Curriculum

Of

Bachelor of Science in Computer Science

BS (CS)

Revised: Spring 2019

Department of Computer Science,

School of Systems and Technology, University of Management and Technology,

Lahore, Pakistan

Table of Contents

1. Curriculum Review Committee

1. Dr. Yaser Daanial Khan
2. Dr. Adnan Abid
3. Dr. Shoaib Farooq
4. Dr. Shahid Awan
5. Nabeel Sabir Khan
6. Tahreem Yasir

2. Preface

The curriculum has been designed keeping in view the recommendation of HEC and the emerging needs of the industry. An effort has been made to incorporate state of art technologies and tools in the field of computer science which can be most beneficial to students. Also a balance has been kept in the amount of theoretical knowledge and its applications.

3. Acknowledgment (To the Rector for his vision, and Faculty members for their support)

We acknowledge the honorable rector of UMT for his vision and motivation to make our academic programs up-to-date by including state-of-the-art courses, materials and practices. We also acknowledge the support of our faculty members for sharing their experiences and pieces of advice to improve the curriculum.

4. Program Objectives

Curriculum has sufficient technical content to prepare students for industry and graduate studies, and has adequate humanities and social sciences content to help students become well-rounded individuals. It has been designed to train and develop skill of the students enabling them to achieve the following goals

- Effective Communication in career and educational environments
- A strong foundation in computing and mathematical theoretical concepts and applying these concepts to problems requiring computer solutions.
- Proficiency in the use of system design notations and system design engineering process to develop software systems and solving+ problems in computer science
- Apply their computer science knowledge and skills to develop a career in an information technology oriented business or industry, or for graduate study in computer science or other scientific or technical fields

Curriculum meets the minimum requirements recommended in the HEC computing curriculum recommendations, which have the following area-wise breakdown. Curriculum coverage is well-planned over four years.

5. Effectiveness of the Program

At all the departments of School of Systems and Technology (SST) are following state of the art curriculums for their respective programs, which have recently been revised in the light of latest Program Roadmap as recommended by HEC of Pakistan. The outlines of all the foundation, core or elective courses in each program has been designed using the latest research, methodologies, practices, skill set and technologies. The outlines of foundation level undergraduate courses have been structured in a way so as to introduce students with the essentials foundations of a particular area. The roadmaps of program offer courses through a closely knit relationship among course where pre-requisites are to be studied and passed first before commencing an advanced level course. For each program, roadmaps also focus on courses which are related to the respective program majors and the offering these courses follow each other.

The program wise foundation and core courses have also been supplemented with respective labs in which students practice and apply the methodologies and skills learnt in the theory component of the respective course. In these lab sessions, student practice their skills on latest toolset and technologies available. At SST there are 4 computer and 1 digital system design classroom and 1 general purpose lab where state of the art tools have already been installed to assist the students in improving their skills.

Technology Adoption:

In addition each classroom is equipped with power point projectors and classroom PC's also contain the tools and platforms where during the lecture instructor can also demonstrate the real examples to the students. At SST, faculty uses a combination of projector and white board based instruction to teach the courses whereas to communicate with students a customized version of Google's Moodle has been used. A part from lab component, quiz or exams students are also graded through programming projects and assignments which further improve their analysis, design and programming skills. To maintain the quality and content in each section of a particular course, course moderator activity has been implemented closely overlook the progress of content in each section and report the same to the respective head.

Emerging Development Trends:

To prepare the students for their future endeavors and to keep up with the latest research and technological trends, a number of elective courses are offered to Junior and Senior batch students which range from Big Data Programming, Cloud Computing and Advanced Web Design to Natural Language Processing, Bioinformatics and Smart Environments to Open Source Software Development, Digital Image Processing, Data Mining and Machine Learning.

In addition, to further introduce the students and faculty with current industry and research practices several seminars and workshops are regularly arranged by the faculty where distinguished field experts and professionals are invited to discuss about the existing trend, tools and methodologies. Till date various seminars and workshops ranging from Machine Learning, AWS Application Development, DevOps, Cloud Computing, Design of IOT systems, Responsive Web Design and Cyber Security has been conducted.

