operations on a system of equations, matrix properties, solutions, and study of their properties, to enable the students about Practical applications in Bio-Informatics, the aim is to provide a practical description of the topics, tools, issues and current trends in the fields including their impact on biology and human health and medicine. • Make students polish their analytical skills. Have well understanding to utilize this course in this program.	
Course Outline: Systems of Linear Equations and Matrices; Introduction to systems of linear equations, Gaussian Elimination, Matrices and Matrices Operations, Inverses, Rules of Matrix Arithmetic, Elementary Matrices and a method for finding inverse, systems of equations and invertibility, diagonal triangular and symmetric matrices, Determinants; the determinant , evaluating determinants by row reduction, properties of the determinant function, co-factor, crammer's rule, Vector in 2space and 3 space, introduction of vector, norm, vector athematic , dot product cross product, General vector spaces; real vector spaces, subspace, Linear Independence, basis and	

dimension, row space ,column space and null space, rank and nullity , Eigen values and Eigen vector, diagonalization, Orthogonal diagonalization, Linear transformation; general linear transformation, kernel and range, matrices of general linear transformation pace.

Intended Learning Outcomes: Students will understand: some applications of the system of linear equations, and apply matrix multiplications in digraphs and communication matrices.

Recommended Books: Latest Edition of the Following Books

1. Anton, H., (2018). Elementary Linear Algebra, Eighth Edition, United Kingdom: Wiley.
2. Anton, H., Rorres, C., Kaul, A. (2019). Elementary Linear Algebra. United Kingdom: Wiley.
3. Messer, R. (2021). Linear Algebra: Gateway to Mathematics. United States: American Mathematical Society.
4. Boyd, S., Vandenberghe, L. (2018). Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares. United Kingdom: Cambridge University Press.

Course Name: Object Oriented Programming	**Course Code: CSC-312**
Course Structure: Lectures: 3, Labs: 1	**Credit Hours:4**

Prerequisites: Programming Fundamentals

Course Objective:

- The course aims to focus on object-oriented concepts, analysis and software development.
- The basic concept of OOP is covered in this course.

Course Outline:

Introduction to object oriented design, history and advantages of object oriented design, introduction to object oriented programming concepts, classes, objects, data encapsulation, constructors, destructors, access modifiers, const vs non-const functions, static data members& functions, function overloading, operator overloading, identification of classes and their relationships, composition, aggregation, inheritance, multiple inheritance, polymorphism, abstract classes and interfaces, generic programming concepts, function & class templates, standard template library, object streams, data and object serialization using object streams, exception handling.

Course Outcomes: After completion of the course students will be able to:

- Understand principles of object-oriented paradigm.
- Identify the objects & their relationships to build objectoriented solution
- Model a solution for a given problem using objectoriented principles
- Examine an object oriented solution

Recommended Books: Latest Edition of the Following Books

1. Paul Deitel, Java: How to Program, 9th Edition, 2011,Pearson College Div.
2. Ivor Horton, Beginning Java 2, 7th Edition, 1999, Apress
3. C. ThomasWu , An Introduction to Object Oriented Programming with Java, 5th Edition,2009,McGeraw Hills
4. Tony Gaddis, Starting Out with C++ from Control Structures to Objects, 9th Edition, 2017,Pearson.
5. Deitel & Deitel, C++ How to Program, 10th Edition, 2005, Prentice Hall
6. Robert Lafore, Object Oriented Programming in C++, 3rd Edition,2008,Sams.

Course Name: Digital Logic and Design	**Course Code: CSC-305**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: None	

Course Objective:

- The course introduces the concept of digital logic, gates and the digital circuits.
- Itfocuses on the design and analysis combinational and sequential circuits.
- It also serves to familiarize the student with the logic design of basic computer hardware components.

Course Outline:

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

Course Outcomes:

- Acquire knowledge related to the concepts, tools and techniques for the design of digital electronic circuits .
- Demonstrate the skills to design and analyze both combinational and sequential circuits using a variety of techniques.
- Apply the acquired knowledge to simulate and implement small-scale digital circuits.
- Understand the relationship between abstract logic characterizations and practical electrical implementations.

Recommended Books: Latest Edition of the Following Books.

1. Floyd, Digital Fundamentals.2014, Pearson.
2. Verilog Design, Stephen Brown, Fundamental of Digital Logic, 2013, McGraw Hill

Course Outlines of Courses of Semester 3rd

Course Name: Multivariate Calculus	**Course Code: MTH-404**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**
Prerequisites: Calculus and Analytical Geometry	

Course Objectives:

The goals are to develop the skills to have ground knowledge of multivariate calculus and appreciation for their further computer science courses.

Intended Learning Outcomes:

Students will be able: to apply mathematical and computational methods to range of application problems in multivariate calculus, evaluate partial derivations and multiple integrals of multivariate functions.

Course Outline:

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

Reference Material:

1. Stewart,J. (6th ed.).(2007).Multivariable Calculus. Cengage Learning publishers.
2. Swokowski,Olinick,M.,Pence,D.(6th ed.). (1994). Calculus and Analytical Geometry. Thomson Learning EMEA, Ltd.
3. Anton,H ,Herr,A.(5th Ed.).(1995). Multivariable Calculus. John Wiley.

Course Name: Data Structures	**Course Code: CSC-423**
Course Structure: Lectures: 3, Labs: 1	**Credit Hours: 4**
Prerequisites: Object Oriented Programming	

Course Objective:

- The course is designed to teach students structures and schemes, which allow them to writeprogrammer to efficiently manipulate, store, and retrieve data.
- Students are exposed to the concepts of time and space complexity of computer programs.

Course Outline

Abstract data types, complexity analysis, Big Oh notation, Stacks (linked lists and array implementations), Recursion and analyzing recursive algorithms, divide and conquer algorithms, Sorting algorithms (selection, insertion, merge, quick, bubble, heap, shell, radix,bucket), queue, dequeuer, priority queues (linked and array implementations of queues), linked list & its various types, sorted linked list, searching an unsorted array, binary search for sorted arrays, hashing and indexing, open addressing and chaining, trees and treetraversals, binary search trees, heaps, M-way tress, balanced trees, graphs, breadth-first anddepth-first traversal, topological order, shortest path, adjacency matrix and adjacency list implementations, memory management and garbage collection.

Course Outcomes: After completion of the course students will be able to:

- Implement various data structures and their algorithmsand apply them in implementing simple applications.
  - Analyze simple algorithms and determine theircomplexities.
  - Apply the knowledge of data structure to other applicationdomains.
  - Design new data structures and algorithms to solveproblems.

Recommended Books: Latest Edition of the Following Books

1. Mark A. Weiss , Data Structures and Algorithm Analysis in Java,2011,Pearson.
2. Frank M. Carrano & Timothy M. Henry ,Data Structures and Abstractions with Java,2014,Pearson.
3. Adam Drozdek, Data Structures and Algorithms in C++ ,2012,Cengage Learning.
4. Mark Allen Weiss, Data Structures and Algorithm Analysis in C++ ,2013,Pearson.
5. John Lewis & Joseph Chase Java Software Structures: Designing and Using Data Structures, 2013, Pearson.

Course Name: Computer Organization and Assembly Language	**Course Code: CSC-431**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Digital Logic Design	

Course Objective:

- The main objective 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.

Course Outline:

Introduction to computer systems: Information is bits + context, programs are translated byother programs into different forms, it pays to understand how compilation systems work, processors read and interpret instructions stored in memory, caches matter, storage devices form a hierarchy, the operating system manages the hardware, systems communicate with other systems using networks; Representing and manipulating information: information storage, integer representations, integer arithmetic, floating point; Machine-level representation of programs: a historical perspective, program encodings, data formats,accessing information, arithmetic and logical operations, control, procedures, array allocation and access, heterogeneous data structures, putting it together: understanding pointers, life in the real world: using the gdb debugger, out of-bounds memory references and buffer overflow, x86-64: extending ia32 to 64 bits, machine-level representations of floating-point programs; Processor architecture: the Y86 instruction set architecture, logic design and the Hardware Control Language (HCL), sequential Y86 implementations, generalprinciples of pipelining, pipelined Y86 implementations

Course Outcomes: After completion of the course students will be able to:

- Acquire the basic knowledge of computer organizationcomputer architecture and assembly language
- Understand the concepts of basic computer organization,architecture, and assembly language techniques
- Solve the problems related to computer organization andassembly language

Recommended Books: Latest Edition of the Following Books

1. M. Morris Mano,, Computer System Architecture, Latest Edition,1992,Pearson.
2. Kip R Irvin, Assembly Language Programming for Intel- Computer, Latest Edition,2008,Pearson.
3. Randal E. Bryant andDavid R.O' Hallaron, Computer Systems: A Programmer's Perspective, 3/E (CS:APP3e), Carnegie Mellon University,2015,Pearson.
4. Robert Britton, MIPS Assembly Language Programming, Latest Edition,2003,Pearson.

Course Name: Human Computer Interaction and Computer Graphics	**Course Code: CSC-423**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- This course introduces the human issues of usability and its importance.
- It considers the implications of human understanding on the usability of computer systems and the importance of understanding the context of use.
- It describes guidelines for use of different media and interface styles. In addition, the basics of computer graphics is also covered.

Course Outline:

Introduction to Human Computer Interaction, The Human, input-output channels, Human memory, reasoning, and problem solving, Emotion, Individual differences, psychology and the design of interactive systems. The computer, text entry devices, positioning, pointing and drawing, Display Devices, Devices for virtual and 3D interaction, Physical controls sensors and special devices, printing and scanning, Memory, processing and networks. The Interaction, Models of Interaction, Frameworks and HCI, Ergonomics, Interaction Styles, Elements of the WIMP Interface, Interactivity, The context of the interaction in Software Process, Software life cycle, usability engineering, Interactive design and prototyping, design rationale. Fundamental Concepts: forward and backward rendering (i.e., ray-casting andrasterization), applications of computer graphics: including game engines, cad, visualization, virtual reality, polygonal representation, basic radiometry, similar triangles, and projection model, use of standard graphics APIs (see HCI GUI construction); geometric modeling: basic geometric operations such as intersection calculation, proximity tests, polynomial curves and surfaces. Approximation techniques such as polynomial curves, bezier curves, spline curves and surfaces, animation as a sequence of still images.

Course Outcomes: After completion of the course students will be able to:

- understand different media and interface styles and able to develop interactive systems.
- Compare key algorithms for modelling and rendering graphical data.
- Develop design and problem-solving skills with applications to computer graphics.

Recommended Books: Latest Edition of the Following Books

1. Dix ,A., Finlay,J.E. , Abowd,G.D, Beale,R**.** Human-Computer Interaction. 2004,Prentice Hall.
2. Shneiderman, B, Plaisant,C. Designing the User Interface: Strategies for Effective Human-Computer Interaction.2016, Addison-Wesley.
3. Donald D. Hearn, Computer Graphics with Open GL Prentice Hall,2010, ISBN-10: 0136053580.
4. S. J. Gortler, Foundations of 3D Computer Graphics,2012, The MIT.
5. A K Peters, Fundamentals of Computer Graphics, 2009, A K Peters/CRC Press.
6. Addison Wesley, Computer Graphics: Principles and Practice, 2013, Addison-Wesley Professional.
7. A K Peters, Real-Time Rendering, 2008, A K Peters/CRC Press.

Course Outlines Courses of Semester 4th

Course Name: Computer Architecture	**Course Code: CSC-428**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Computer Organization and Assembly Language

Course Objective:

- Get a deeper understanding of how computers work, working knowledge of various subsystems and the general principles that affect their performance.
- Analyze the performance of systems and quantify the performance measurements, fundamentals of all technologies, and advanced architectural features that boost the performance of computers.

Course Outline:

Introduction and Performance, Technology trends, Measuring CPU performance, Amdahl's law and averaging performance metrics, Instruction Sets. Components of an instruction set, Understanding instruction sets from an implementation perspective, RISC and CISC and example instruction sets.

Computer Arithmetic, Ripple carry, carry lookahead, and other adder designs, ALU and Shifters,Floating-point arithmetic and floating-point hardware design. Datapath and Control, Single-cycle and multi-cycle datapaths, Control of datapaths and implementing control finite-state machines.Pipelining,Basic pipelined datapath and control, Data dependences, data hazards, bypassing, code scheduling,Branch hazards, delayed branches, branch prediction/Memory Hierarchies, Caches (direct mapped, fully associative, set associative),Main memories, Memory hierarchy performance metrics and their use,Virtual memory, address translation, TLBs. Input and Output, Common I/O device types and characteristics, Memory mapped I/O, DMA, program-controlled I/O, polling, interrupts, Networks. Multiprocessors, Introduction to multiprocessors, Cache coherence problem. Exception handling, Parallelism, multiprogramming, design of computer systems and components.

Course Outcomes:

- Students would have in-depth understanding of internal structure and working of computer system**.**

Recommended Books: Latest Edition of the Following Books.

1. Patterson, H., Kauffman, M., Computer Architecture: A Quantitative Approach, 2012, Morgan Kaufmann.
2. Patterson, H., Kauffman, M., Computer Organization & Design: The Hardware/Software Interface,2013, Mogan Kaufmann.

Course Name: Operating Systems	**Course Code: CSC-432**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Data Structures

Course Objective:

- To help students gain a general understanding of the principles and concepts governing the functions of operating systems.
- To acquaint students with the layered approach thatmakes design, implementation and operation of the complex OS possible.

Course Outlines:

Operating systems basics, system calls, process concept and scheduling, inter-processcommunication multithreaded programming, multithreading models, threading issues, process scheduling algorithms, thread scheduling, multiple-processor scheduling, synchronization, critical section, synchronization hardware, synchronization problems, deadlocks, detecting and recovering from deadlocks, memory management, swapping, contiguous memory allocation, segmentation & paging, virtual memory management, demand paging, thrashing, memory-mapped files, file systems, file concept, directory and disk structure, directory implementation, free space management, disk structure and scheduling, swap space management, system protection, virtual machines, operating systemsecurity

Course Outcomes: After completion of the course students will be able to:

- Understand the characteristics of different structures of the Operating Systems and identify the core functions ofthe Operating Systems
- Analyze and evaluate the algorithms of the core functionsof the Operating Systems and explain the major performance issues with regard to the core functions
- Demonstrate the knowledge in applying system softwareand tools available in modern operating systems.

Recommended Books: Latest Edition of the Following Books

1. Abraham Silberschatz , Operating Systems Concepts, 9th edition,2008,Willey
2. Andrew S. Tanenbaum,, Modern Operating Systems, 4th edition, 2014, Pearson.
3. William Stallings, Operating Systems, Internals and Design Principles, 9th edition,2014,Pearson.

Course Name: Web Technologies	**Course Code: CSC-426**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- The course will cover basics of Web Technolgies..

Course Outline

Introduction to Web Applications, TCP/IP Application Services. Web Servers: Basic Operation, Virtual hosting, Chunked transfers, Caching support, Extensibility. SGML, HTML5, CSS3. XML Languages and Applications: Core XML, XHTML, XHTM MP. Web Service: SOAP, REST, WML, XSL. Web Services: Operations, Processing HTTP Requests, Processing HTTP Responses, Cookie Coordination, Privacy and P3P, Complex HTTP Interactions, Dynamic Content Delivery. Server Configuration. Server Security. Web Browsers Architecture and Processes. Active Browser Pages: JavaScript, DHTML, AJAX. JSON, Approaches to Web Application Development. Programing in anyScripting language. Search Technologies. Search Engine Optimization. XML Query Language, Semantic Web, Future Web Application Framework.

