
Islamic University of Science & Technology, Kashmir, J&K - 192122
Department of Computer Science SoET

Semester - II

Semester-II (26 Credit Semester)						
Course Code	**Course Name**	**Paper category**	**Hours / Week**			**Credits**
			L	T	P	
18 Core Credit Units						
MCA-2T1-C	Data Structures	Core	2	1	4	5
MCA-2T2-C	Database Management System	Core	2	1	4	5
MCA-2T3-C	Software Engineering	Core	4	0	0	4
MCA-2T4-C	Operating System	Core	4	0	0	4
6 Elective Credit Units						
Pool A: 3 Elective Credit Units						
MCA-2E1-DCE	Advanced Computer Organization & Architecture	DCE	2	1	0	3
MCA-2E2-DCE	Operational Research	DCE	2	1	0	3
MCA-2E3-DCE	Microprocessor & Assembly Language Programming	DCE	2	1	0	3
Pool B: 3 Elective Credit Units						
MCA-2E4-DCE	Modelling & Simulation	DCE	2	0	2	3
MCA-2E5-DCE	Computer Graphics	DCE	2	0	2	3
MCA-2E6-DCE	Numerical and Statistical Computing	DCE	2	0	2	3
MCA-2E7-DCE	Java Programming	DCE	2	0	2	3
2 credit units to be taken from outside departments						

Course Title:	Data Structures	**Course Code:**	MCA-2T1-C
Semester:	2nd	**Paper Type:**	Core
Credits:	05	**Max Marks::**	125
Pre Requisite:	MCA-1T1-C, MCA-1T3-C	**Co-Requisite:**	MCA-2T2-C

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVE:

The objective of this course is to introduce the students to the significant topics of basic as well as advanced data structures and subsequently make them understand & learn the commonly used data structures along with their applications. The focus is also to appreciate the need and working of different ways of storing data and demonstrate the advantages and disadvantages of specific data structures both linear as well as non-linear. The course emphasizes on lab work wherein the students learn not only to make use of different data structures, but also their application in different synthetic problems. The practical demonstration for autonomous realization of simple programs or program parts is to understand the behavior of these basic data structures and to have a thorough understanding of how data structures influence the performance of algorithms.

COURSE CONTENT:

UNIT I

Data Structures: Overview & Significance. Introduction & Analysis of Algorithms. Linear & Non Linear Data Structures. Array & Strings, Application of Arrays, Sparse Matrix, Searching & Sorting: Linear & Binary Search, Bubble Sort, Selection Sort & Insertion Sort. Recursion & Applications.

UNIT II

Stack: Operations & Applications, Polish Expressions Queue: Operations & Applications, Circular Queue: Operations & Applications. Linked List: Singly Linked List, Doubly Linked List, Circular Linked List. Linked implementation of Stack & Queue, Operations & Applications of Linked Lists.

UNIT III

Trees: Basic Concept, Terminology & Representation of a Tree. Binary Tree, Binary Search Tree, Tree Traversal Techniques, BST Sort, AVL Trees, Heap Sort, Quick Sort, Merge Sort. Threaded Binary Tree, Red Black Trees, B Trees, B* Trees, Applications of Trees.

UNIT IV

Graph: Matrix & List Representation, Elementary Graph Operations (BFS & DFS), Single Source Shortest Path, Dijkstras Algorithm, Bellman-Ford Algorithm. All Pair Shortest Paths, Transitive

Closer, Floyd-Warshall Algorithm. Spanning Trees: Minimum Cost Spanning Trees, Prims & Kruskal Algorithm. Hashing: Overview, Hashing Functions, Collision Resolution Techniques.

COURSE OUTCOMES:

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

- *Develop some simple applications, like a desk calculator using stacks.*
- *Understand advanced searching methods like B-tree, B+ tree, AVL/red-black trees.*
- *Learn to develop a basic file system.*
- *Use standard libraries for data structures.*

Text Books:

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahni, 2nd Ed.

References:

1. Data structures and Algorithms in C++ -- by Adam Drozdek (1994 2001).
2. Fundamentals of Data Structures in C -- by Horowitz, Sahni and Anderson-Freed (Silicon Press 2007).
3. Data Structure Using C and C++ -- by Y. Langsam, M. J. Augenstein and A. N. Tanenbaum (Pearson Education, 2nd Edition, 2015).