Industry Linkages

"Bridging the gap between the Academia and the Corporate World"

Introduction:

UMT has established Industry Linkage office to promote the activities of SST and develop a strong link with industry. Office of Industry Linkages has played its role in collaborating with various private and government sector organizations to meet one of its prime objectives i.e. building up a strong university-industry linkage. Experimental learning through workshops, industry visits, industry collaborations and inspirational lectures are the key aspects of industrial linkage activities. Industry linkage office aims to bridge the gap between academia and the corporate world. A formal relationship between academia and industry is bringing the practical aspect of the industry to SST.

Industrial Projects:

The School of Systems and Technology (SST) has active collaboration with different national and multinational software development companies. Different teams of students and faculty members are actively engaged in offshore software development and research activities. This also helps in confidence building, internships and future recruitment, and student induction in these companies. The major areas of collaboration are software development, web application development, Mobile phone applications, data science, robotics, IOT, and machine learning projects.

Industrial Partnerships:

University of Management and Technology (UMT) as part of its commitment to foster a generation of responsible youth have started a program for ensuring a meaningful and consistent involvement of its students in community services. A long term partnership with Akhuwat Foundation with a vision to channelize the positive energies of students into a sustainable model for volunteerism, community services, and socially responsible youth leadership, has been established. Similarly, a number of other partnerships are established with industrial and society partners.

Industrial Visits:

The SST organizes time to time visits of students and faculty members to partner industrial organizations to foster learning and share ideas. Frequent visits to Pakistan Telecommunication Corporation Limited and Punjab Safe City have been organized.

Industrial Talks:

To keep students updated regarding the latest industrial trends and software development framework. A number of technical talks are arranged during the running semesters. The distinguished speakers from industry are invited for interactive and hand-on sessions.

University Collaborations:

UMT has collaborated with North Eastern Illinois University (NEIU) for initiating a student exchange program. MOU has been signed between UMT and NEIU. Students from both universities will be able to benefit from this program. Student exchange is an investment for life. This experience can be beneficial in many different and valuable ways e.g. Personal, Academic and Employment. There are several research collaboration formed with many national and international universities both at faculty and school levels.

Industry Collaborated Events:

Several events are organized time to time in collaboration with software industry. Many times such events are organized on request of industrial partners. Some recent events are listed below.

- First AI Meetup (March 2018)
- First Free Lancers Meetup (August 2018)

Modelling and Design

The problem solving and solution modelling and design skills are developed through different core and elective courses offered to undergraduate students. The core competency of Object Oriented Programming and Software Design and Modelling comes through following courses.

Software Engineering: This course covers the modeling and design of following components. Modeling with UML, Requirements Elicitation, Analyses, System Design, Decomposing the System, Addressing Design Goals, Object Design, Reusing Pattern Solutions, Specifying Interfaces, Mapping Models to Code, Testing, Rationale Management.

The modelling of behavioral goals, quality goals, use cases, goal modelling heuristics are covered in software requirement engineering course. Further object modelling for requirements engineering, object modelling notations, object modelling heuristics, Identifying objects from goals. In Software Architecture and Design course, Prototyping and Use Case modelling is taught. Whereas, Generic design processes and design management are also covered in preceding course. The other skills of Software Verification and Validation, Quality Assurance in Software Projects, Quality Management, Quality Assurance and Standards, Quality Planning and Quality Control, Software Project Management, Scheduling, Project Monitoring and Control are taught in relevant courses.

All of the aforementioned software design and modeling skills are aligned and relevant to the local and international software industry. The contents of above mentioned courses are updated as per industry requirements.

6. Program Structure

Course Group	Credit Hours	% age
General Education	21	15%
University Electives	12	9%
Mathematics & Science Foundation	12	9%
Computing – Core	39	30%
Common Courses	**84**	**63%**

CS Core	24	18%
CS Electives	15	12%
CS Supporting	9	7%
Domain Courses	**48**	**37%**
Total	**132**	**100%**