Course Outcomes: After completion of the course students will be able to:

- Understand Web application development tools.
- Develop web applications

Recommended Books: Latest Edition of the Following Books

1. Leon Shklar and Richard Rosen, Wiley;Web Application Architecture: Principles, protocols and practices 2nd Edition (May 5, 2009). ISBN-10:047051860X
2. Jeffrey C. Jackson,Web Technologies: A Computer Science Perspective, Prentice Hall; 1st Edition (August 27, 2006). ISBN-10:0131856030

Course Name: Cyber Security	**Course Code: CSC-429**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Information Security	

Course Objective:

The course is all about concept, issues covering cyber security.

Course Outline

Basic security concepts, Information security terminology, Malware classifications, Types of malware. Server side web applications attacks. Cross-site scripting, SQL Injection, Cross-site request forgery, Planning and policy, Network protocols and service models. Transport layer security, Network layer security, Wireless security, Cloud & IoT security.

Course Outcomes: After completion of the course students will be able to:

- Understand cyber security and its issues.
- Prevet from cyber crimes.

Recommended Books: Latest Edition of the Following Books

1. Mark Ciampa, Security+ Guide to Network Security Fundamentals Latest Edition,2017,Cengage Learning.
2. Randall J.Boyle, Corporate Computer Society, 3rd Edition, 2012, Pearson.

Course Outlines of Courses of Semester 5th

Course Name: Probability and Statistics	**Course Code: STATS-402**
Course Structure: Lectures: 3, Labs: 1	**Credit Hours: 0**
Prerequisites:	

Course Objective:

- The course is designed to enable the students to understand basic concepts of Statistics, descriptive statistics and probability; conditional probability, random variables and probability distributions.

Course Outlines:

Introduction to Statistics: Descriptive Statistics, Graphical presentation of data, Histogram, Bar charts, Pie charts, box-plot, stem and leaf plot. Measures of Central Tendency; mean, median and mode. Measure of dispersion; Variance and standard deviation; properties. Co-efficient of variation. Correlation and regression. Hypothesis testing. Introduction to counting techniques; Permutation, combination. Basic concept of probability, random experiment, event, sample space. Laws of probability, conditional probability, Bayes theorem with application to discrete and continuous random variable. Random variables and Probability Distributions; Discrete Random Variables, Bernoulli trials, Binomial and Poisson distributions. Continuous Random Variable, probability density function and its properties. Normal Distribution and its properties.

Course Outcomes: After completion of the course students will be able to:

- Demonstrate basic descriptive statistics and analyse and interpret data.
- Demonstrate the basic knowledge of probability and probability distributions.
- Use basic counting techniques (multiplication rule, combinations, and permutations) to compute probability and odds.

Recommended Books: Latest Edition of the Following Books

1. Clark, G.M. and Cooke, D. (1998), "A Basic Course in Statistics" 4th ed, Arnold, London.
2. Chaudhry. S.M.and Kamal, S. (1996), "Introduction to Statistical Theory" Parts I & II, 6th ed, Ilmi Kitab Khana, Lahore, Pakistan.
3. Mclave, J.T., Benson, P.G. and Snitch, T. (2005) "Statistics for Business & Economics" 9t ed, Prentice Hall, New Jersey.
4. Spiegel, M.R., Schiller, J.L. and Sirinivasan, R.L. (2000) "Probability and Statistics", 2nd ed. Schaums Outlines Series. McGraw Hill. NY.
5. Walpole, RE., Myers, R.H and Myers, S.L. (1998), 'Probability and Statistics for Engineers and Scientist" 6th edition, Prentice Hall, NY.
6. Weiss, N.A. (1997), "Introductory Statistics" 4th ed. Addison-Wesley Pub. Company, Inc.

Course Name: Theory of Automata	**Course Code: CSC-502**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- Explain and manipulate the different concepts in automatatheory and formal languages such as formal proofs, automata, regular expressions, Turing machines etc.
- Prove properties of languages, grammars and automatawith rigorously formal mathematical methods

Course Outline

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

Course Outcomes: After completion of the course students will be able to:

- Design of automata, RE and CFG
- Transform between equivalent NFAs, DFAs and Res
- Define Turing machines performing simple tasks
- Differentiate and manipulate formal descriptions of languages, automata and grammars with focus on regular and context-free languages, finite automata and regular expressions

Recommended Books: Latest Edition of the Following Books

1. Daniel I. A. Cohen, Introduction to computer theory, 2nd Edition,1996,wiley.
2. Elaine Rich,, Automata, Computability and Complexity: Theory and Applications, 2011
3. Peter Linz, An Introduction to Formal Languages and Automata, 4th edition, 2006, Jones &Bartlett Publishers.
4. S. P. Eugene, Kavier, Theory of Automata, Formal Languages and Computation, 2005, New Age Publishers

Course Name: Database System	**Course Code: CSC-524**
Course Structure: Lectures: 3, Labs: 1	**Credit Hours: 4**

Prerequisites: None

Course Objective:

- The course aims to introduce basic database concepts, different data models, data storageand retrieval techniques and database design techniques.
- The course primarily focuses onrelational data model and DBMS concepts.

Course Outlines:

Basic database concepts, Database approach vs. file based system, database architecture, three level schema architecture, data independence, relational data model, attributes, schemas, tuples, domains, relation instances, keys of relations, integrity constraints, relational algebra, selection, projection, Cartesian product, types of joins, normalization, functional dependencies, normal forms, entity relationship model, entity sets, attributes, relationship, entity-relationship diagrams, Structured Query Language (SQL), Joins and sub-queries in SQL, Grouping and aggregation in SQL, concurrency control, database backup and recovery, indexes, NoSQL systems.

Course Outcomes: After completion of the course students will be able to:

- Explain fundamental database concepts.
- Design conceptual, logical and physical database schemasusing different data models.
- Identify functional dependencies and resolve databaseanomalies by normalizing database tables.
- Use Structured Query Language (SQL) for databasedefinition and manipulation in any DBMS

Recommended Books: Latest Edition of the Following Books

1. Thomas Connolly and Carolyn Begg Database Systems: A Practical Approach to Design, Implementation, and Management,6th Edition,2014,Pearson.
2. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom, Database Systems: The Complete Book, 2nd Edition,2008,Pearson.
3. Avi Silberschatz, Henry F. Korth and S.Sudarshan. Database System Concepts, 6th Edition,2010, McGraw Hills.
4. Raghu Ramakrishnan, Johannes Gehrke Database Management Systems, 3rd Edition ,2002,McGraw Hills.

Course Name: Computer Networks	**Course Code: CSC-515**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- This course introduces the basic concept of computer network to the students. Network layers, Network models (OSI, TCP/IP) and protocol standards are part of the course.

Course Outlines:

Introduction and protocols architecture, basic concepts of networking, network topologies, layered architecture, physical layer functionality, data link layer functionality, multiple access techniques, circuit switching and packet switching, LAN technologies, wireless networks, MAC addressing, networking devices, network layer protocols, IPv4 and IPv6, IPaddressing, sub netting, CIDR, routing protocols, transport layer protocols, ports andsockets, connection establishment, flow and congestion control, application layer protocols,latest trends in computer networks.

Course Outcomes: After completion of the course students will be able to:

- Describe the key terminologies and technologies ofcomputer networks

- Explain the services and functions provided by each layerin the Internet protocol stack.
- Identify various internetworking devices and protocolsand their functions in a networking
- Analyze working and performance of key technologies,algorithms and protocols
- Build Computer Network on various Topologies

Recommended Books: Latest Edition of the Following Books

1. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 6th edition, 2016,Pearson.
2. Andrew S. Tanenbaum, Computer Networks, 5th Edition, 2010, Pearson.
3. William Stallings, Data and Computer Communications, 10th Edition,2013,Pearson.
4. Behrouz A. Forouzan, Data Communication and Computer Networks, 5th Edition,2006, McGraw Hill.

Course Name: Software Engineering	**Course Code: CSC-533**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- The students will study techniques for software verification, validation and testing.
- They would also study reliability and performance issues in software design and development.
- Upon successful completion of this course the student will be to understand the importance of software engineering to computer science and the most important general approaches to structuring the software production process, analyze the requirements for a software system and produce a software design from requirements (Data Flow Diagram (DFD)), assess software productivity using metrics, use different testing techniques used in software engineering to test software systems, manage the important issues for planning a project.

Course Outlines:

Nature of Software, Overview of Software Engineering, Professional software development,Software engineering practice, Software process structure, Software process models, Agile software Development, Agile process models, Agile development techniques, Requirementsengineering process, Functional and non-functional requirements, Context models,Interaction models, Structural models, behavioral models, model driven engineering, Architectural design, Design and implementation, UML diagrams, Design patterns, Softwaretesting and quality assurance, Software evolution, Project management and project planning,configuration management, Software Process improvement

Course Outcomes: After completion of the course students will be able to:

- Describe various software engineering processes andactivates
- Apply the system modeling techniques to model amedium size software systems
- Apply software quality assurance and testing principles tomedium size software systems
- Discuss key principles and common methods for softwareproject management such as scheduling, size estimation, cost estimation and risk analysis

Recommended Books: Latest Edition of the Following Books

1. Sommerville I, Software Engineering, 10th Edition, 2014, Pearson Inc.
2. Pressman R. S.& Maxim B. R.,Software Engineering, A Practitioner's Approach, 8thEdition, 2014, McGraw-Hill

Course Outlines of Courses of Semester 6th

Course Title: Financial Accounting I	**Course Code: MS-305**
Course Structure: Lectures , Assignmens, quiz & practice session of numerical	**Credit Hours: 3**
Prerequisites: Nil	

Course Objectives

This course aims at giving students knowledge about the basic principles of accounting and to provide students understanding of process and function of financial reporting. Whilst a large proportion of the course is aims at understanding accounting as a process, taking a preparer‟ perspective, and emphasizing the importance of accounting in the process of decision making in an organization..

Course Outline

Accounting information for decision making, Basic Financial Statements, Financial Statements-Concept of Assets, Liabilities and Owner's Equity, The Accounting Equation, **Generally Accepted Accounting Principles-** Business Entity Principle, Going Concern Principle, Historical Cost Principle, Stable Dollar Principle, Adequate Disclosure Principle, Time Period Principle, Statement of Financial Position-Income Statement, Cash- Flow Statement, Exercise and problems. **Forms of Business Organizations-**Sole proprietorship, Partnership,Corporation.**The Accounting Cycle: Capturing Economic Event-** The Journal. Debts And Credits, Posting, The Ledger, The Use of Account, Net income, Revenue and Expense, The Trial Balance, Uses and limitations of Trial Balance, Adjusting entries, Types of adjusting entries, Characteristics of Adjusting Entries, The Adjusted Trial Balance, Preparing Financial Statements, Closing entries ,After-Closing Trial Balance
Exercise and problems

Intended Learning Outcomes (ILOs)

After studying this course, the student will be able to understand:

1. The language of accounting and financial reporting.
2. Complete Accounting Cycle.
3. Preparation and the role of Journal, Ledger, and subsidiary books.
4. Preparation of balance sheet, profit and loss account and cash flow statement

Recommended Books: Latest Edition of the Following Books

1 .Meigs, R., & Meigs, M. (1998). Accounting: A Basis for Decision Making (11th ed. or Latest Edition). Publisher.
2. Williams, J. R., Haka, S. F., & Bettner, M. S. (2000). Financial & Managerial Accounting (Latest Edition). Prentice Hall.
3. Horngren, C. T., Sundem, G. L., Schatzberg, J. O., & Burgstahler, D. (Year of 2002). Introduction to Management Accounting (16th ed. or Latest Edition). Prentice Hall.
4. Warren, C. S., Reeve, J. M., & Duchac, J. E. (2019). Financial and Managerial Accounting (15th ed. or Latest Edition). Cengage Learning.
5. Weygandt, J. J., Kimmel, P. D., & Kieso, D. E. (2007). Financial Accounting (9th ed. or Latest Edition). Wiley.

Course Name: Basics of Marketing	**Course Code: MS-402**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours:** 3
Prerequisites: None	

Course Objectives

This course is designed to introduce foundations of marketing as they relate to the whole business enterprise. This course will focus on developing an understanding of key marketing concepts.

The objective of this course, specifically is to enhance the conceptual knowledge of marketing as applicable to decision making process with a focus on tactical marketing mix decisions. Further it will provide the student with a comprehensive framework to evaluate marketing decisions and to create successful marketing initiatives. The course, will therefore, provide an understanding of the principles of marketing in relation to the product and services including the planning process, organizing the marketing functions, implementing the marketing decisions keeping in mind the ethical, legal and societal consideration.

Course Contents

Introduction to Marketing, An Overview of market and marketing, Definitions, Marketing and the Society,Importance and Scope, Marketing Environments, Introduction to Products and Services, Brands Brand and its elements, Sponsorship,Introduction to Market Segmentation,Product Development, Consumer Behavior- types, role, Consumer Buying Behavior/ Decision Process, Pricing,Distribution, Promotion, Promotion Approaches, Trade Promotion, Consumer Sales Promotion,Sponsorship and •Event Marketing

Indented Learning Outcomes

By the end of this course it is expected that the student will be able:

- Understand the marketplace and the consumers.
- Understand the elements in marketing mix and their application in marketingdecisions.
- Outline the functions of marketing communication.
- Understand how to incorporate social responsibility and ethics in marketing.
- Understand the importance of customer relationship and the creation of customervalue.

Recommended Books: Latest Edition of the Following Books

1. Kotler, P. (2000). Marketing management: Analysis, planning, implementation, and control. Prentice Hall.
2. Trout, J., & Ries, A. (2000). Positioning: The battle for your mind. McGraw-Hill.
3. Godin, S. (2005). Purple cow: Transform your business by being remarkable. Penguin.
4. Cialdini, R. B. (2006). Influence: The psychology of persuasion. HarperCollins.
5. Heath, C., & Heath, D. (2007). Made to stick: Why some ideas survive and others die. Random House.
6. Gladwell, M. (2000). The tipping point: How little things can make a big difference. Little, Brown.

Course Name: Parallel and Distributed Computing	**Course Code: CSC-517**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Operating Systems

Course Objective:

- The course focuses on learning of parallel and distributed architecture.
- To know how to get benefited from parallel and distributed architecture

Course Outlines:

Asynchronous/synchronous computation/communication, concurrency control, fault tolerance, GPU architecture and programming, heterogeneity, interconnection topologies, load balancing, memory consistency model, memory hierarchies, Message passing interface(MPI), MIMD/SIMD, multithreaded programming, parallel algorithms & architectures, parallel I/O, performance analysis and tuning, power, programming models (data parallel, task parallel, process-centric, shared/distributed memory), scalability and performance studies, scheduling, storage systems, synchronization, and tools (Cuda, Swift, Globus,Condor, Amazon AWS, OpenStack, Cilk, gdb, threads, MPICH, OpenMP, Hadoop, FUSE).