Course Title:	Database Management System	**Course Code:**	MCA-2T2-C
Semester:	2nd	**Paper Type:**	Core
Credits:	05	**Max Marks::**	125
Pre Requisite:	MCA-1T1-C, MCA-1T3-C	**Co-Requisite:**	MCA-2T1-C

Marks Distribution: (Mid Term:30, End Term:50, Lab:25, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To understand the role of a database management system in an Organization.*
- *To understand basic database concepts including the structure and Operation of the relational data model.*
- *To construct simple and moderately advanced database queries using Structured Query Language (SQL).*
- *To understand and successfully apply logical database design principles, including E-R diagrams and database normalization.*

COURSE CONTENT:

UNIT I

Basic Concepts and Conceptual Database Design: Database Users, Characteristics of the Database, Advantage of using Database Systems, Data Models, schemas and instances, Three Tier Architecture & Data Independence, Database Languages & Interfaces. Overview of Legacy Data Base Management Systems. Data Modeling Using The Entity-Relationship Model – Entities, Attributes and Relationships, Cardinality of Relationships, Strong and Weak Entity Sets, Translating your ER Model into Relational Model.

UNIT II

Relational Model, Languages & Systems: Relational Data Model, Relational Model Concepts, Relational Model Constraints, Relational Algebra, SQL – A Relational Database Language, Data Definition & Manipulation in SQL, Queries in SQL, Specifying Constraints in SQL, Practicing SQL commands using ORACLE, "PL/SQL, Stored Procedures".

UNIT III

Functional Dependencies & Normalization for Relational Databases: Functional Dependencies, Normal Forms based on primary keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependencies.
"Mini Project: Data Analysis and Data Modelling"

UNIT IV

Transaction Management: Transaction Concept and State, Desirable Properties of a Transaction, Characterizing Schedules based on Serializability and Recoverability, Concurrency Control Techniques: Lock-Based Protocols, Timestamp-based Protocols, Validation–based Protocols. Database Recovery Techniques: Recovery Concepts, Recovery based on Deferred Update and Immediate Update. Shadow Paging. Overview of Object Oriented Database Management Systems, Distributed Data Base Management Systems.

COURSE OUTCOMES:

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

- *Introduces the role of a database management system, basic database concepts, including the structure and operation of the relational data model.*
- *Familiarize themselves with the concepts of integrity constraints, relational algebra, relational domain & tuple calculus, data normalization.*
- *Construct simple and moderately advanced database queries using Structured Query Language (SQL).*
- *Have knowledge of database transaction including concurrency control, backup and recovery, and data object locking.*
- *Design and implementation of a small database project using Oracle.*

Text Books:

1. Korth, Silberschatz, "Database System Concepts", TMH
2. Elmsari & Navathe, "Fundamentals of Database Systmes", A. Wesley
3. Ullman J. D., "Principles of Database Systems", Galgotia Publications

References:

1. Steve Bobrowski, "Oracle 8 Architecture", TMH
2. Date C. J., "An Introduction to Database Systems",Narosa Publishing
3. William Page, "Using Oracle 8i – Special Edition", Que/PHI
4. Ivan Bayross, "SQL & PL/SQL Using Oracle 8i & 9i with SQLJ", BPB
5. Desai.B, "An introduction to Database Concepts", Galgotia Publications

Course Title:	Software Engineering	**Course Code:**	MCA-2T3-C
Semester:	2nd	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	MCA-1T1-C, MCA-1T3-C	**Co-Requisite:**	MCA-2T1-C, MCA-2T2-C

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To familiarize students with the fundamentals of software engineering.*
- *To introduce students to the basic concepts and principles of structured software engineering.*
- *To make students understand the software management issues.*
- *To introduce the students to the basic concepts of OOSE.*
- *To study advanced concepts in software engineering like Reverse engineering, Reengineering, etc.*

COURSE CONTENT:

UNIT I

Software Engineering: Definition & Evolution, its Role & Impact in Computer Science. Software Process, Characteristics of a SW Process, CMMI, TSP & PSP, Software Product, Characteristics of a Good Software Product, Software Process Models, Comparative Study & Applications. Basic concepts of Agile Process. Software Requirements Analysis (SRA): Requirements - Types, Steps Involved in SRA. SW Requirements Specification (SRS): Need & Characteristics for an SRS, Components of an SRS, Prototype for a Good SRS. Structured Analysis: DFD'S, Control Flow Diagrams, Data Dictionary, State Transition Diagrams, & Entity - Relationship Diagrams. Case Study: Developing a Complete SRS.