Computing Courses	
Core	1. Programming Fundamentals 2. Object Oriented Programming 3. Data Structures and Algorithms 4. Discrete Structures 5. Operating Systems 6. Database Systems 7. Software Engineering 8. Computer Networks 9. Information Security 10. FYP 1 11. FYP 2
Courses	11
Credit Hours	39
Supporting Areas	1. Calculus 1 2. Probability and Statistics 3. Linear Algebra 4. Applied Physics
Courses	4
Credit Hours	12
Computer Science Courses	
Core	1. Compiler Construction 2. COAL 3. Digital Logic Design 4. Analysis of Algorithms 5. Parallel and Distributed Computing 6. Artificial Intelligence 7. Theory of Automata
Courses	7
Credit Hours	24

Supporting	**Any 3 from following list** 1. Differential Equations 2. Calculus 2 3. Graph Theory 4. Theory of Programming Languages 5. Numerical Computing (Numerical Analysis)
Courses	3
Credit Hours	9
General Education Courses	
1. English 1 (English Composition & Comprehension) 2. English 2 (Technical & Business Writing) 3. English 3 (Communication & Presentation Skills) 4. Professional Practices 5. Introduction to Info. & Communication Technologies 6. Pakistan Studies 7. Islamic Studies	
Courses	7
Credit Hrs	21
University Elective Courses	
	1. Foreign Language 2. Social Service 3. Management Related 4. Social Science Related 5. Economy Related
Courses	4
Credit Hrs	12

Technical Elective Courses	
	● Advanced Networks ● Advanced Web Technologies ● Big Data Programming ● Bioinformatics ● Block Chain Technology and Applications ● Cloud Computing ● Cognitive Radio Communication and Networks ● Component Based Software Engineering ● Computer Animations ● Computer Graphics ● Computer Vision ● Cyber Forensics ● Data Communications ● Data Mining ● Data Science & Big Data Analytics ● Data Science Technologies ● Data Warehousing ● Database Administration ● Deep Learning and Neural Networks ● Design Pattern and Refactoring ● Digital Image Processing ● Distributed Database Systems ● Distributed Systems ● Embedded Systems ● Enterprise Application Development ● Enterprise Architecture ● Expert Systems and Knowledge Management ● Free Space Optics ● Games Design and Development ● Information Retrieval ● Information Systems ● Internet of Things ● IPhone Applications Development ● Machine Learning ● Mobile Application Development ● Multi-Agent Systems ● Multimedia Security ● Natural Language Processing ● Network Flows ● Next Generation Networks ● Pervasive Smart Environments ● Secure Software Development ● System Integration and Architecture ● Web Technologies
Courses	5
Credit Hrs	15
Total Courses	42
Total Credit Hrs	**132**

7. Major Changes in Structure

Newly Added Courses

- Introduction to ICT
- Applied Physics
- Foreign Language
- Professional Practices
- Social Service
- Parallel and Distributed Computing

Courses Removed

- Introduction to Computing
- Basic Electronics
- Computer Architecture
- Human Computer Interaction

Credit Hours

- Credit hours in revised Roadmap 132
- Credit hours in previous Roadmap 131

No. of Courses

- Number of courses in revised Roadmap are 42
- Number of courses in previous Roadmap are 41

1st Semester				2nd Semester			
Pre-Req	**Course Code**	**Course**	**Cr. Hr.**	**Pre-Req**	**Course code**	**Course**	**Cr. Hr.**
	CC1021	Programming Fundamentals	3		ISL101	Islamic Studies	3
	CC1021L	Programming Fundamentals Lab	1	EN111	EN125	Composition and Communication	3
	MA100	Calculus and Analytical Geometry **(Calculus 1)**	3		MA103	Calculus 2	3
	IT1091	Introduction to ICT	2	PF	CC1022	Object Oriented Programming	3
	IT1091L	Introduction to ICT Lab	1	PF	CC1022L	Object Oriented Programming Lab	1
	EN111	English Grammar and Comprehension	3		CC1041	Discrete Structures	3
	NS125	Applied Physics	2				
	NS125L	Applied Physics Lab	1				
		Semester Total	**16**			**Semester Total**	**16**
3rd Semester				**4th Semester**			
Pre-Req	**Course Code**	**Course**	**Cr. Hr.**	**Pre-Req**	**Course code**	**Course**	**Cr. Hr.**
OOP	CC2042	Data Structures and Algorithms	3	DLD	CS2032	Computer Organization and Assembly Language	3
OOP	CC2042L	Data Structures and Algorithms Lab	1	DLD	CS2032L	Computer Organization and Assembly Language Lab	1
	MA210	CS Supporting 1 (Linear Algebra)	3	DSA	CC2141	Database Systems	3
	SD422	Foreign Language	2	DSA	CC2141L	Database Systems Lab	1
	HU4092	Professional Practices	3		CS2051	CS Supporting 2 (Numerical Analysis)	3
AP	CS2031	Digital Logic Design	3			Technical Elective 1	3
AP	CS2031L	Digital Logic Design Lab	1		MA150	Probability and Statistics	3
		Semester Total	**16**			**Semester Total**	**17**