Course Outcomes: After completion of the course students will be able to:

- Learn about parallel and distributed computers.
- Write portable programs for parallel or distributed architectures using Message-Passing Interface (MPI) Library.
- Analyze complex problems with shared memory.
- Programming with openMP.

Recommended Books: Latest Edition of the Following Books

1. A. S. Tanenbaum and M. V. Steen, Distributed Systems: Principles and Paradigms,2nd Edition, 2007, Prentice Hall.
2. KHwang, J Dongarra and GC. C. Fox, Distributed and Cloud Computing: Clusters, Grids, Clouds, and the Future Internet, 1st Ed, Elsevier, 2011.

Course Name: Information Security	**Course Code: CSC-521**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- This course provides a broad overview of the threats to the security of information systems,the responsibilities and basic tools for information security, and the levels of training and expertise needed in organizations to reach and maintain a state of acceptable security.
- It covers concepts and applications of system and data security. Areas of particular focus include secure network design, implementation and transition issues, and techniques for responding to security breaches.

Course Outlines:

Information security foundations, security design principles; security mechanisms, symmetric and asymmetric cryptography, encryption, hash functions, digital signatures, keymanagement, authentication and access control; software security, vulnerabilities and protections, malware, database security; network security, firewalls, intrusion detection; security policies, policy formation and enforcement, risk assessment, cybercrime, law and ethics in information security, privacy and anonymity of data.

Course Outcomes: After completion of the course students will be able to:

- Explain key concepts of information security such as design principles, cryptography, risk management, andethics

- Discuss legal, ethical, and professional issues ininformation security
- Apply various security and risk management tools forachieving information security and privacy
- Identify appropriate techniques to tackle and solveproblems in the discipline of information security

Recommended Books: Latest Edition of the Following Books

1. William Stallings Computer Security: Principles and Practice, 3rd edition ,2017, Pearson.
2. M. Whitman and H. Mattord Principles of Information Security, 6th edition,2017,Cengage Learning.
3. Dieter Gollmann Computer Security, 3rd edition, 2011, Willey.
4. William Easttom, Computer Security Fundamentals, 3rd edition Official (ISC)2 Guide to the CISSP CBK, 2019.

Course Name: Artificial Intelligence	**Course Code: CSC-543**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Object Oriented Programming

Course Objective:

- Artificial Intelligence has emerged as one of the most significant and promising areas of computing.
- This course focuses on the foundations of AI and its basic techniques like Symbolic manipulations, Pattern Matching, Knowledge Representation, Decision Making and Appreciating the differences between Knowledge, Data and Code.
- AI programming language Python has been proposed for the practical work of this course.

Course Outlines:

An Introduction to Artificial Intelligence and its applications towards Knowledge Based Systems; Introduction to Reasoning and Knowledge Representation, Problem Solving by Searching (Informed searching, Uninformed searching, Heuristics, Local searching, Min- max algorithm, Alpha beta pruning, Game-playing); Case Studies: General Problem Solver,Eliza, Student, Macsyma; Learning from examples; ANN and Natural Language Processing; Recent trends in AI and applications of AI algorithms. Python programming language will be used to explore and illustrate various issues and techniques in Artificial Intelligence.

Course Outcomes: After completion of the course students will be able to:

- Understand the fundamental constructs of Pythonprogramming language.
- Understand key concepts in the field of artificialintelligence
- Implement artificial intelligence techniques and casestudies

Recommended Books: Latest Edition of the Following Books

1. Russell, S. and Norvig, P. Artificial Intelligence. A Modern Approach, 3rd ed, 2015, PrenticeHall, Inc..
2. Norvig, P., Paradigms of Artificial Intelligence Programming: Case studies in CommonLisp, 1992 Morgan Kaufman Publishers, Inc..
3. Luger, G.F. and Stubblefield, W.A., AI algorithms, data structures, and idioms in Prolog,Lisp, and Java, 2009, Pearson Addison-Wesley.
4. Severance, C.R., Python for everybody: Exploring data using Python 3.2016, CreateSpace Independent Publ Platform.
5. Miller, B.N., Ranum, D.L. and Anderson, J., Python programming in context. 2019, Jones & Bartlett Pub.
6. Joshi, P.,Artificial intelligence with python. 2017, Packt Publishing Ltd.

Course Outlines of Courses of Semester 7th

Course Name: Compiler Construction	**Course Code: CSC-629**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Theory of Automata	
Course Objective: The course focusses on understanding and development of Compiler for computer programming	
Course Outline Introduction to interpreter and compiler. Compiler techniques and methodology;Organization of compilers; Lexical and syntax analysis; Parsing techniques. Types of parsers, top-down parsing, bottom-up parsing, Type checking, Semantic analyser, Grammers and its types, Parser types .Objectcode generation and optimization, detection and recovery from errors.	
Course Outcomes: After completion of the course students will be able to: • Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation • Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines • Design and implement a compiler using a softwareengineering approach • Use generators (e.g. Lex and Yacc)	
Recommended Books: Latest Edition of the Following Books 1. A. V. Aho, R. Sethi and J. D. Ullman Compilers: Principles, Techniques, and Tools, 2006 ,Addison-Wesley, 2nd ed.. 2. D. Grune, H. E. Bal, C. J. H. Jacobs, K. G. Langendoen Modern Compiler Design, 2003,John Wiley. 3. A. W. Appel, M. Ginsburg, Modern Compiler Implementation in C,2004, CambridgeUniversity Press.	

Course Name: Analysis of Algorithms	**Course Code: CSC-672**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**
Prerequisites: Data Structures	
Course Objective: • Detailed study of the basic notions of the design of algorithms and the underlying data structures. Several measures of complexity are introduced. Emphasis on the structure, complexity, and efficiency of algorithms.	
Course Outlines: Introduction; role of algorithms in computing, Analysis on nature of input and size of inputAsymptotic notations; Big-O, Big Ω, Big Θ, little-o, little-ω, Sorting Algorithm analysis, loop invariants, Recursion and recurrence relations; Algorithm Design Techniques, Brute Force Approach, Divide-and-conquer approach; Merge, Quick Sort, Greedy approach;Dynamic programming; Elements of Dynamic Programming, Search trees; Heaps; Hashing;Graph algorithms, shortest paths, sparse graphs, String matching; Introduction to complexityclasses.	
Course Outcomes: After completion of the course students will be able to: • Explain what is meant by “best”, “expected”, and “worst”case behavior	

of an algorithm

- Identify the characteristics of data and/or other conditionsor assumptions that lead to different behaviors.
- Determine informally the time and space complexity ofsimple algorithms
- List and contrast standard complexity classes
- Use big O, Omega, Theta notation formally to give asymptotic upper bounds on time and space complexity ofalgorithms
- Use of the strategies(brute-force, greedy, divide-and- conquer, and dynamic programming) to solve an appropriate problem
- Solve problems using graph algorithms, including single- source and all-pairs shortest paths, and at least one minimum spanning tree algorithm
- Trace and/or implement a string-matching algorithm

Recommended Books: Latest Edition of the Following Books

1. Thomas H. Corman, Charles E. Leiserson,Ronald L. Rivest and Clifford Stein, Introduction to Algorithms,2009, MIT Press.
2. Jon Kleinberg, Eva Tardos , Algorithm Design, 2013/2014,Pearson.
3. Robert Sedgewick, Kevin Wayne, Algorithms, 2011, Addison-Wesley.

Course Outlines of Courses of Semester 8th

Course Name: Advance Database Management Systems	**Course Code: CSC-628**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Database Systems	

Course Objective:

- Advanced Database Management Systems is an extension to "Database Systems" course. The aim of the course is to enhance the previous knowledge of database systems by deepening the understanding of the theoretical and practical aspects of the database technologies, and showing the need for distributed database technology to tackle deficienciesof the centralized database systems.
- It focuses to introduce the basic principles and implementation techniques of distributed database systems, and expose emergingresearch issues in database systems and application development.

Course Outline

Introduction to advance data models such as object relational, object oriented. File organizations concepts, Transactional processing and Concurrency control techniques, Recovery techniques, Query processing and optimization, Database Programming (PL/SQL,T-SQL or similar technology), Integrity and security, Database Administration (Role management, managing database access, views), Physical database design and tuning, Distributed database systems, Emerging research trends in database systems, MONGO DB,NO SQL (or similar technologies).

Course Outcomes: After completion of the course students will be able to:

- Understanding advance data models, technologies andapproaches for building distributed database systems.
- Applying the models and approaches in order to becomeenabled to select and apply appropriate methods for a particular case
- To develop a database solution for a given scenario/ challenging problem in the domain of distributed database systems

Recommended Books: Latest Edition of the Following Books

1. Thomas Connolly and Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation, and Management, 2014,Pearson.
2. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, 2002, McGraw Hill.
3. Avi Silberschatz, Henry F. Korth and S.Sudarshan, Database System Concepts, 2010,Mcraw Hill.
4. Hector Garcia-Molina, Jeffrey, Database Systems: The Complete Book, 2008,Pearson.

Course Outlines of BSSE

Note: The course outlines of courses common to both BSCS nnd BSSE are mentioned above

Course Details of Domain Core (BSSE Engineering)

Course Name: Software Design and Architecture	**Course Code: CSC-504**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Software Requirements Engineering	
Course Objective: • To learn role of design and its major activities within the OO software development process, with focus on the Unified process.	
Course Outline Software Design Concepts, Design principles, Object-Oriented Design with UML, Systemdesign and software architecture, Object design, Mapping design to code, User interface design, Persistent layer design, Web applications design, State machine diagrams and modeling, Agile software engineering, Design Patterns, Exploring inheritance, Interactive systems with MVC architecture, Software reuse. Architectural design issues, , Software Architecture, Architectural Structures & Styles-, Architectural Patterns, Architectural & Design Qualities, Quality Tactics, Architecture documentation, Architectural Evaluation, Model driven development.	
Course Outcomes: After completion of the course 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. • Comprehend the advantages of consistent and reliable software design. • Design OOD models and refine them to reflect implementation details • Apply and use UML to visualize and document the design of software systems. • Implement the design model using an object-oriented programming language.	
Recommended Books: Latest Edition of the Following Books 1. Roger S. Pressman, Bruce R.Maxim, Software Engineering: A Practitioner's Approach, 8th Ed, 2015, McGraw-Hill Education. 2. Brahma Dathan, SarnathRamnath,Object-Oriented Analysis, Design and Implementation, 2nd Ed, 2014, Universities Press, India. 3. Hassan Gomaa, Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures, 2011, Cambridge University Press. 4. Eric Freeman, Elisabeth Freeman, Kathy Sierra and Bert Bates Head First Design Patterns, 2004, O'Reilly Media, Inc.	

Course Name: Software Construction and Development	**Course Code: CSC-631**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Software Design and Architecture	
Course Objective: • The course focuses on learning and use of OO software development process.	
Course Outline Software development process, Software engineering process infrastructure, Software engineering process improvement, Systems engineering life cycle models, Process implementation, Levels of process definition, Life	

cycle model characteristics, Individual and team software process, Lehman's Laws, code salvaging, and configuration management. Martin Fowler's refactoring concepts and their application to small projects. Apply Michael Feathers' "legacy code" concepts. Exception handling, making methods robust by having them check their inputs sent from calling objects. Software configuration management, Release management, Software configuration management processes, Software deployment processes, Distribution and backup, Evolution processes and activities, Basic concepts of evolution and maintenance, Working with legacy systems, Refactoring, Error handling, exception handling, and fault tolerance. Personal reviews (design, code, etc.), Peer reviews (inspections, walkthroughs, etc.).

Course Outcomes: After completion of the course 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.

Recommended Books: Latest Edition of the Following Books

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

Course Name: Software Project Management	**Course Code: CSC-538**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Software Engineering

Course Objective:

- The course focus on principles of the project lifecycle and how to identify opportunities to work with learners on relevant and appropriate project scenarios to share this understanding.

Course Outlines

Introduction to Software Project Management, Project Management concepts, ProjectManagement Tools, PMI's Knowledge areas, PMI Framework, PMI Process Groups.Understanding Organizations. Project Planning, Project Evaluation, Selection of anAppropriate Approach in Project, Software Effort Estimation, Activity Planning, Risk Management, Evaluating the Risks to the Schedule, Risk Control, ConfigurationManagement and Maintenance, Environment for Configuration Control, Resource Allocation, Monitoring & Control, Review and Evaluation, Challenges of Outsourcing in Project Management

Course Outcomes: After completion of the course students will be able to:

- Explain principles of the project lifecycle and how to identify opportunities to work with learners on relevant and appropriate project scenarios to share this understanding.
- Critically evaluate and discuss the issues around project management and its application in the real world with course participants and learners.
- Choose project management techniques for IT projects to initiate, plan, execute and evaluate a project and work in teams to create a project plan for a project scenario that includes key tasks, critical path, dependencies and a realistic timeline.
- Present strategies for gaining confidence in managing projects through simple project planning examples.

Recommended Books: Latest Edition of the Following Books

1. Bob Hughes and Mike Cotterell, Software Project Management, 2009, McGraw-HillEducation.
2. A Guide to the Project Management Body of Knowledge, 5th Edition (PMBOKGuides),
3. Murali K. Chemuturi and Thomas M. Cagley Jr., J, Mastering Software Project Management: Best Practices, Tools and Techniques,2010, Ross Publishing.
4. Robert K. Wysocki ,Effective Project Management: Traditional, Agile, Extreme,2011 ,Wiley.

Course Name: Software Quality Engineering	**Course Code: CSC-516**
Course Structure: Lectures: 3. Labs: 0	**Credit Hours: 3**
Prerequisites: Software Engineering	
Course Objective: • This course focuses on through insight of software testing and software quality assurance principles.	
Course Outlines: Software Quality, Software Quality Attributes, Quality Engineering., Testing: Concepts, Issues, and Techniques, Software testing lifecycle., Testing Scopes., Testing Approaches., Testing Concepts., Test Planning Process, Introduction to testing process, Requirement of software test planning, Testing documentation, Reporting and historical data recording., Software testing techniques, Testing philosophies , Testing strategies, Model based testing, Software testing techniques, Testing using models, Domain and combinatorial testing, Unit and integration testing, Acceptance testing, Test automation, Slicing, Software reliability models and engineering, Introduction, Exponential model., Reliability growth models, Modeling process, Software inspections, Software reviews, Inspection checks and metrics, Quality Models, Models for quality assessment, Product quality metrics, Quality Measurements, In-Process metrics for software testing, In-Process quality management, Effort/outcome models, System testing, Introduction to sub-system testing, From functional to system aspects of testing, System testing, Introduction to system testing, Scenarios development, System testing, Use-cases for testing, Specification-based testing, Open issues on software testing	
Course Outcomes: After completion of the course students will be able to: • Outline software testing and software quality assurance principles. • Prepare test case and test suites for completely testing all aspects of a system under test (SUT) • Analyze which of the software testing techniques are relevant for a particular case and know software reliability analysis tools and techniques. • Compile findings of a quality assurance cycle.	
Recommended Books: Latest Edition of the Following Books 1. Paul Jorgensen, Software Testing, A Craftsman's Approach, 4th Ed,2015, CRC Press,Taylor and Francis Group. 2. Bernard Homes, Fundamentals of Software Testing, 2012, ISTE, Wiley. 3. Software Engineering, "Ian Sommerville, 9th Edition, 2011, Addison Wesley.	