UNIT II

Software Design: Concepts & Principles, Design Considerations & Good Design. Characterization of Effective Modular Design (Functional Independence, Cohesion, Coupling). Design: Architectural Design, Procedural Design, Interface Design, & Data Design. SW Architecture Styles: (Dataflow, Call & Return Architectures, Independent Process Architectures, Virtual Machine Architectures). Concept of Verification & Validation. Goals of SW Testing, Testing Principles.

UNIT III

Approaches to the Design of Test Cases: Black Box & White Box Testing, Techniques used by these Approaches: Basis Path & Loop Testing, Graph Based Testing, Equivalence Partitioning, Cyclomatic Complexity, Documentation of Test Cases, Phases in Testing Activity: Unit, Integration, Validation & System Tests. Software Project Management, Phases of Management, Project Planning & Control, Scheduling, Organization & Team Structures, Project Estimation Techniques – KLOC, FP & COCOMO, Risk Analysis & Management, , "Risk Exposure", Software Quality Assurance, Software Configuration Management.

UNIT IV

Technical Metrics for Software. Object Oriented Software Engineering: Object Oriented Paradigm, Concepts - Classes & Objects, Inheritance, Abstraction, Polymorphism, etc. OOA & OOD. Design Methodology – Dynamic Modeling, Functional Modeling. Advanced Concepts: Software Reuse, Re-engineering, Reverse Engineering, Restructuring, Client/Server Software Engineering, Computer Aided Software Engineering, Advances & Future Scope in Software Engineering.

COURSE OUTCOMES:

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

- *Describe software engineering layered technology and process framework.*
- *Understand theories, models, and techniques that provide a basis for the software development life cycle.*
- *Understand software testing approaches including verification and validation, static analysis, reviews, inspections, and audits.*
- *Understand the role of project management including planning, scheduling, risk management, etc.*
- *Work as an individual and/or in team to develop and deliver quality software.*

Text Books:

1. Pressman, Roger, "Software Engineering- A Practitioners Approach", McGraw Hill

References:

1. Gheezi, Jazayeri Et Al, "Fundamentals Of Software Engineering", PHI
2. Ian Sommerville, "Software Engineering", Pearson Education
3. PankajJalote, "An Integrated Approach To Software Engineering",Narosa
4. Peters & Pedrycz, "Software Engineering an Engineering Approach", Wiley

Course Title:	Operating System	**Course Code:**	MCA-2T4-C
Semester:	2nd	**Paper Type:**	Core
Credits:	04	**Max Marks:**	100
Pre Requisite:	MCA-1T1-C, MCA-1T2-C	**Co-Requisite:**	MCA-2T1-C, MCA-2T3-C

Marks Distribution: (Mid Term:30, End Term:50, Viva:10, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To learn the fundamentals of Operating Systems.*
- *To learn the mechanisms of OS to handle processes and threads and their communication*
- *To learn the mechanisms involved in memory management I/O management and file management in OS*
- *To gain knowledge on Mutual exclusion algorithms, deadlock detection algorithms*
- *To know the components and management aspects of concurrency management*
- *To implement the concepts learnt through case studies of different operating systems.*

COURSE CONTENT:

UNIT I

Overview of an Operating system, Functions of Operating System. Process and thread Management Concepts: Process Scheduling, Scheduling Criteria, Scheduling algorithms, Thread Scheduling. Process Synchronization, Semaphores, Critical Section and Monitors, Inter-process Communication, shared memory model and message passing. Deadlocks: Concept of Deadlock, Deadlock prevention, avoidance, detection and Deadlock recovery", Case Studies: LINUX/Windows OS/Android OS (Any one)

UNIT II

Memory Management: Linking, Loading, Memory Allocation, Design Issues & Problems, Fragmentation, Compaction, Memory Management Unit, Paging, Segmentation, Virtual Memory, Demand Paging. Page Replacement Algorithms. Allocation Algorithms, Thrashing Case Studies: LINUX/Windows OS/Android OS (Any one)

UNIT III

File Management: File Structure, File Protection, File System Implementation, Directory Structure, Free Space Management, Allocation Methods, Efficiency & Protection. Case Studies: UNIX/LINUX/Windows NT OS Disk Management: Disk Structure, Disk Scheduling Algorithm, Disk Management, Swap Space concept & Management, RAID Structure, Disk Performance issues. Case Studies: LINUX/Windows OS/android OS (Any One)

UNIT IV

Multiprocessor Systems: Types of Multiprocessor Operating Systems, Functions & Requirements, Design & Implementation Issues. Distributed Operating System: Difference between Distributed & Centralized Operating System, Advantages & Disadvantages of Distributed Operating System,

Hardware & Software Concepts, Loosely Coupled Systems, Types of Distributed Operating System.