8. Semester-wise road map

Total Credit Hours **132**

5th Semester				6th Semester			
Pre-Req	**Course Code**	**Course**	**Cr. Hr.**	**Pre-Req**	**Course code**	**Course**	**Cr. Hr.**
DSA	CC3011	Operating Systems	3		CC3071	Computer Networks	3
DSA	CC3011 L	Operating Systems Lab	1		CC3071L	Computer Networks Lab	1
	CS3043	Theory of Automata	3	EN125	EN220	Research Paper Writing and Presentation	3
	MA230	CS Supporting 3 (Differential Equations)	3	DIS	CS3151	Artificial Intelligence	3
	CC2101	Software Engineering	3	DIS	CS3151L	Artificial Intelligence Lab	1
DSA	CS3044	Analysis of Algorithms	3	TOA	CS3045	Compiler Construction	3
					SS3080	Social Service	1
						Technical Elective 2	3
		Semester Total	**16**			**Semester Total**	**18**
7th Semester				**8th Semester**			
Pre-Req	**Course Code**	**Course**	**Cr. Hr.**	**Pre-Req**	**Course code**	**Course**	**Cr. Hr.**
OS	CS4172	Parallel and Distributed Computing	3		CC3121	Information Security	3
		Technical Elective 3	3			Univ. Elective 2	3
		Technical Elective 4	3			Univ. Elective 3	3
	POL101	Pakistan Studies	3			Technical Elective 5	3
	CC4181	FYP 1	3		CC4182	FYP 2	3
		Univ. Elective 1	3				
		Semester Total	**18**			**Semester Total**	**15**
							132

9. Course Outlines

Programming Fundamentals					
Credit Hours:	3 (3,0)	**Prerequisites:**			
Course Learning Outcomes (CLOs):					
At the end of the course the students will be able to:				**Domain**	**BT Level***
Understand basic problem solving steps and logic constructs				C	2
2. Apply basic programing concepts				C	3
3. Design and implement algorithms to **solve** real world problems.				C	3
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain					
Course Content:					
Introduction to problem solving, a brief review of Von-Neumann architecture, Introduction to programming, role of compiler and linker, introduction to algorithms, basic data types and variables, 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					
Teaching Methodology:					
Lectures, Written Assignments, Practical labs, Semester Project, Presentations					
Course Assessment:					
Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam					
Reference Materials:					
1. Starting out with Python, 4th Edition, Tony Gaddis. 2. Starting out with Programming Logic & Degins, 4th Edition, Tony Gaddis, 3. The C Programming Language, 2nd Edition by Brian W. Kernighan, Dennis M. Ritchie 4. Object Oriented Programming in C++ by Robert Lafore 5. Introduction to Computation and Programming Using Python: With Application to Understanding Data, 2nd Edition by Guttag, John 7. C How to Program, 7th Edition by Paul Deitel & Harvey Deitel 8. Problem Solving and Program Design in C++, 7th Edition by Jeri R. Hanly & Elliot B. Koffman					

Object Oriented Programming			
Credit Hours:	3 (3,0)	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand principles of object oriented paradigm.	C	2
2. Identify the objects & their relationships to build object oriented solution	C	3
3. Model a solution for a given problem using object oriented principles	C	3
4. Examine an object oriented solution.	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Starting Out with C++ from Control Structures to Objects, 9th Edition, Tony Gaddis
2. C++ How to Program, 10th Edition, Deitel & Deitel.
3. Object Oriented Programming in C++, 3rd Edition by Robert Lafore
4. Java: How to Program, 9th Edition by Paul Deitel
5. Beginning Java 2, 7th Edition by Ivor Horton
6. An Introduction to Object Oriented Programming with Java, 5th Edition by C. Thomas Wu