Course Name: Software Requirements Engineering	**Course Code: CSC-580**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**
Prerequisites: Software Engineering	
Course Objective: • This course focuses on software requirements for the development of cost-effective and efficient technical solutions.	
Course Outline: Introduction to Requirements Engineering, Software Requirements, classification ofrequirements, Requirements process, Levels/layers of requirements, Requirement characteristics, Analyzing quality requirements, Software requirements in the context of systems engineering, Requirement evolution, requirement traceability, requirement prioritization, trade-off analysis, risk analysis and impact analysis, Requirement management, interaction between requirement and architecture, Requirement elicitation, elicitation sources and techniques, Requirement specification and documentation, specification sources and techniques, Requirements validation and techniques, Management of Requirements, Introduction to Management, Requirements ManagementProblems , Managing Requirements in an Acquisition Organization, Supplier Organizations, Product Organizations, Requirements engineering for agile methods.	

Course Outcomes: After completion of the course students will be able to:

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

Recommended Books: Latest Edition of the Following Books

1. Wiegers K. &Beatty J, Software Requirements, 3rd Ed, 2012, Microsoft Press.
2. Elizabeth Hull, Ken Jackson and Jeremy Dick, Requirements Engineering,. 3rd Ed, 2011, Springer-Verlag London Limited.
3. Chemuturi M, Requirements Engineering and Management for Software Development Projects, 2013, Springer New York.

Course Contents of Domain Elective

Course Name: Agent Based Software Engineering	**Course Code: CSC-660**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- The course focuses on agent system terminology an development process.

Course Outline

Overview of agent-based software engineering. Methodologies for agent-based modeling,analysis and design: Agent-based Unified Modeling Language (AUML), Agent-based analysis and design, Other agent-based analysis and design methods. Agent communication and knowledge sharing: knowledge level communication among software agents, Knowledge Interchange Format (KIF), Agent-based System Architecture and Organization. FIPA: Foundation for Intelligent Physical Agents: FIPA specification, the application, abstract architecture, agent communication, agent management and agent message transport standards and guidelines.

Course Outcomes: After completion of the course students will be able to:

- Understand the agent system terminology and development process of agent-based systems.
- Understand the techniques to design agent-based system.
- Understand how to modify architecture of the current software systems and restructure them to be agent-based

1. Gerhard Weiss, Multi-agent Systems: A Modern Approach to Distributed Artificial Intelligence, Edt., 1st edition, 2000, MIT Press.
2. Paolo Giorgini, Agent-Oriented Methodologies, 2005, Idea Group Publishing.
3. Fausto Giunchiglia, James J. Odell, GerhardWeiss ,Agent-Oriented Software Engineering III,2002, Springer Verlog - LNCS 2585.

Course Name: Big Data Analytics	**Course Code: CSC-660**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Introduction to Data Science

Course Objective:

- The course is all about fundamental information to get insight into the Challenges with big data.

Course Outline

Introduction to Big Data Analytics, Big Data Platforms, Data Store & Processing using Hadoop, Big Data Storage and Analytics, Big Data Analytics ML Algorithms, Recommendation, Clustering, and Classification, Linked Big Data: Graph Computing and Graph Analytics, Graphical Models and Bayesian Networks, Big Data Visualization, Cognitive Mobile Analytics.

Course Outcomes: After completion of the course students will be able to:

- Provide fundamental information to get insight into the Challenges with big data.
- Understand techniques for storing and processing large amounts of structured and unstructured data
- Application of big data concepts to get valuable information on market trends
- Implement and deploy a sample project for extracting useful information from a mid-sized dataset.

Recommended Books: Latest Edition of the Following Books

1. Jure Leskovec, Anand Rajaraman, Jeff Ullman, Mining of Massive Datasets, 2nd edition, 2011,Cambridge Unv Press.
2. Tom White , Hadoop: The Definitive Guide, 4th edition. 2012,Yahoo Press.
3. Jimmy Lin and Chris ,Data-Intensive Text Processing with Map Reduce, 2010, Morgan and Claypool.

Course Name: Mobile Application Development	**Course Code: CSC-424**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Object Oriented Programming	
Course Objective: • To understand different architectures & framework for Mobile Application development. • To develop mobile applications.	
Course Outline Mobiles Application Development Platform; HTML5 for Mobiles; Android OS: Architecture, Framework and Application Development; iOS: Architecture, Framework; Application Development with Windows Mobile; Eclipse; Fragments; Calling Built-in Applications using Intents; Displaying Notifications; Components of a Screen; Adapting to Display Orientation; Managing Changes to Screen Orientation; Utilizing the Action Bar; Creating the User Interface; Listening for UI Notifications; Views; User Preferences; Persisting Data; Sharing Data; Sending SMS Messages; Getting Feedback; Sending E- mail; Displaying Maps; Consuming Web Services Using HTTP; Web Services: Accessing and Creating; Threading; Publishing, Android Applications; Deployment on App Stores; Mobile Programming Languages; Challenges with Mobility and Wireless Communication; Location-aware Applications; Performance/Power Tradeoffs; Mobile Platform Constraints; Emerging Technologies.	
Course Outcomes: After completion of the course students will be able to: • Discuss different architectures & framework for Mobile Application development. • Develop mobile applications using current software development environments. • Compare the different performance tradeoffs in mobile application development.	
Recommended Books: Latest Edition of the Following Books 1. Reto Meier, Professional Android application development, 2015, Wrox Programmer toProgrammer. 2. Conway, J., Hillegass, A., & Keur, C. iOS Programming: The Big Nerd Ranch Guide,2014,Addison-Wesley . 3. Phillips, B. & Hardy, B., Android Programming: The Big Nerd Ranch Guides, 2014, Addison-Wesley.	

Course Name: Visual Programming	**Course Code: CSC-531**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Object Oriented Programming	
Course Objective: • The course focuses on visual programming language as building blocks to develop correct, coherent programs.	
Visual Programming Basics; Introduction to Events; Fundamentals of Event-driven Programming, message handling, user interfaces, graphics device interface, painting and drawing, windows management, input devices, resources, string and menu resource, dialogs and windows controls, common controls, dynamic link libraries, threads and synchronization, network programming, Building Class Libraries at the Command Line, Class Libraries, Using References, Assemblies, Private Assembly Deployment, Shared Assembly Deployment, Configuration Overview, Configuration Files, Programmatic Access to Configuration, Using SDK Tools for Signing and Deployment, Metadata, Reflection, Late Binding, Directories, Files, Serialization, Attributes, Memory Management and Garbage Collection, Threading and Synchronization, Asynchronous Delegates, Application Domains, Marshal by Value, Marshal by Reference, Authentication and Authorization, Configuring Security, Code Access Security, Code Groups, Evidence, Permissions, Role-Based Security, Principals and Identities, Using Data Readers, Using Data Sets, Interacting with XML Data, Tracing Event Logs, Using the Boolean Switch and Trace Switch Classes, Print	

Debugging Information with the Debug Class, Instrumenting Release Builds with the Trace Class, Using Listeners, and Implementing Custom Listeners.

Course Outcomes: After completion of the course students will be able to:

- Use the different elements of a visual programming language as building blocks to develop correct, coherent programs.
- Program using the fundamental software development process, including design, coding, documentation, testing, and debugging.
- Analyze problems, develop conceptual designs that solve those problems, and transform those designs to Visual Programs.

Recommended Books: Latest Edition of the Following Books

1. Deitel and Deitel, Visual C#: How to Program, 6/e Edition, 2017, Prentice Hall / PearsonEducation.
2. J.C. Bradley, A.C. Millspaugh, Programming in C# .NET, 2014, McGraw-Hill.
3. Sharp, J, Microsoft Visual C# 2013 Step by Step (Step by Step Developer), 1stEdition, 2013, Microsoft Press.

Course Name: Numerical Computing	**Course Code:**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: QR1

Course Objective:

- The course objective is to understand computer algebra system to investigate and solve mathematical problems relating to integration, differential equations and approximation

Course Outline

Mathematical preliminaries and error analysis, round-off errors and computer arithmetic, Calculate Divided Differences. Use Divided-difference Table. Find Newton's Interpolation Polynomial. Calculate Interpolation with Equally Spaced Data. Find the Difference Table. Calculate, Newton's Forward & Backward Difference Formulae. UseGauss Formulae. Use Stirling's Interpolation Formula. Use Bessel's Interpolation Formula. Use Everett's Interpolation Formula. Solve Nonlinear Equations. Solve Equations by Bisection Method. Solve Equations by Regula Falsi Method. Solve Equations by Secant Method. Solve Equations by Newton-Raphson Method. Find Fixed Point Iteration. Solve Equations by Jacobi Iterative Methods. Solve Equations by GaussSeidel Method Calculate Numerical Differentiation. Find Numerical Differentiation Formulae Based on Equally Spaced Data. Find Numerical Differentiation Based on Newton's Forward Differences. Find Numerical Differentiation Based on Newton's Backward Differences. Find Numerical Differentiation Based on Stirling's Formula. Find Numerical Differentiation Based on Bessel's Formula. Find Numerical Differentiation Based on Lagrange's Formula. Calculate Error Analysis of Differentiation Formulae. Solve Richardson Extrapolation. Calculate Numerical Integration. Use Trapezoidal Rule with Error Term. Use Simpson's 1/3 Rule with Error Term. Use Simpson's 3/8 Rule with Error Term. Use Composite Numerical Integration. Use Composite Trapezoidal Rule.

Use Composite Simpson's Rule. Find Richardson's Extrapolation. Find Newton-Cotes Closed Quadrature Formulae.

Course Outcomes: After completion of the course students will be able to:

- The student would understand the fundamental concepts of Scientific Programming using programing Language(s)
- Use a computer algebra system to investigate and solve mathematical problems relating to integration, differential equations and approximation.

Recommended Books: Latest Edition of the Following Books

1. Richard L. Burden, J. Douglas Faires , Numerical Analysis, 9th Edition, 2011, Brooks/Cole Boston USA.
2. J.H. Heinbockel Trafford, Numerical Methods for Scientific Computing, 2006, Publishing USA.

Course Name: Web Engineering	**Course Code: CSC-430**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Web Technologies

Course Objective:

- This course focuses on Web application performance, testing and maintenance

Course Outline

Web programming languages (e.g., HTML5, CSS 3, Java Script, PHP/JSP/ASP.Net), Design principles of Web based applications, Web platform constraints, Software as a Service (SaaS), Web standards, Responsive Web Design, Web Applications, Browser/Server Communication, Storage Tier, Cookies and Sessions, Input Validation, Full stack state management, Web App Security - Browser Isolation, Network Attacks, Session Attacks, Large scale applications, Performance of Web Applications, Data Centers, Web Testing and Web Maintenance.

Course Outcomes: After completion of the course students will be able to:

- Discuss how web standards impact software development.
- Describe the constraints that the web puts on developers.
- Design and implement a simple web application.
- Review an existing web application against a current web standard.

Recommended Books: Latest Edition of the Following Books

1. Rajiv Chopra, Web Engineering, 2016, Prentice-Hall of India.
2. Emilia Mendes and Nile Mosley, Web Engineering, 2010, Springer Verlag.
3. Roger S. Pressman, Web Engineering: A Practitioners' Approach, 2008, McGraw Hill.
4. Dynamic HTML: The Definitive Reference: A Comprehensive Resource for XHTML,CSS, DOM, JavaScript 3rd Edition, O'Reilly Media 2007.
5. David Flanagan., JavaScript: The Definitive Guide, 8th Edition, 2014, O'Reilly Media.

Course Name: Data Encryption and Security	**Course Code: CSC-659**
Course Structure: Lectures:3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

The course intends to learn security provision using cryptography.

Course Outline

Principle of number theory and probability theory, Primes, random numbers, modular arithmetic and discrete logarithms. Cryptographic algorithms and design principles,including conventional and symmetric encryption (DES, IDEA, Blowfish, Rijndael, RC- 4, RC-5), public key or asymmetric encryption (RSA, Diffie-Hellman), key management,hash functions (MD5, SHA-1, RIPEMD-160, HMAC), digital signatures and certificates.

Authentication protocols (X.509, Kerberos), electronic mail security (S/MIME, PGP), web security and protocols for secure electronic commerce (IPSec, SSL, TLS, SET).

Course Outcomes: After completion of the course students will be able to:

- Understand cryptography, digital signature and certificate.
- Develop and use cryptographic techniques for security

Recommended Books: Latest Edition of the Following Books

1. William Stallings , Cryptography and Network Security: Principles and Practice, 6thedition, 2016, Pearson.

Course Name: E-Commerce	**Course Code: CSC605**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Web Engineering

Course Objective:

- To learn concepts and standards related to the discipline of E-Commerce.
- To Analyze complex real world problems found in E- Commerce

Course Outline

An overview of E-Commerce & its business models and concepts, Planning an E-Commerce Framework, Managing Products and Categories, Product Variations and UserUploads, Enhancing the User Experience. The Shopping Basket, The Checkout and OrderProcess, Shipping and Tax, Discounts, Vouchers, and Referrals, Checkout, Taking Payment for Orders, User Account Management, Administration: Dashboard, Managing Products and Categories, Managing Orders, Customers, Refunds, Voucher Codes, Shipping, Deploying, Security, and Maintenance, Web Payment Systems, Social, Legal, and Ethical Issues of E-Commerce, Auctions, Portals, and Communities, SEO.

Course Outcomes: After completion of the course students will be able to:

- Understand the concepts and standards related to the discipline of E-Commerce.
- Analyze complex real world problems found in E- Commerce

Recommended Books: Latest Edition of the Following Books

1. Kenneth Laudon and Carol Guercio Traver ,E-Commerce,13th Edition, 2017, Pearson.
2. Michael Peacock, PHP 5 E-commerce Development,2010, Packt Publishing.
3. Jeffrey F. Rayport, Introduction to E-Commerce, 2nd Edition, 2007, McGraw-Hill.
4. Gary Schneider,, Electronic Commerce, 12th Edition ,2016, Course Technology.

Course Name: Global Software Development	**Course Code: CSC-661**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- To learn principles of the software engineering in context of global software development Evaluate.
- To discuss the issues around global software development and techniques for managing distributed projects.

Course Outline

Introduction to Global Software Development. Global Teams and Organization. Guideline for making the virtual team. The Geography of Coordination. Dealing with Distance. Architectures and Coordination: Reconfiguration of Existing Product Technologies, Identification of Coordination Requirements. Distributed Development Environments: Software configuration management, Awareness among Configuration Management.Challenges of Culture: Managing distances and differences in geographically distributed work groups. The Outsourcing Relationship. Facilitating Cross-site Trust, Cooperation, and Social Capital: Communication and Trust in Global Virtual Teams. Social Networksand Knowledge Networks. Communication and Awareness: dealing with distance. Assessing Coordination Risk.

Course Outcomes: After completion of the course students will be able to:

- Understand the principles of the software engineering in context of global software development
- Evaluate and discuss the issues around global software development and techniques for managing distributed projects.
- Understand Configuration management systems, release management and task assignments in context of distributed projects.
- Acquire strategies for effectively dividing tasks among teams, controlling the communication among teams, planning tasks and collaborating on modular project with the help of realistic examples.

Recommended Books: Latest Edition of the Following Books

1. Christof Ebert ,Global Software and IT: A Guide to Distributed Development, Projects,2011, Wiley.
2. Erran Carmel, Global Software Teams: Collaborating Across Borders and Time Zones, 1999, Prentice Hall.

Course Name: Graph Theory	**Course Code: CSC-658**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- To learn fundamental concepts of Graph Theory.
- To learn how to apply knowledge for application of Graph Theory

Course Outline

Introduction to Graph Theory, Basic definitions, computer representations and propertiesof Graph, Data structure for representing Graphs, Fundamental theorem of Graph Theory,Isomorphic and Special Graphs, Properties of Trees and Forests, Binary tree, Balanced binary tree, Directed and Undirected rooted tree, Minimum Spanning Tree algorithms and implementation, Path and Distance in graphs, Shortest path algorithms and implementation, Cycle and distance in weighted graph and digraphs, Distance algorithmsand implementation, Eulerian graphs and Hamiltonians graphs with applications, Flow networks, Max-flow Min-cut Theorem, Graph coloring, Edge coloring, Planar graphs,Four color theorem, Deadlock of computer system, Matching Algorithms, Dominance &Ramsey theory.

Course Outcomes: After completion of the course students will be able to:

- To introduce the fundamental concepts of Graph Theory.
- To provide knowledge for application of Graph Theory in subsequent courses in the design and analysis of algorithms, computability theory, software engineering, and computer systems.

Recommended Books: Latest Edition of the Following Books

1. Fournier ,Graph Theory & Applications, 1st Edition, 2011, Wiley-ISTE.
2. Chartrand, Applied Algorithmic Graph Theory, 1st Edition, 1995, McGraw-Hill College.
3. Jonathan, Handbook of Graph Theory, Series Edition, 2004, CRC Press.
4. J. A. Bondy, Graph Theory with Applications, 8th Edition, 1982,ElsevierUSA.

Course Name: Information Systems Audit	**Course Code: CSC-657**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- To learn concepts and standards related to the discipline of Information System Audit.
- To review Audit Information Systems

Course Outline

Introduction to Auditing, IS Audit charter, Polices, Procedures, The Audit Process, Auditcomputer networks and communication, Auditing software development, Acquisition, Maintenance, Auditing IT infrastructure, Auditing Management and Organization, Business process re-engineering: IS audit proposal, report, evidence and follow-up, complaint to standard, Enterprise service agreement, IP pro count policies and process, Backup and procedures, Overview of Computer-Assisted Audit Tools and Techniques

Course Outcomes: After completion of the course students will be able to:

- Understand the concepts and standards related to the discipline of Information System Audit.
- Analyze and Audit Information Systems

Recommended Books: Latest Edition of the Following Books

1. Abraham Nyirongo ,Auditing Information Systems: Enhancing Performance of the Enterprise, 2015, Trafford.
2. Ron Weber, Dorling Kindesley ,Information Systems Control and Audit, 2014, Pearson Education.
3. Peter Gregory, CISA® Certified Information Systems Auditor All-in-One Exam Guide, 3rd Edition, 2016, McGraw-Hill Education.

Course Name: Management Information System	**Course Code: CSC-656**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- To learn information technology management.
- Assess and apply IT to solve common business problems.
- Suggest and defend effective solutions to business problems, and design a database application to solve a business problem.

Course Outline

Introduction to Information Systems in Organizations; Business Process and Decision Making; Productivity, Innovation and Strategy; Database and Content Management; Decision Making and Business Intelligence; Competitive Advantage and Business Processes; Networks and Collaboration; ERP and E-commerce, Social

Networking, and Web 3.0; Acquiring Information Systems Through Projects; Security, Governance, and Ethics; Managing Information Security and Privacy
Course Outcomes: After completion of the course students will be able to: • Understand and articulate concepts of information technology management. • Assess and apply IT to solve common business problems. • Suggest and defend effective solutions to business problems and design a database application to solve a business problem. • Explain in detail the ethical aspects of information technology use in the organization and its governance issues.
Recommended Books: Latest Edition of the Following Books 1. D. M. Kroenke, A. Gemino and P. Tingling. P. Experiencing MIS, 4th Edition, 2016, Toronto: Pearson. 2. P. Baltzan, B. Detlor, and C. Welsh, Business driven information systems, 4th Ed, 2015, McGraw Hill Ryerson Press.

Course Name: Multimedia Communication	**Course Code: CSC-654**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**
Prerequisites: None	
Course Objective: • To learn core concepts behind multimedia systems • To learn multimedia programming.	
Course Outline Overview of multimedia systems, Audio/Video fundamentals (representation, human perception, equipment and applications). Audio and video compression (e.g., JPEG, MPEG, H.26X, etc.), scalable coding, perceptual audio encoders. Performance comparison of coding algorithms, Algorithms for image and video processing, multimedia programming.	
Course Outcomes: After completion of the course students will be able to: • Understand fundamental concepts of multimedia systems. • To write performance efficient algorithms.	
Recommended Books: Latest Edition of the Following Books 1. Fred Halsall, Multimedia Communications: Applications, Networks, Protocols, andStandards, 2000,Addison-Wesley. 2. Puri, Multimedia Systems, Standards and Networks, 2002, CRC Press. 3. Steve Heath, Multimedia and Communication Technology,1999, Focal Press. 4. Bill Whyte, Multimedia Telecommunication, 1997, Chapman and Hall.	

Course Name: Real Time Systems	**Course Code: CSC-540**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**
Prerequisites: Software Engineering	
Course Objective: • The course focuses on learning of issues and basic concepts of real-time software development. • To develop ability to make embedded real-time software using appropriate software methods and tools.	

Course Outline

Introduction to Real-Time Systems, Categories, Characteristics and challenges, Requirement Specification and Design, Design fundamentals, Elements of modular design, Concurrency, Real-time & other application areas, Real-Time Operating Systems, Memory management, Fundamental of microprocessor based systems, Input-output interfacing technique, Real-time programming, Real-Time Analysis, Schedulability analysis, Scheduling policies, Designing with rate-monotonic analysis

Course Outcomes: After completion of the course students will be able to:

- Understand the issues and basic concepts of real-time software development.
- Demonstrate the ability to develop embedded real-time software using appropriate software methods and tools.
- Analyze the timing performance of a real-time software design using real-time analysis tools.
- Apply real-time software engineering knowledge in developing a medium to complex program

Recommended Books: Latest Edition of the Following Books

1. Cooling J, Software Engineering for Real-Time Systems, 2002, Addison-Wesley.
2. Burns A., Wellings A.J, Real-time Systems and Programming Languages, 2nd Edition, 2009, Addison Wesley, UK.
3. Ben-Ari M, Principles of Concurrent and Distributed Programming, 2006,Addison-Wesley.

Course Name: Semantic Web	**Course Code: CSC-544**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- To learn concept structure of the semantic web technology and how this technology revolutionizes the World Wide Web and its uses.
- To analyze the concepts of metadata, semantics of knowledge and resource, ontology, and their descriptions in XML-based syntax and web ontology language (OWL).

Course Outline

Introduction to the semantic web, introduction to ontologies, ontology languages for the semantic web, Resource Description Framework (RDF), lightweight ontologies: RDF Schema, Web Ontology Language (OWL), query language for RDF: SPARQL, Ontology Engineering, Semantic web and Web 2.0 and applications of Semantic Web.

Course Outcomes: After completion of the course students will be able to:

- Understand the concept structure of the semantic web technology and how this technology revolutionizes the World Wide Web and its uses.
- Understand the concepts of metadata, semantics of knowledge and resource, ontology, and their descriptions in XML-based syntax and web ontology language (OWL).
- Describe logic semantics and inference with OWL.
- Use ontology engineering approaches in semantic applications program semantic applications with Java API.

Recommended Books: Latest Edition of the Following Books

1. Toby Segaran, Colin Evans, JamieTaylor, Build Flexible Applications with Graph Data, 2009, O'Reilly Media.
2. Pascal Hitzler, Markus Krotzsch,Sebastian Rudolph, Foundations of Semantic Web Technologies,2009. Chapman and Hall.
3. Liyang Yu, Introduction to the Semantic Web and Semantic Web Services,2007. Chapmanand Hall/CRC.

Course Name: Software Engineering Economics	**Course Code: CSC-545**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- To learn economic analysis techniques and their applicability to software engineering
- To learn how to develop software cost estimation skills using industry standards.

Course Outline

Programming aspects, economic aspects, human relations aspects, software trends: cost, social impact, the plurality of SE Means, The GOALS Approach to Software Engineering, The Software Work Breakdown Structure (WBS), Software Maintenance, introduction toCOCOMO, definitions and assumptions, development effort and schedule, phase distribution, The Raylaigh Distribution, interpolation, basic software maintenance effort estimation. Performance Models, Optimal Performance, Sensitivity Analysis, Cost-Effectiveness Models. Cost Drivers: Project Attributes–Modern Programming Practices, Use of Software Tools, Schedule Constraint.

Course Outcomes: After completion of the course students will be able to:

- Overview economic analysistechniques and their applicability to software engineering
- Develop software cost estimation skills using industry standards.

- Critically evaluate and discuss the issues in cost estimation of different applications in the real world with course participants and learners.

Recommended Books: Latest Edition of the Following Books

1. by Pamela T. Geriner,Thomas R. Gulledge, William P. Hutzler , Software Engineering Economics and Declining Budgets, 2012, Springer Verlag.
2. Capers Jones, Estimating Software Costs: Bringing Realism to Estimating, 2nd Edition, 2007, McGraw-Hill Osborne Media.
3. ShariLawrence Pfleeger, Software Cost Estimation and Sizing Methods, Issues, and Guidelines, 2005, Rand Publishing.

Course Name: Information Assurance	**Course Code: CSC-535**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- To learn security governance principles
- To learn and apply management concepts.

Course Outline

Introduction to (IS) Information System (Concept, Design, Functions, Architecture, Components and applications of IS); Secure System Planning and Administration; Information Security Policies and Procedures; Asset Management; Organizational andHuman Security; Cyber Security Management Concepts; NIST Cyber Security Framework;Enterprise Roles and Structures; Strategic Planning; Security Plans and Policies; Contingency Planning; Laws;

Laws and Regulatory Requirements; Security Standards and Controls, Risk Management Process, NIST Risk Management Framework, Security Metrics and Key Performance Indicators (KPIs); Physical Security and Environmental Events; Contingency Planning; Security Education, ISO 27001 Compliance, Training, and Awareness.

Course Outcomes: After completion of the course students will be able to:

- Apply security governance principles; organizational processes; developing and implementing a documented security policy.
- Understand and apply risk management concepts
- To understand the business, legal, and technical knowledge needed to secure vital government and business assets.

Recommended Books: Latest Edition of the Following Books

1. Michael E. Whitman, Principles of Information Security, 6th Edition, 2017, Cengage Learning.
2. Reference Material: CISSP Study Guide, 7th Edition
3. Joseph Boyceand Daniel Jennings, Information Assurance: Managing Organizational IT Security Risks, 2002, Butterworth-Heinemann.
4. Andrew Blyth and Gerald L. Kovacich, Information Assurance: Security in the Information Environment
5. 2006, Springer.

Course Name: Software Metrics	**Course Code: CSC-529**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**
Prerequisites: Software Engineering	
Course Objective: • To learn quantitative and empirical methods are applied to software engineering problems. • To analyze the fundamentals of measurement, experimentation, data collection and analysis.	
Course Outline Overview of software metrics; Basics of measurements; Goal-based framework forsoftware measurement; Software measure classification; Empirical investigation, principles and techniques; Formal experiments: Planning, principles, types and selection;Measuring internal product attributes: size and structure; Measuring cost and effort; Measuring external product attributes: quality and reliability; Software test metrics; Object-oriented metrics	
Course Outcomes: After completion of the course students will be able to: • Explains how quantitative and empirical methods are applied to software engineering problems. • Presents the fundamentals of measurement, experimentation, data collection and analysis • Critically evaluate and discuss different software matrices of different applications in the real world with course participants and learners. • Have a working knowledge of software size measurement (Function Point counting, etc.)	
Recommended Books: Latest Edition of the Following Books 1. N.E. Fenton and J.Bieman,Software Metrics: A Rigorous and Practical Approach, 3rd ed, 2014, CRC Press. 2. C. RavindranathPandian, Software Metrics: A Guide to Planning, Analysis, and Application, 2004, Auerbach Publications, CRC Press Company. 3. Stephen H. Kan , Metrics and Models in Software Quality Engineering, 2nd ed. 2002, Addison-Wesley Professional.	

Course Name: Systems Programming	**Course Code: CSC-546**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**
Prerequisites: None	
Course Objective: • To learn system programming.	
Course Outlines Introduction to the Microsoft Windows ® Operating System, File Processing, MemoryManagement, Memory Mapped Files and DLLs, Process management, Threads and scheduling, Thread synchronization, Inter-process Communication, Input/Output, DeviceDrivers (USB or Parallel Port), File System Drivers, Filter Drivers	
Course Outcomes: After completion of the course students will be able to: • Use windows APIs • Implement synchronization, inter process communication concepts.	
Recommended Books: Latest Edition of the Following Books 1. Johnson M. Hart , Windows System Programming 3rd edition,2005, Addison Wesley 2. Art Baker, The Windows NT Device driver book, 2nd edition,1996 , Prentice Hall.	

Course Name: Data Mining	**Course Code: CSC-634**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Advance Statistics, Introduction to Data Science

Course Objective:

- Expand on the student's understanding and awareness of the concepts of data mining basics, techniques, and application.
- Introduce the concepts of Data Pre-processing and Summary Statistics.
- Introduce the concepts of Frequent Item Set Generation, Associations and Correlations measures.
- Introduce the concepts of Classification, Prediction, and Clustering algorithms. Build on the programming and problem-solving skills developed in previous subjects studied by the student, to achieve an understanding of the development of Classification, Prediction, and Clustering pplications.

Course Outlines

Introduction to data mining and basic concepts, Pre-Processing Techniques & Summary Statistics, Association Rule mining using Apriori Algorithm and Frequent Pattern Trees, Introduction to Classification Types, Supervised Classification (Decision trees, Naïve Bae Classification, K-Nearest Neighbors, Support Vector Machines etc.), Unsupervised Classification (K Means, K Median, Hieratical and Divisive Clustering, Kohonan Self Organizing maps), outlier & anomaly detection, Web and Social Network Mining, Data Mining Trends and Research Frontiers. Implementing concepts using Python

Course Outcomes: After completion of the course students will be able to:

- Apply preprocessing techniques on any given raw data.
- Select and apply proper data mining algorithm to discover interesting patterns
- Analyze and extract patterns to solve problems and point out how to deploy solution
- Evaluate systematically supervised, semi supervised and unsupervised models and algorithms with respect to their accuracy

Recommended Books: Latest Edition of the Following Books

1. Jiawei Han & Micheline Kamber, Jian Pei, Data Mining: Concepts andTechniques, 3rd Edition, 2011, Moran-Kaufmann.
2. Pang-Ning Tan, Michael Steinbach, and Vipin Kumar, Introduction to DataMining, 2005,Pearson.
3. Charu C. Aggarwal, Data Mining: The Textbook, 2015, Springer.
4. D. Hand, H. Mannila, P. Smyth, Principles of Data Mining, 2001, MIT Press.