Data Structures and Algorithms			
Credit Hours:	3 (3,0)	**Prerequisites:**	OOP

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Implement various data structures and their algorithms, and **apply** them in implementing simple applications.	C	2,3
2. **Analyze** simple algorithms and determine their complexities.	C	4,5
3. **Apply** the knowledge of data structures to other application domains.	C	3
4. **Design** new data structures and algorithms to solve problems.	C	6

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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 tree traversals, binary search trees, heaps, M-way tress, balanced trees, graphs, breadth-first and depth-first traversal, topological order, shortest path, adjacency matrix and adjacency list implementations, memory management and garbage collection.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Data Structures and Algorithms in C++ by Adam Drozdek
2. Data Structures and Algorithm Analysis in Java by Mark A. Weiss
3. Data Structures and Abstractions with Java by Frank M. Carrano & Timothy M. Henry
4. Data Structures and Algorithm Analysis in C++ by Mark Allen Weiss
5. Java Software Structures: Designing and Using Data Structures by John Lewis

Discrete Structures					
Credit Hours:	3 (3,0)	**Prerequisites:**			
Course Learning Outcomes (CLOs):					
At the end of the course the students will be able to:				**Domain**	**BT Level***
1. **Understand** the key concepts of Discrete Structures such as Sets, Permutations, Relations, Graphs, and Trees etc.				C	2
2. **Apply** formal logic proofs and/or informal, but rigorous, logical reasoning to real problems, such as predicting the behavior of software or solving problems such as puzzles.				C	3
3. **Apply** discrete structures into other computing problems such as formal specification, verification, databases, artificial intelligence, and cryptography.				C	3
4. **Differentiate** various discrete structures and their relevance within the context of computer science, in the areas of data structures and algorithms, in particular.				C	4
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain					
Course Content:					
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.					
Teaching Methodology:					
Lectures, Written Assignments, Practical labs, Semester Project, Presentations					
Course Assessment:					
Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam					
Reference Materials: 1. Starting Out with C++ from Control Structures to Objects, 9th Edition, Tony Gaddis 2. C++ How to Program, 10th Edition, Deitel & Deitel. 3. Object Oriented Programming in C++, 3rd Edition by Robert Lafore 4. Java: How to Program, 9th Edition by Paul Deitel 5. Beginning Java 2, 7th Edition by Ivor Horton 6. An Introduction to Object Oriented Programming with Java, 5th Edition by C. Thomas Wu					

Computer Networks			
Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. **Describe** the key terminologies and technologies of computer networks	C	2
2. **Explain** the services and functions provided by each layer in the Internet protocol stack.	C	2
3. **Identify** various internetworking devices and protocols, and their functions in a network.	C	4
4. **Analyze** working and performance of key technologies, algorithms and protocols.	C	4
5. Build Computer Network on various Topologies	P	3

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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, IP addressing, sub netting, CIDR, routing protocols, transport layer protocols, ports and sockets, connection establishment, flow and congestion control, application layer protocols, latest trends in computer networks.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

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

Database Systems			
Credit Hours:	3 (3,0)	**Prerequisites:**	DSA

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. **Explain** fundamental database concepts.	C	2
2. **Design** conceptual, logical and physical database schemas using different data models.	C	5
3. **Identify** functional dependencies and resolve database anomalies by normalizing database tables.	C	2
4. **Use** Structured Query Language (SQL) for database definition and manipulation in any DBMS	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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.

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

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

Operating Systems			
Credit Hours:	3 (3,0)	**Prerequisites:**	DSA

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. **Understand** the characteristics of different structures of the Operating Systems and identify the core functions of the Operating Systems.	C	2
2. **Analyze** and **evaluate** the algorithms of the core functions of the Operating Systems and explain the major performance issues with regard to the core functions.	C	4,5
3. **Demonstrate** the knowledge in applying system software and tools available in modern operating systems.	C	3

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Operating systems basics, system calls, process concept and scheduling, inter-process communication, 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 system security

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

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

Software Engineering			
Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Describe various software engineering processes and activities	C	1
2. Apply the system modeling techniques to model a medium size software system	C	2
3. Apply software quality assurance and testing principles to medium size software system.	C	3
4. Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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, Requirements engineering 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, Software testing and quality assurance, Software evolution, Project management and project planning, configuration management, Software Process improvement.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Software Engineering, Sommerville I., 10th Edition, Pearson Inc., 2014
2. Software Engineering, A Practitioner's Approach, Pressman R. S.& Maxim B. R., 8th Edition, McGraw-Hill, 2015.

Information Security			
Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. **Explain** key concepts of information security such as design principles, cryptography, risk management, and ethics	C	2
2. **Discuss** legal, ethical, and professional issues in information security.	A	2
3. **Apply** various security and risk management tools for achieving information security and privacy.	C	3
4. **Identify** appropriate techniques to tackle and solve problems in the discipline of information security.	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Information security foundations, security design principles; security mechanisms, symmetric and asymmetric cryptography, encryption, hash functions, digital signatures, key management, 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.

Teaching Methodology:

Lectures, Written Assignments, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

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

Introduction to Information and Communication Technologies			
Credit Hours:	3 (2,1)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Principles of writing good English, understanding the composition process: writing clearly; words, sentence and paragraphs; Comprehension and expression; Use of grammar and punctuation. Process of writing, observing, audience collecting, composing, drafting and revising, persuasive writing, reading skills, listening skills and comprehension, skills for taking notes in class, skills for exams; Business communications; planning messages, writing concise but with impact. Letter formats, mechanics of business, letter writing, letters, memo and applications, summaries, proposals, writing resumes, styles and formats, oral communications, verbal and non-verbal communication, conducting meetings, small group communication, taking minutes. Presentation skills; presentation strategies, defining the objective, scope and audience of the presentation, material gathering material organization strategies, time management, opening and concluding, use of audio-visual aids, delivery and presentation.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Practical Business English, Collen Vawdrey, 1993, ISBN = 0256192740
2. Effective Communication Skills: The Foundations for Change, John Nielsen, 2008, ISBN = 1453506748

Professional Practices			
Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

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

Digital Logic Design			
Credit Hours:	3+1	**Prerequisites:**	Applied Physics

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Acquire knowledge related to the concepts, tools and techniques for the design of digital electronic circuits 2. Demonstrate the skills to design and analyze both combinational and sequential circuits using a variety of techniques 3. Apply the acquired knowledge to simulate and implement small-scale digital circuits 4. Understand the relationship between abstract logic characterizations and practical electrical implementations.		

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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. Binary Arithmetic and Arithmetic Circuits, Memory Elements, State Machines. Introduction Programmable Logic Devices (CPLD, FPGA); Lab Assignments using tools such as Verilog HDL/VHDL, MultiSim

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Digital Fundamentals by Floyd, 11/e.
2. Fundamental of Digital Logic with Verilog Design, Stephen Brown, 2/e.

Computer Organization and Assembly Language			
Credit Hours:	3+1	**Prerequisites:**	Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Acquire the basic knowledge of computer organization, computer architecture and assembly language 2. Understand the concepts of basic computer organization, architecture, and assembly language techniques 3. Solve the problems related to computer organization and assembly language		

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction to computer systems: Information is bits + context, programs are translated by other 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, outof-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, general principles of pipelining, pipelined Y86 implementations

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

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

Artificial Intelligence

Credit Hours:	3 + 1	**Prerequisites:**	Data Structures and Algorithms

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
Understand key components in the field of artificial intelligence	C	2
Implement classical artificial intelligence techniques	C	3
Analyze artificial intelligence techniques for practical problem solving	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction (Introduction, basic component of AI, Identifying AI systems, branches of AI, etc.); Reasoning and Knowledge Representation (Introduction to Reasoning and Knowledge Representation, Propositional Logic, First order Logic); Problem Solving by Searching (Informed searching, Uninformed searching, Local searching.); Constraint Satisfaction Problems; Adversarial Search (Min-max algorithm, Alpha beta pruning, Game-playing); Learning (Unsupervised learning, Supervised learning, Reinforcement learning) ;Uncertainty handling (Uncertainty in AI, Fuzzy logic); Recent trends in AI and applications of AI algorithms (trends, Case study of AI systems, Analysis of AI systems)

Teaching Methodology:

Lectures, Assignments, labs, Projects, Presentations, etc. Major component of the course should be covered using conventional lectures. Practical contact hours are compulsory (~45 hours in a semester).