Course Name: Data Visualization	**Course Code: CSC-644**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Data Warehousing & Business Intelligence

Course Objective:

- Data Visualization is a process of obtaining detailed insights hidden in the data. It is a necessary component in the pipeline of any data science project.
- This course teaches skills specifically in terms of how to effectively present the data and findings. Further, this courseprovides hands on skills using R for data exploration and visualization.

Course Outlines

Introduction of Exploratory Data Analysis and Visualization, Building Blocks and Basic Operations; Types of Exploratory Graphs, single and multi-dimensional summaries, five number summary, box plots, histogram, bar plot and others; Distributions, their representation using histograms, outliers, variance; Probability Mass Functions and their visualization; Cumulative distribution functions, percentile-based statistics, randomnumbers; Modelling distributions, exponential, normal, lognormal, pareto; Probability density functions, kernel density estimation; Relationship between variables, scatter plots, correlation, covariance; Estimation and Hypothesis Testing; Clustering using K-means and Hierarchical; Time series and survival analysis; Implementing concepts with R (or similar language)

Course Outcomes: After completion of the course students will be able to:

- Provides knowledge about importance, necessity, and justification of performing exploratory data analysis andvisualization.
- Introduce various type of charts along with their alternatives solution to show same data from versatile aspects.
- Improving the competency of the students to analyzedifferent problems and select the most appropriate solution.
- Use of R, various recent tools, and technologies to develop hands-on skills for exploratory data analysis andvisualization.
-

Recommended Books: Latest Edition of the Following Books

1. Roger D. Peng, Exploratory Data Analysis with R, 2016.

Course Name: Data Warehousing and Business Intelligence	**Course Code: CSC-641**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Introduction to Data Science

Course Objective:

- Gives an overview about importance & significance of Data Warehousing (DWH) and Business Intelligence (BI).
- Discusses the main concepts and solutions for DWH and BI. Thekey concepts underpinning the logical design, physical design and implementation of data warehouses are appraised. Data collection, data extraction, cleansing, transformation and loading methods are considered along with query optimization techniques.
- Differentiation between OLAP & OLTP. Data Warehousing supports information processing by providinga solid platform of integrated, historical, and consistent data for performing enterprise- wide data analysis.

Course Outlines

Introduction to Data Warehouse and Business Intelligence; Necessities and essentials of Business Intelligence; DW Life Cycle and Basic Architecture; DW Architecture in SQL Server; Logical Model; Indexes; Physical

Model; Optimizations; OLAP Operations, Queriesand Query Optimization; Building the DW; Data visualization and reporting based on Datawarehouse using SSAS and Tableau; Data visualization and reporting based on Cube; Reports and Dashboard management on PowerBI; Dashboard Enrichment; Business Intelligence Tools.

Course Outcomes: After completion of the course students will be able to:

- Demonstrate an appreciation of the role that Data Warehouses and Business Intelligence play in enhancing the decision-making process
- Demonstrate an understanding of the fundamental concepts of the Star and the Snowflake Schema; learn how to design the schema of a DW based on these two models.
- Understand the architecture of DW Systems and be able to specify the advantages and potential problem areas
- Use Analytic SQL to aggregate, analyze and report, and model data.

Recommended Books: Latest Edition of the Following Books

1.W. H. Inmon, Building the Data Warehouse, 2005, Wiley-India Edition.
2.Ralph Kimball, The Data Warehouse Toolkit – Practical Techniques for BuildingDimensional Data Warehouse, 1996, John Wiley & Sons, Inc.
3. Matteo Golfarelli, Stefano Rizzi, "Data Warehouse Design - Modern Principles and Methodologies, 2009, McGraw Hill.

Course Name: Introduction to Data Science	**Course Code: CSC-536**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Artificial Intelligence

Course Objective:

- Data Science is the study of the generalizable extraction of knowledge from data. Being a data scientist requires an integrated skill set spanning mathematics, statistics, machine learning, databases and other branches of computer science along with a good understandingof the craft of problem formulation to engineer effective solutions.
- The aim of this course isto: Introduce students to this rapidly growing field and equip them with some of its basic principles and tools as well as its general mindset. Explain the significance of exploratory data analysis in data science.
- Identify common approaches used for Feature Generation as well as Feature Selection, and finally discuss the Ethical and Privacy issues. Programming language Python has been proposed for the practical work of this course.

Course Outlines

Introduction: What is Data Science? Big Data and Data Science hype, Datafication, Current landscape of perspectives, Skill sets needed; Statistical Inference: Populations and samples,Statistical modeling, probability distributions, fitting a model, Intro to Python; Exploratory Data Analysis and the Data Science Process; Basic Machine Learning Algorithms: Linear Regression, k-Nearest Neighbors (k-NN), k-means, Naive Bayes; Feature Generation and Feature Selection; Dimensionality Reduction: Singular Value Decomposition, Principal Component Analysis; Mining Social-Network Graphs: Social networks as graphs, Clustering of graphs, Direct discovery of communities in graphs, Partitioning of graphs, Neighborhood properties in graphs; Data Visualization: Basic principles, ideas and tools for data visualization; Data Science and Ethical Issues: Discussions on privacy, security, ethics,Next-generation data scientists.

Course Outcomes: After completion of the course students will be able to:

- Describe what Data Science is and the skill sets needed to be a data scientist.
- Apply EDA and the Data Science process in a case study. C3 (Apply)
- Comprehend the fundamental constructs of Python programming language.
- Apply basic machine learning algorithms to solve real world problems of moderate complexity.

Recommended Books: Latest Edition of the Following Books

1. Blum, A., Hopcroft, J., & Kannan, R., Foundations of data science, 2016.
2. Jeffrey S. Saltz, Jeffrey M. Stanton, An Introduction to Data Science, 2017, SAGEPublications.

3. Severance, C.R., Python for everybody: Exploring data using Python 3, 2016, CreateSpaceIndependent Pub Platform.
4. Cathy O'Neil and Rachel Schutt, Doing Data Science, Straight Talk from the Frontline, 2014, O'Reilly.
5. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing andPresenting Data, 2015, EMC Education Services, John Wiley & Sons.

Course Name: Deep Learning	**Course Code: CSC-663**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Artificial Neural Network

Course Objective:

- Deep neural networks have achieved state of the art performance on several compute visionand speech recognition benchmarks. Deep learning algorithms extract layered high and low-level features from raw data.
- With increasing non-line hidden layers, the discriminative power of the network improves. This course builds on the fundamentals of Neural networksand artificial intelligence and covers advanced topics in neural networks, convolutional andrecurrent network structures, deep unsupervised and reinforcement learning.
- It also embedsapplications of these algorithms to several real-world problem in computer vision, speech recognition, natural language processing, game theory, etc.

Course Outlines

Basics of deep learning, learning networks, Shallow vs. Deep learning etc.; Machine learningtheory – training and test sets, evaluation, etc. Theory of Generalization; Multi-layer perceptrons, error back-propagation; Deep convolutional networks, Computational complexity of feed forward and deep convolutional neural networks; Unsupervised deep learning including auto-encoders; Deep belief networks; Restricted Boltzman Machines; Deep Recurrent Neural Networks (BPTT, LSTM, etc.); GPU programming for deep learningCuDNN; Generative adversarial networks (GANs); Sparse coding and auto-encoders; Data augmentation, elastic distortions, data normalization; Mitigating overfitting with dropout, batch normalization, dropconnect; Novel architectures, ResNet, GoogleNet, etc

Course Outcomes: After completion of the course students will be able to:

- Apply deep learning algorithms to real-world problems
- Analyze results from deep learning to select appropriate solutions
- Code the novel neural network architectures from scratch and evaluating the performance on application specific standard benchmarks

Recommended Books: Latest Edition of the Following Books

1. Ian Goodfellow,Yoshua Bengio, Aaron Courville Deep Learning (http://www.deeplearningbook.org/)
2. Francoise Chollet, Deep learning with python, 2017, Manning.

Course Name: Knowledge Representation and Reasoning	**Course Code: CSC-529**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: Artificial Intelligence

Course Objective:

- Knowledge representation is one of the fundamental areas of Artificial Intelligence. It is thestudy of how knowledge about the world can be represented and manipulated in anautomated way to enable agents to make intelligent decisions.
- This course will provide an overview of existing knowledge representation frameworks developed within AI including but not limited to propositional and first-order logic, ontologies, planning, reasoning and decision making under uncertainty.

- The assignments component of the course would providehands-on experience of software like Prolog, Protégé, probabilistic reasoning APIs and toolsto support complex decision making. It is expected that after completing this course, studentswill understand (a) the foundations of Knowledge Representation & Reasoning and (b) which tools and techniques are appropriate for which tasks.

Course Outlines

Propositional Logic, First-order Logic, Horn Clauses, Description Logic, Reasoning using Description Logic, Forward and Backward Chaining in Inference Engines, Semantic Networks, Ontologies and Ontology Languages, Logical Agents, Planning, Rule-based Knowledge Representation, Reasoning Under Uncertainty, Bayesian Networks Representation, Inference in Bayesian Networks, Fuzzy Logic, Inferene using Fuzzy Rules, Markov Models, Commonsense Reasoning, Explainable AI.

Course Outcomes: After completion of the course students will be able to:

- Understand the fundamentals of knowledge representation and reasoning in deterministic situations
- Understand the challenges in representing knowledge and reasoning under uncertainty
- Analyze different situations and apply appropriate knowledge representation frameworks
- Development of hybrid approaches by synergizing the existing framework to solve complex decision-making problems.

Recommended Books: Latest Edition of the Following Books

1. Stuard Russell and Peter Norvig, Artificial Intelligence: A Modern Approach, 3rd Ed, 2015,Pearson.
2. David Poole and Alan Mackworth, Artificial Intelligence: Foundations of Computational Agents, 2nd Ed, 2017, Cambridge.
3. Ronald Brachman and Hector Levesque. Knowledge Representation and Reasoning,2004, Morgan Kaufmann.

Course Name: Machine Learning	**Course Code: CSC-528**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Artificial Intelligence

Course Objective:

Machine learning is one of the fastest growing areas of computer science, with far-reachingapplications. The aim of this course is to:

- Present the basic machine learning concepts;
- Present a range of machine learning algorithms along with their strengths and weaknesses;
- Apply machine learning algorithms to solve problems of moderate complexity.

Course Outlines

Introduction to machine learning; concept learning: General-to-specific ordering of hypotheses, Version spaces Algorithm, Candidate elimination algorithm; Supervised Learning: decision trees, Naive Bayes, Artificial Neural Networks, Support Vector Machines, Overfitting, noisy data, and pruning, Measuring Classifier Accuracy; Linear andLogistic regression; Unsupervised Learning: Hierarchical Aglomerative Clustering. k-meanspartitional clustering; Self-Organizing Maps (SOM) k-Nearest-neighbor algorithm; Semi- supervised learning with EM using labeled and unlabled data; Reinforcement Learning: Hidden Markov models, Monte Carlo inference Exploration vs. Exploitation Trade-off, Markov Decision Processes; Ensemble Learning: Using committees of multiple hypotheses. Bagging, boosting.

Course Outcomes: After completion of the course students will be able to:

- Describe basic machine learning concepts, theories and applications.
- Apply supervised learning techniques to solve classification problems of moderate complexity.
- Apply unsupervised learning techniques to solve clustering problems of moderate complexity
- Apply reinforcement learning algorithms to environments with complex dynamics.
- Develop a reasonable size project using suitable machine learning technique

Recommended Books: Latest Edition of the Following Books

1. Tom, M, Machine Learning, 1997, McGraw Hill.
2. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, 2012, MIT Press.

Course Name: Artificial Neural Networks	**Course Code: CSC-656**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Programming for Artificial Intelligence

Course Objective:

- This course will introduce Artificial Neural Networks and Deep Learning. ANN's basic architecture and how they mimic the human brain using simple mathematical models. Manyof the important concepts and techniques around brain computing and the major types of ANN will also be introduced.
- Emphasis is made on the mathematical models, understanding learning laws, selecting activation functions and how to train the networks to solve classification problems. Deep neural networks have achieved state of the art performance onseveral computer vision and speech recognition benchmarks.
- This course will further build on the fundamentals of Neural networks and artificial intelligence and will introduceadvanced topics in neural networks, convolutional and recurrent network structures, deep unsupervised and reinforcement learning.

Course Outlines

Introduction and history of neural networks, Basic architecture of neural networks, Perceptron and Adaline (Minimum Error Learning) for classification. Basics of deep learning, learning networks, Shallow vs. Deep learning etc.; Machine learning theory – training and test sets, evaluation, etc. Selected topics from: Gradient descent (Delta) rule, Hebbian, Neo-Hebbian and Differential Hebbian Learning, Drive Reinforcement Theory, Kohonen Self Organizing Maps, Associative memory, Bi-directional associative memory (BAM), Energy surfaces, The Boltzmann machines, Backpropagation Networks, Feedforward Networks; Theory of Generalization; Multi-layer perceptrons, error back- propagation; Deep convolutional networks, Computational complexity of feed forward and deep convolutional neural networks; Unsupervised deep learning including auto-encoders; Deep belief networks; Restricted Boltzman Machines; Deep Recurrent Neural Networks (BPTT, LSTM, etc.); GPU programming for deep learning CuDNN; Generative adversarialnetworks (GANs); Sparse coding and auto-encoders; Data augmentation, elastic distortions,data normalization; Mitigating overfitting with dropout, batch normalization, dropconnect; Novel architectures, ResNet, GoogleNet, etc

Course Outcomes: After completion of the course students will be able to:

- Understand the fundamentals of neural networks in AI
- Explain how simple ANNs can be designed
- Apply ANN for classification Problems
- Apply deep learning algorithms to real-world problems
- Analyze results from deep learning to select appropriate solutions

Recommended Books: Latest Edition of the Following Books

1. Martin T. Hagan, Howard, B. Demuth, Mark Hudson Beale and Orlando De Jesus , Neural Network Design, 2nd Edition, 2014, Martin Hagan; 2 edition (September 1, 2014), ISBN-10: 0971732116
2. James A Anderson, An Introduction to Neural Networks1995, A Bradford Book , ISBN-10: 0262011441
3. Mohammad Hassoun, Fundamentals of Artificial Neural Networks, 2003, A Bradford Book, ISBN-10: 0262514672
4. Ian Goodfellow, Yoshua Bengio, Aaron Courville Deep Learning (http://www.deeplearningbook.org/)
5. Francoise Chollet , Deep learning with python, 2017, ISBN-10: 9781617294433, Manning,

Course Name: Digital Forensics	**Course Code: CSC-651**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Introduction to Cyber Security

Course Objective:

- This course is an introduction to computer forensics and investigation. It provides an understanding of how to conduct investigations to correctly gather, analyze and present digital evidence to different audiences.
- It also outlines the tools to locate and analyze digitalevidence on a variety of devices, how to keep up to date with changing technologies, and laws and regulations in digital forensics.

Course Outlines

Introduction to Cyber Security An introduction to Digital Forensics; use of digital forensics; Key technical concepts; Challenges in Digital Forensics ; The Difference between Computer Experts and Digital Forensics Experts; Investigative Process Methodologies ; Education, Training, and Awareness; Laws, Standards, and Regulations; Ethics and Professional Conduct; Digital Evidence Management; Collecting evidence; Antiforensics; Network forensics; Mobile andEmbedded Forensics; Cloud forensics; Internet Forensics; social media forensics;Investigation Methods for Collecting Digital Evidence; Digital Forensic Readiness; Digital forensics tools; Discovery of Computers and Storage Media; Discovery of Audio/ Video Evidence; Data Visualization; Data Sources; Graphing and Charting; Analyzing Data; DataDistributions; Analysis Scenarios; Data Visualization Tools.

Course Outcomes: After completion of the course students will be able to:

- To develop knowledge about forensic law, standards, regulations and ethical values
- To be able to conduct digital forensics for multiple platforms and applications by various tools
- To be able to generate reports based on digital forensic tools for security systems and platforms

Recommended Books: Latest Edition of the Following Books

1. John Sammons, The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics, 2012, Singress.
2. Gerard, Digital Forensics and Incident Response : Incident Response Techniques and Procedures to Respond to Modern Cyber Threats, 2020, Packt Publishing.
3. Oakim Kävrestad , Guide to Digital Forensics : A Concise and Practical Introduction, 2017, Springer.

Course Name: Network Security	**Course Code: CSC-650**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Introduction to Cyber Security	
Course Objective: • The module aims to develop core competencies in the fields of Network security and offer the opportunity of learning the current network security landscape, understanding current threats and vulnerabilities and examining ways of developing effective countermeasures. • It also provides a brief overview to network forensics for analyzing network traffic for the purposes of information gathering, legal evidence, or intrusion detection.	
Course Outlines Introduction to network security, Networking Concepts and Protocols, Network Threats andVulnerabilities, Network Security Planning and Policy, Access Control, Defense against Network Attacks, DOS and DDOS detection and prevention, Firewalls, Intrusion Detectionand Prevention Systems, Antivirus Filtering, Naming and DNS Security, DNSSEC, IP security, Secure Sockets Layer, VPN, Packet Sniffing and spoofing, Honeypot, Ethernet Security, Wireless Security, Wireless Attacks, Wireless LAN Security with 802.11i, Wireless Security Protocols, Wireless Intrusion Detection, Physical access and Security, TorNetwork, Network Forensics. Defense against Network Attacks	
Course Outcomes: After completion of the course students will be able to: • To be able to understand network security threats and methods for security networks • To be able to secure wired networks by deploying various methods • To be able to secure wireless networks by deploying various methods	
Recommended Books: Latest Edition of the Following Books 1. Chris McNab, Network Security Assessment: Know Your Network,2017, O'Reilly Media. 2. Randall J. Boyle, Corporate Computer Security, 2014,Pearson. 3. Praphul Chandra, Bulletproof Wireless Security, 2005,Newnes. 4. William Stallings,Network Security Essentials: Applications and Standards, 2016,Pearson. 5 William Stallings, Cryptography and Network Security Principles and Practices,2016,Pearson.	

Course Name: Secure Software Design and Development	**Course Code: CSC-652**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Introduction to Cyber Security	
Course Objective: The module aims to develop core competencies in the fields of Secure Software Concepts, Secure Software Requirements, Secure Software Design, Secure SoftwareImplementation/Coding, and Secure Software Testing. The course details the software security activities that need to be incorporated throughout the software development lifecycle. It provides comprehensive coverage that includes the people, processes, and technology components of software, networks, and host defenses.	
Course Outlines Secure software concepts; System issues; System properties; Software Project Time Management; Software Project Costing; Software Quality Assurance; Security Concepts inthe SDLC; Risk management; Security standards (e.g., coding standards, NIST standards, Federal Information Processing Standards); Best practices (e.g., OWASP development guide, OWASP code review guide, OWASP testing guide); Security methodologies (e.g., Socratic Methodology, Operationally Critical Threat, Asset, and Vulnerability Evaluation, STRIDE and DREAD, Open Source Security Testing Methodology Manual); Security frameworks (e.g., Zachman	

Framework, Control Objectives for Information and Related Technology, Sherwood Applied Business Security Architecture (SABSA)); Regulations- Privacy and Compliance; Security Models (e.g., BLP Confidentiality Model, Clark and Wilson Model (Access Triple Model)); Trusted Computing; Secure Software Requirements (Sources for Security Requirements, Types of Security Requirements); Secure Software Design (Design consideration, Information Technology Security Principles and Secure Design, Designing Secure Design Principles); Design Processes; Secure Software Implementation/Coding; Software Development Methodologies; Common Software Vulnerabilities and Controls; Defensive Coding Practices—Concepts and Techniques; Code Vulnerabilities and Avoiding Polymorphic Malware Attacks: Buffer overflow, Format string bug, Code vulnerabilities SQL Injection, Cross-site Scripting, Cross-site Request Forgery, Session management, Replication of vulnerabilities and exploitation; Secure Software Testing; Security Testing Methodologies; Software Security Testing; Software Acceptance; Legal Protection Mechanisms; Software Deployment- Operations- Maintenance and Disposal.

Course Outcomes: After completion of the course students will be able to:

- A good comprehension of software security standards, models, processes and best practices that need to be incorporated throughout the software development lifecycle.
- Identify insecure programming patterns and the ability to replace them with secure alternatives.
- Know tools for software security analysis and testing, and the ability to use them in practice and understand their capabilities and limitations

Recommended Books: Latest Edition of the Following Books

1. Mano Paul, Official (ISC)2 Guide to the CSSLP , 2013, Auerbach.
2. Gary McGraw , Software Security: Building Security, 2006,Addison-Wesley.

Course Name: Computing Vision	**Course Code: CSC-622**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Artificial Neural Network & Deep Learning

Course Objective:

- The course focus on field of computer vision in general for different applications, etc.

Course Outlines

Introduction to Computer Vision (Problems faced, History and Modern Advancements). Image Processing, Image filtering, Image pyramids and Fourier transform, Hough transform.Camera models, Setting up a camera model from parameters, Camera looking at a plane, Relationship of plane and horizon line, Rotation about camera center. Concatenation,Decomposition and Estimation of transformation from point correspondences, Points and planes in 2D/3D, Transformations in 2D/3D, Rotations in 2D/3D. Edge detection, corner detection. Feature descriptors and matching (HoG features, SIFT, SURF). Applications of Computer Vision Traditional Methods: Image Stitching: Making a bigger picture from smaller pictures Single View Geometry: Converting a single image into a 3D model. Applications of CV using Deep Learning: Image Detection (Localization, Historical Techniques, RCNN, FRCNN, YOLO, Retina), Image Segmentation (UNet, SegNet, MaskRCNN), Image Generation (GANN)

Course Outcomes: After completion of the course students will be able to:

- Understand and explain the field of computer vision in general for different applications, etc.
- Understand and implement camera calibration 3. Work under OpenCV or Matlab computer vision toolbox, etc.
- Implement an algorithm to assemble the extracted features to develop a higher-level perception
- Implement different algorithms for spatial and frequency domain filtering, feature detection, structure from motion, motion estimation, etc.
- To detect, recognize and track different types of the objects in the scene
- Develop an algorithm for context awareness or scene understanding

Recommended Books: Latest Edition of the Following Books

1. Richard Szeliski. Computer Vision: Algorithms and Applications, 2010,Springer.
2. Richard Hartley and AndrewZisserman, Multiple View Geometry in Computer Vision, 2004, Cambridge University Press.
3. David Forsyth and Jean Ponce, Computer Vision: A Modern Approach, 2011,Pearson.
4. Rafael Gonzalez and Richard Woods, Digital Image Processing,2017, Pearson.

Course Name: Professional Practices	**Course Code: CSC-530**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: None

Course Objective:

- The course aim to equip students with computing ethics
- To learn Software code of ethics

Course Outlines

Computing Profession, Computing Ethics, Philosophy of Ethics. The Structure of Organizations, Finance and

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

Course Outcomes: After completion of the course students will be able to:

- Understand computing ethics
- How to implement code of ethics

Recommended Books: Latest Edition of the Following Books

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

Course Name: Vulnerability Assessment & Reverse Engineering	**Course Code: CSC-653**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Introduction to Cyber Security	
Course Objective: • The course aims to develop core competencies in the field of vulnerability assessment covering software, networks and Web applications. • It also covers reverse engineering techniques to analyze software, exploit targets, and defend against security threats like malware and viruses.	
Course Outlines Understanding the need for security assessments; Classifying vulnerabilities; Software vulnerabilities; Network vulnerabilities; Vulnerability assessment versus penetration testing; Vulnerability Assessment Tools; Vulnerability management Regulatory compliance; Calculating ROIs; Application review process; Pre-assessment; Code navigation; Code- auditing tactics; Memory corruption; understanding issues in programming languages; Stepsin Reverse engineering, Common tools used for Reverse engineering; Binary Obfuscation techniques; Understanding core assembly concepts to perform malicious code analysis, Identifying key assembly logic structures with a disassembler, Malware analysis Types of malware analysis; Malware Taxonomy; Static analysis; Dynamic analysis; Malware Inspection; Malware analysis tools; Sandboxing and virtualization;	
Course Outcomes: After completion of the course students will be able to: • Basic Understanding of Hacking and Ethical Hacking. • Apply techniques for vulnerability assessment and penetration testing. • Understand Software vulnerabilities, Network vulnerabilities, Types of Malware and its Analysis.	
Recommended Books: Latest Edition of the Following Books 1. Philip S. Anton Finding and Fixing Vulnerabilities in Information Systems: The Vulnerability Assessment and Mitigation Methodology,2004, RAND Cooperation. 2. Mark Dowd, The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities,2006,Addison-Wesley.	

Course Name: Introduction to Cyber Security	**Course Code: CSC-534**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Information Security	
Course Objective: • This course provides students an introduction to common cyber security threats, vulnerabilities, and risks related to web applications, networks, software and mobile applications. • The course provides basic concepts and terminology used in the information and cyber security fields. Moreover, it will also enable students to differentiate between thevarious forms of malware and how they affect computers and networks.	
Course Outlines Introduction to Cyber security; Networks and the Internet; cyber threat landscape; understanding security; information security Principles (Confidentiality, Integrity, Availability); Information Security Terminology; Who are the attackers; Advanced Persistent Threat (APT); Malware, types of malware; Attacks using malware; Malware Attack Lifecycle: Stages of Attack; Social engineering attacks; types of payload; Industrial Espionage in Cyberspace; Basic cryptography; Web application attacks; Database security;Cyber kill chain; Privacy and anonymity; Network security; Software security; Mobile device security; Mobile app security; Cyber Terrorism and Information Warfare; Introduction to Digital Forensics; Digital Forensics Categories.	
Course Outcomes: After completion of the course students will be able to: • To be able to identify computer system threats	

- To be able to identify Malware attacks, and understand the stages of attack and payloads.
- Implement various cryptographic techniques and simulate attack scenarios

Recommended Books: Latest Edition of the Following Books

1. Chuck Easttom, Computer Security Fundamentals, 2019,Pearson.
2. Mark Ciampa, Security+ Guide to Network Security Fundamentals, 2017,Cenegae Learning.
3. C.P. Pfleeger, Security in Computing, 2015, Prentice-Hall.

Course Name: Digital Image Processing	**Course Code: CSC-632**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites:

Course Objective:

- To learn digital image processing concepts and applications.

Course Outlines

The human visual system, electromagnetic system, working and components inside digitalcamera, pixels, image representation, sampling, quantization, mathematics of imageformation, convolution, camera projection, point-based image processing, Fourier theory,image filtering in spatial and frequency domain, wavelets, image registration, morphological operations, color models, multispectral images, feature detection, image segmentation, Pattern recognition, etc.

Course Outcomes: After completion of the course students will be able to:

- Understand the basics, applications in general, working inside the digital camera, sampling and quantization, image representation, etc.
- Implement image enhancement, image segmentation, image transformations, spatial and frequency domain processing, filtering, convolution, image registration, feature detection,
- pattern recognition, etc.
- Evaluate the performance of different image processing algorithms.

Recommended Books: Latest Edition of the Following Books

1. Gonzalez R. C., Woods R. E., Eddins S. L., Digital Image Processing Using Matlab,2009,Pearson Education.
2. Gonzalez R. C., Woods R. E., Digital Image Processing, 2008, Pearson Education.
3. Richard G. Lyons, Understanding Digital Signal Processing, 2010, Prentice Hall.

Course Name: Object Oriented Analysis and Design	**Course Code: CSC-440**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Programming Fundamentals	
Course Objective: • The course focuses on learning and implementation of Object oriented analysis and design.	
Course Outlines Principles of Object Technology. OOP Review. Principles of Modeling. OOA&D Overview. OO Development Process. Requirements Engineering, Analysis, and Specification: Requirements Engineering, Use Cases, Prototyping, Class Models. Interaction Diagrams. Verification and Validation. Architectural and Detailed Design. Class Diagrams. Interaction Diagrams. State Machines and Diagrams. Implementation, Package Diagrams. Activity Diagrams. OO Patterns, Verification and Validation. Note: Students may also be introduced to Object Diagram, Component Diagram, Package Diagram, Deployment Diagram, Network Diagram.	
Course Outcomes: After completion of the course students will be able to: • Understand OOA and Design • Develop activity and component diagram.	
Recommended Books: Latest Edition of the Following Books 1. Craig Larman, Applying UML and patterns: An introduction to Object-Oriented Analysis andDesign and Iterative Development, 2004, Prentice Hall, ISBN-10: 0131489062 2. Perdita Stevens, Using UML: Software Engineering with Objects and Components,2006, Addison-Wesley, ISBN-10: 0321269675 3. Meiler Page-Jones,Fundamental of Object-Oriented Design in UM, 2000, AddisonWesley, ISBN: 020169946X. 4. G. Booch, J. Rambaugh and I, The Unified Modeling Language User Guide, 2005, Jakobson, Addison-Wesley Professional, ISBN- 10:0321267974.	

Course Name: Software Re-Engineering	**Course Code: CSC-620**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**
Prerequisites: Software Construction and Development	
Course Objective: • To learn concept of reengineering techniques	
Course Outlines Salient topics include the terminology and the processes pertaining to software evolution,fundamental re-engineering techniques to modernize legacy systems including source code analysis, architecture recovery, and code restructuring, software refactoring strategies, migration to Object Oriented platforms, quality issues in re-engineering processes, migration to network-centric environments, and software integration, reverse engineering, program comprehension, source code transformation and refactoring strategies, software maintenance and re-engineering economics	
Course Outcomes: After completion of the course students will be able to: • Explain the concepts and technique of software re-engineering. • Apply reengineering techniques to maintain and modify • software systems • Analyze and understand maintenance related problems associated with object oriented software systems. • Able to perform complex design reengineering and reverse engineering problems.	