Course Assessment:

Exams, Assignments, Quizzes, Project, Presentations. Course will be assessed using a combination of written examinations and project(s). Practical evaluation, using rubrics, is encouraged and suggested to make up around 20% of the course.

Reference Materials:

1. Stuart Russell and Peter Norvig, Artificial Intelligence. A Modern Approach, 3rd edition, Prentice Hall, Inc., 2010.
2. Hart, P.E., Stork, D.G. and Duda, R.O., 2001. Pattern classification. John Willey & Sons.
3. Luger, G.F. and Stubblefield, W.A., 2009. AI algorithms, data structures, and idioms in Prolog, Lisp, and Java. Pearson Addison-Wesley.

Design and Analysis of Algorithms			
Credit Hours:	3	**Prerequisites:**	Data Structures and Algorithms

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Explain what is meant by "best", "expected", and "worst" case behavior of an algorithm 2. Identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors. 3. Determine informally the time and space complexity of simple algorithms 4. List and contrast standard complexity classes 5. Use big O, Omega, Theta notation formally to give asymptotic upper bounds on time and space complexity of algorithms 6. Use of the strategies(brute-force, greedy, divide-andconquer, and dynamic programming) to solve an appropriate problem 7. Solve problems using graph algorithms, including singlesource and all-pairs shortest paths, and at least one minimum spanning tree algorithm 8. Trace and/or implement a string-matching algorithm		

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

Introduction; role of algorithms in computing, Analysis on nature of input and size of input Asymptotic 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 complexity classes;

Teaching Methodology:

Lectures, Written Assignments, Semester Project.

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Final Exam

Reference Materials:

1. Introduction to Algorithms (3rd edition) by Thomas H. Corman, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein
2. Algorithm Design, (1st edition, 2013/2014), Jon Kleinberg, Eva Tardos,
3. Algorithms, (4th edition, 2011), Robert Sedgewick, Kevin Wayne

Theory of Automata					
Credit Hours:	3	**Prerequisites:**			
Course Learning Outcomes (CLOs):					
At the end of the course the students will be able to:				**Domain**	**BT Level***
1. Explain and manipulate the different concepts in automata theory and formal languages such as formal proofs, automata, regular expressions, Turing machines etc; 2. Prove properties of languages, grammars and automata with rigorously formal mathematical methods 3. Design of automata, RE and CFG 4. Transform between equivalent NFAs, DFAs and REs 5. Define Turing machines performing simple tasks. 6. Differentiate and manipulate formal descriptions of languages, automata and grammars with focus on regular and context-free languages, finite automata and regular expressions.					
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain					
Course Content:					
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.					
Teaching Methodology:					
Lectures, Written Assignments, Practical labs, Semester Project, Presentations					
Course Assessment:					
Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam					
Reference Materials: 1. Introduction to computer theory, Daniel I. A. Cohen, 2nd Edition 2. Automata, Computability and Complexity: Theory and Applications, by Elaine Rich, 2011 3. An Introduction to Formal Languages and Automata, by Peter Linz, 4th edition, Jones & Bartlett Publishers, 2006 4. Theory of Automata, Formal Languages and Computation, by S. P. Eugene, Kavier, 2005, New Age Publishers					

Compiler Construction			
Credit Hours:	3	**Prerequisites:**	Theory of Automata

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Understand the basic techniques used in compiler construction such as lexical analysis, top-down, bottom-up parsing, context-sensitive analysis, and intermediate code generation 2. Understand the basic data structures used in compiler construction such as abstract syntax trees, symbol tables, three-address code, and stack machines. 3. Design and implement a compiler using a software engineering approach 4. Use generators (e.g. Lex and Yacc)		

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Compilers: Principles, Techniques, and Tools, A. V. Aho, R. Sethi and J. D. Ullman, Addison-Wesley, 2nd ed., 2006
2. Modern Compiler Design, D. Grune, H. E. Bal, C. J. H. Jacobs, K. G. Langendoen, John Wiley, 2003.
3. Modern Compiler Implementation in C, A. W. Appel, M. Ginsburg, Cambridge University Press, 2004.