Recommended Books: Latest Edition of the Following Books

1. David Lorge Parnas, Chris Birchall, Safari Books,Shelter Island, NY, Re-engineering legacy software, 2016, Manning Publications.
2. Priyadarshi Tripathy and Kshirasagar Naik, Reengineering, 2015, John Wiley & Sons,Inc.
3. K.H.Bennett and V.T Rajlich, Software Maintenance and Evolution: a Roadmap, 2006, Pearson.
4. Sabastian, TheFuture of Software Engineering, 2010, Springer.

Course Name: Computer Graphics	**Course Code: CSC-525**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Introduction to Cyber Security

Course Objective:

- The course focuses on learning of structure of modern computer graphics systems

Course Outlines

Fundamental Concepts: forward and backward rendering (i.e., ray-casting and rasterization), applications of computer graphics: including game engines, cad, visualization, virtual reality, polygonal representation, basic radiometry, similar triangles, and projection model, use of standard graphics APIs (see HCI GUI construction); basic rendering: rendering in nature, i.e., the emission and scattering of light and its relation to numerical integration, affine and coordinate system transformations, ray tracing, visibility and occlusion, including solutions to this problem such as depth buffering, painter's algorithm, and ray tracing, the forward and backward rendering equation, simple triangle rasterization, rendering with a shader-based API, texture mapping, including minification and magnification (e.g., trilinear MIP-mapping),application of spatial data structures to rendering, sampling and anti-aliasing, scene graphs and the graphics pipeline; geometric modeling: basic geometric operations such as intersection calculation, proximity tests, polynomial curves and surfaces, Approximation techniques such as polynomial curves, bezier curves, spline curves and surfaces, animation as a sequence of still images.

Course Outcomes: After completion of the course students will be able to:

- Comprehend the structure of modern computer graphics systems
- Explain the basic principles of implementing computer graphics fundamentals
- Compare key algorithms for modelling and rendering graphical data
- Develop design and problem solving skills with applications to computer graphics
- Construct interactive computer graphics programs using OpenGL

Recommended Books: Latest Edition of the Following Books

1. Donald D. Hearn, Computer Graphics with Open G, 4th Edition, 2010, Prentice Hall, ISBN-10: 0136053580.
2. S. J. Gortler, Foundations of 3D Computer Graphics, 2012, The MIT press.
3. A K Peters, Fundamentals of Computer Graphics, 2009, Pearson.
4. John Hughes, Computer Graphics: Principles and Practice, 3rd Edition, 2013, Addison Wesley.
5. Tomas Akenine, Real-Time Rendering, 3rd Edition, 2018, A K Peters.

Course Name: Visual Programming	**Course Code: CSC-531**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Object Oriented Programming

Course Objective:

- To learn visual and event driven programming.

Course Outlines

Visual Programming Basics; Introduction to Events; Fundamentals of Event-driven Programming, message

handling, user interfaces, graphics device interface, painting and drawing, windows management, input devices, resources, string and menu resource, dialogs and windows controls, common controls, dynamic link libraries, threads and synchronization, network programming, Building Class Libraries at the Command Line, Class Libraries, Using References, Assemblies, Private Assembly Deployment, Shared Assembly Deployment, Configuration Overview, Configuration Files, Programmatic Access to Configuration, Using SDK Tools for Signing and Deployment, Metadata, Reflection, Late Binding, Directories, Files, Serialization, Attributes, Memory Management and Garbage Collection, Threading and Synchronization, Asynchronous Delegates, Application Domains, Marshal by Value, Marshal by Reference, Authentication and Authorization, Configuring Security, Code Access Security, Code Groups, Evidence, Permissions, Role-Based Security, Principals and Identities, Using Data Readers, Using Data Sets, Interacting with XML Data, Tracing Event Logs, Using the Boolean Switch and Trace Switch Classes, Print Debugging Information with the Debug Class, Instrumenting Release Builds with the Trace Class, Using Listeners, and
Implementing Custom Listeners.

Course Outcomes: After completion of the course students will be able to:

- Use the different elements of a visual programming language as building blocks to develop correct, coherent programs.
- Program using the fundamental software development process, including design, coding, documentation, testing, and debugging.
- Analyze problems, develop conceptual designs that solve those problems, and transform those designs to Visual Programs.

Recommended Books: Latest Edition of the Following Books

1. Deitel and Deitel, Visual C#: How to Program, 6/e Edition,2017, Prentice Hall / PearsonEducation.
2. J.C. Bradley, A.C. Millspaugh, Programming in C# .NET, 2014, McGraw-Hill.
3. Sharp, J, Microsoft Visual C# 2013 Step by Step (Step by Step Developer), 2013, Microsoft Press.

Course Name: Programming for Artificial Intelligence	**Course Code: CSC-539**
Course Structure: Lectures: 2, Labs: 1	**Credit Hours: 3**

Prerequisites: Artificial Intelligence

Course Objective:

- This course aims to introduce standard programming practices and to help develop programming skills necessary for designing and implementing Artificial Intelligence systems.
- The course introduces a modern state of the art programming language for ArtificialIntelligence, and builds up the necessary programming background for the main courses likeKnowledge Representation & Reasoning, Machine Learning, Artificial Neural Networks, and Natural Language Processing.
- This course will help the students of Artificial Intelligencedevelop the programming acumen and style. The ultimate aim of this course is to help students in using the programming language to solve problems of interest to them.

Introduction to Programming language (Python): The first objective of the course is to introduce and then build the proficiency of students in the programming language. The basicsinclude IDE for the language (e.g., Jupyter Notebook or IPython), variables, expressions, operands and operators, loops, control structures, debugging, error messages, functions, strings, lists, object-oriented constructs and basic graphics in the language. Special emphasis is given to writing production quality clean code in the programming language using versioncontrol (git and subversion).

Introducing libraries/toolboxes necessary for data analysis: The course should introducesome libraries necessary for interpreting, analyzing and plotting numerical data (e.g.,NumPy, MatPlotLib, Anaconda and Pandas for Python) and give examples of each library using simple use cases and small case studies.

Course Outcomes: After completion of the course students will be able to:

- Comprehend the fundamental constructs of programming language for data analysis and representation.
- Understand and apply the Object-oriented concepts in the programming language

- Solve and analyze programming and data analysis problems using standard libraries and/or toolboxes of the programming language.

Recommended Books: Latest Edition of the Following Books

1. Severance, C.R, Python for everybody: Exploring data using Python 3, 2016, CreateSpace Independent Publ Platform.
2. Miller, B.N., Ranum, D.L. and Anderson, J, Python programming in context, 2019, Jones & Bartlett Pub.
3. McKinney, W, Python for data analysis: Data wrangling with Pandas, NumPy, and IPython. 2012, O'Reilly Media, Inc.
4. Joshi, P, Artificial intelligence with python, 2017, Packt Publishing Ltd.
5. Janert, P.K, Data analysis with open source tools: a hands-on guide for programmers and data scientists, 2010, O'Reilly Media, Inc.

Course Name: Natural Language Processing	**Course Code: CSC-649**
Course Structure: Lectures: 3, Labs: 0	**Credit Hours: 3**

Prerequisites: Artificial Neural Network & Deep Learning

Course Objective:

- Natural Language Processing (NLP) is the application of computational techniques to the analysis and synthesis of natural language and speech.
- This course is an introduction to NLPwith prior programming experience in Python.

Course Outlines:

Introduction & History of NLP, Parsing algorithms, Basic Text Processing, Minimum Edit Distance, Language Modeling, Spelling Correction, Text Classification, Deterministic and stochastic grammars, CFGs, Representing meaning /Semantics, Semantic roles, Semantics and Vector models, Sentiment Analysis, Temporal representations, Corpus-based methods, N-grams and HMMs, Smoothing and backoff, POS tagging and morphology, Information retrieval, Vector space model, Precision and recall, Information extraction, Relation Extraction (dependency, constituency grammar), Language translation, Text classification, categorization, Bag of words model, Question and Answering, Text Summarization

Course Outcomes: After completion of the course students will be able to:

- Understand techniques for information retrieval, language translation, and text classification.
- Understand the advantages of using standard corpora. Identify examples of current corpora for a variety of NLP tasks.
- Understand and contrast deterministic and stochastic grammars, providing examples to show the adequacy of each Solve classic and stochastic algorithms for parsing natural language.

Recommended Books: Latest Edition of the Following Books

1. Daniel Jurafsky and James H. Martin, Speech and Language Processing: AnIntroduction to Natural Language Processing, 2018, Prentice Hall.
2. Manning and Schütze, Foundations of Statistical Natural Language Processing,1999, MITPress. Cambridge.

ANNEXTURE A

Approved Computing Courses to be offered for other departments 2023-onwards.

The details of computing courses that are required to teach in the department of Bioinformatics as given below. The BOS and BOF members recommended and approved the below-mentioned courses to be taught in other departments.

1. Graphics and Visualization
2. Database Management System
3. Modeling and Simulation

Course Name: Graphics and Visualization	**Course Code: CSC-526**
Course Structure: Lectures: 3, Labs: 1	**Credit Hours:** 4
Prerequisites: Programming Fundamentals	
Course Objective • This course is designed to provide a comprehensive introduction to computer graphics, learn concepts of computer graphics and its algorithms. • The aim is to learn concepts of illumination, animation, shading and geometrical transformations. This will lead to the ability to understand contemporary terminology, progress, issues, and trends in computer graphics. **Course Outcomes:** After completion of the course students will be able to: • Have an understanding of the concepts of computer graphics • Be able to implement the concepts of computer graphics on a suitable platform	
Course Outline: Introduction, Applications of Computer Graphics, Overview of Image Representation, Graphics Hardware, Applications of Graphics, Image resolution, Image quality issues, Cathode Ray Tubes, Vector Display Devices, Colour Lookup Table, LCD Technology, Plasama, classes of logical input devices, Physical input devices, Interactive Devices,Scan-converting point, lines, ellipse, arcs and sectors, polygon, characters and circles,Line Drawing Algorithm: Digital Differential Analyzer, Bresenham's, Circle Drawing: mid-point, Line Algorithms, Scaling, Rotation, Translation, composite transformations: 2D and 3D,Region Filling, aliasing effects, anti-aliasing, image compression, Window to Viewport mapping, 2-D Clipping, point clipping, line clipping, polygon clipping, Panning and Zooming, Projections: Taxonomy of projection, perspective projection orthogonal projection, Advanced geometric and Raster Algorithms, Clipping scan converting primitives, special problems with text, making copy Pixel fast, Page description languages, Hidden Surfaces, depth comparisons, Z-buffer algorithm, Back-Face Removal, Scan-line algorithm, Hidden Line Elimination, Rendering of mathematical surfaces, Curve and Surface Design, simple geometric forms, wireframe models, curved surfaces, curve design, polynomial basis function, curved-surface design, Rendering, Shading, Phong Model, Colour and Animation, basic rules of animation, Ray tracing, the Pinhole Camera, Ray-surface intersection, A recursive ray tracer, Hardware for interactive graphics: fundamental graphics unit, rasterizer, pixel driver, Segmentation, two and three dimensional image geometry and transformation	
Lab Outline: Line drawing techniques, clipping effects, 2D and 3D representations and transformations using open GL, development of graphical user interface with various blocks and modules, elliptical and curve creation exercises.	
Recommended Books: Latest Edition of the Following Books. 1. Foley, J.D., Dam,A.V., Feiner,S.K., Hughes,J.F. (Latest edition) Computer Graphics, Principles and Practice. Addison-Wesley . 2. Hill,. F.S.(Latest edition) Computer Graphics.MacMillan.	

3. Burger,P., Gillies,D.F. (Latest edition). Interactive Computer Graphics: Functional, Procedural and Device-level methods. Addison-Wesley.
4.Computer Graphics (C Version), by Donald Hearn and M. Pauline Baker (Prentice Hall, 1997)
5.Graphics & Visualization Principles and Algorithms by Theoharis, Georgios Papaioannou.
6.Matthew Ward,Georges Grinstein, Daniel Keim. Interactive Data Visualization: Foundations, Techniques, and Applications. (May, 2010).

Course Name: Database Management Systems	**Course Code: CSC-524**
Course Structure: Lectures: 3, Labs: 1	**Credit Hours:** 4

Prerequisites: None

Course Objective:

- The course aims to introduce basic database concepts, different data models, data storage and retrieval techniques and database design techniques.
- The course primarily focuses on relational data model and DBMS concepts.

Course Outcomes:

- After completing the course, the students will be familiar with data modelling concepts used in DB development.
- Undertake and successfully complete logical data base design tasks.
- Be familiar with a broad range of data management issues, including data integrity, concurrency and security.

Course Outline:

Basic database concepts; Entity Relationship modeling, Relational data model and algebra,Structured Query language; RDBMS; Database design, functional dependencies and normal forms;query optimization concepts,Transaction processing and optimization concepts; concurrency control and recovery techniques; Database security and authorization. Small Group Project implementing a database. Physical database design: Storage and file structure; indexed files; b-trees; files with dense index; files with variable length records; Database efficiency and tuning.

Lab Outline:

Structured Query Language commands, creating and populating tables, design of simple databases, database normalization techniques, query optimization, indexing techniques, partial and full recovery techniques, developing GUI techniques, implementation of database security mechanisms.

Recommended Books: Latest Edition of the Following Books.

1. Jeffrey A. Hoffer, V. Ramesh, Heikki Topi. Modern Database Managment 11th Edition, 2012, Prentice Hall.
2. Connolly,R., Begg,P. Database Systems: A Practical Approach to Design, Implementation and Management, 5th Edition, 2009, Addison-Wesley Pub. Co.
3. Ramez Elmasri and Shamkant B. Navathe. Fundamentals of Database Systems. 6th Edition, 2010, Pearson.
4. C.J.Date,. An Introduction to Database System, 8th Edition, 2004, Addison-Wesley.

Course Name: Modeling & Simulation	**Course Code: CSC-637**
Course Structure: Lectures: 2, Lab: 1	**Credit Hours:** 3

Prerequisites: None

Course Objective

- This course emphasizes the development of modeling and simulation concepts and analysis skills necessary to design, program, implement, and use computers.

Course Outline

Baics, Performance modeling and evaluation, bench marking, performance evaluation of high parallel systems architecture, application of performance evaluation, measurement techniques, hardware monitoring, software monitoring, hybrid monitoring, fundamentals of queuing models, structure and performance parameters, operational analysis of queuing models, general features of queuing models, birth and death processes, m/m/i and m/g/1 systems, dependability modeling, analysis of reliable, available and 30 high assurance systems, fault-tolerant techniques, software reliability modeling, adaptive modeling, agent based modeling.

Lab Outline

Introduction to modeling techniques using simulation tools like MATLAB. Bioinformatics toolbox for various performance modeling,evaluation, analysis and study various queuing techniques.

Recommended Books:

1. I. Mitran, Simulation Techniques for Discrete Event Systems,Cambridge Computer Science text 14, Published by University Press, Cambridge,1982.
2. Banks,Jerry, Discrete Event system simulation, Pearson Prentice Hall, 2009
3. Law,A.M,Kelton and W.David Simulation modeling and analysis, Mcgraw Hill,2000.