Parallel and Distributed Computing			
Credit Hours:	3	**Prerequisites:**	Operating Systems

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Learn about parallel and distributed computers. 2. Write portable programs for parallel or distributed architectures using Message-Passing Interface (MPI) library 3. Analytical modelling and performance of parallel programs. 4. Analyze complex problems with shared memory programming with openMP.		

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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).

Teaching Methodology:

Lectures, Written Assignments, Practical labs, Semester Project, Presentations

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

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

Technical Elective Courses

Database Administration & Management			
Credit Hours:	3 (3,0)	**Prerequisites:**	Database Systems

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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, Integrity and security, Database Administration, Physical database design and tuning, Distributed database systems, Emerging research trends in database systems.

Teaching Methodology:

Lecturing, Written Assignments, Project & Research

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Presentation, Final Exam

Reference Materials:

1. Fundamentals of Database Systems, by Ramez Elmasri and Shamkant Navathe, Addison Wesley, 5th Edition.
2. Database System Concepts by Henry F. Korth and Abraham Silberschatz, 4th edition, McGraw Hill, 2002, ISBN: 0-07-12268-0

Web Technologies			
Credit Hours:	3 (3,0)	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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 any Scripting language. Search Technologies. Search Engine Optimization. XML Query Language, Semantic Web, Future Web Application Framework.

Teaching Methodology:

Lecturing, Written Assignments, Presentation, Final Exam

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Report Writing, Presentation, Final Exam

Reference Materials:

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

Big Data Analytics			
Credit Hours:	3 (2, 1)	**Prerequisites:**	Probability and Statistics, Programming Fundamentals

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Provide fundamental information to get insight into the challenges with big data.	C	1
2. Understand techniques for storing and processing large amounts of structured and unstructured data	C	2
3. Application of big data concepts to get valuable information on market trends	C	3
4. Implement and deploy a sample project for extracting useful information from a mid-sized dataset.	C	4

* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain

Course Content:

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.

Teaching Methodology:

Lecturing, Written Assignments, Project, Report Writing

Course Assessment:

Sessional Exam, Home Assignments, Quizzes, Project, Presentations, Final Exam

Reference Materials:

1. Mining of Massive Datasets, Jure Leskovec, Anand Rajaraman, Jeff Ullman, 2nd edition, 2011
2. Hadoop: The Definitive Guide, Tom White, 4th edition. 2009.
3. Data-Intensive Text Processing with Map Reduce, Jimmy Lin and Chris, 2010.

Computer Graphics			
Credit Hours:	3	**Prerequisites:**	

Course Learning Outcomes (CLOs):		
At the end of the course the students will be able to:	**Domain**	**BT Level***
1. Comprehend the structure of modern computer graphics systems 2. Explain the basic principles of implementing computer graphics fundamentals 3. Compare key algorithms for modelling and rendering graphical data 4. Develop design and problem solving skills with applications to computer graphics 5. Construct interactive computer graphics programs using OpenGL		
* BT= Bloom's Taxonomy, C=Cognitive domain, P=Psychomotor domain, A= Affective domain		

Course Content:

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.

Teaching Methodology:

Lectures, Written Assignments, Project, Report Writing

Course Assessment:

Midterm exam, Final Exam, Assignments

Reference Materials:

1. Computer Graphics with Open GL (4th Edition) by Donald D. Hearn, Prentice Hall, 2010, ISBN-10: 0136053580. 2. Foundations of 3D Computer Graphics by S. J. Gortler, The MIT press, 2012. 3. Fundamentals of Computer Graphics, 3rd Edition, A K Peters, 2009. 4. Computer Graphics: Principles and Practice, 3 rd Edition, Addison Wesley, 2013. 5. Real-Time Rendering, 3rd Edition, A K Peters, 2008.