COURSE OUTCOMES:

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

- *Master understanding of design issues associated with operating systems.*
- *Master various process management concepts including scheduling, synchronization, and deadlocks.*
- *Be familiar with various types of operating systems including UNIX.*

Text Books:

1. Tanenbaum, A.S., "Modern Operating System", PHI
2. Peterson, J.L. Abraham, Silberschatz, "Operating System Concepts", Addison Wesley

References:

1. Dietel H.M. "An Introduction To Operating System", Addison Wesley
2. Karnetkar, "UNIX Shell Programming", BPB
3. W.Stallings, "Operating systems"
4. Dhamdhere, "An Operating System –Design & principles"
5. Madnick E, Donovan J, "Operating Systems", TMH
6. Marko Gergent, "Learning android", O'rielly

Course Title:	Advanced Computer Organization & Architecture	**Course Code:**	MCA-2E1-DCE
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA-1T2-C	**Co-Requisite:**	MCA-2T1-C, MCA-2T4-C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To provide an insight about Classification of parallel computers, advanced processor, pipelining, memory design & architecture and multi-processor architecture.*
- *To Understand concepts of parallel processing and design choices of implementing parallel execution within a single processor (pipeline, VLIW, and superscalar) and multiprocessor systems.*
- *To gain knowledge of the state of the art research topics on advanced computing systems.*

COURSE CONTENT:

UNIT I

Computational Models : Introduction , Interpretation of the concept of a computational model , Relationship between the concepts of computational model , programming language & architecture , Basic Computational models , The Von , Neumann computational model ,Key concepts related to computational models .The concept of computer architecture : Evolution & interpretation of the concept of Computer Architecture at different levels of abstraction. The concept of computer architecture at multilevel hierarchical framework.

UNIT II

Introduction to Parallel Processing: Basic Concepts about program, process, thread, process & treads in languages, concurrent & parallel execution, concurrent & parallel programming languages, Types & levels of Parallelism, Classification of Parallel architectures, Basic Parallel Techniques

UNIT III

Introduction to Instruction level Parallel Processors, Evolution & overview, dependencies, instruction scheduling, preserving sequential consistency, Pipelined Processors, Basic Concepts, Pipelined instruction processing. VLIW, Basic Principles, Superscalar processing, introduction, superscalar instruction issue, shelving, parallel execution, preserving the sequential consistency of instruction execution & exception processing

COURSE OUTCOMES:

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

- *Understand the basic structure of computer.*
- *Familiarize the instructions in central processing unit of a computer.*
- *Understand memory organization in a computer.*
- *Understand input/output mechanisms.*
- *Understand parallel processing in a computer.*

Text Books:

1. Advanced Computer Architecture DEZSO SIMA, TERENCE Mountain, PETER KACSUK, Pearson Education, Fifth Indian reprint 2004.

References:

1. V.C. Hamacher. A.G. Vranesic & S. G. Zaky: "Computer Organization", Tata McGraw Hill.
2. J.P. Hayes: "Computer Architecture & Organization", McGraw Hill.
3. Morris Mano: "Computer System Architecture", Pearson Education, 3/e.

Course Title:	Operational Research	**Course Code:**	MCA-2E2-DCE
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA-1T3-C	**Co-Requisite:**	MCA-2T1-C, MCA-2T2-C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To identify and develop operational research models from the verbal description of the real system.*
- *To understand the mathematical tools that are needed to solve optimization problems.*
- *To define and formulate linear programming problems and appreciate their limitations.*
- *To understand how to solve linear programming problems using appropriate techniques and optimization solvers, interpret the results obtained and translate solutions into directives for action.*
- *To conduct and interpret post-optimal and sensitivity analysis and explain the primal-dual relationship.*
- *To develop mathematical skills to analyze and solve integer programming and network models arising from a wide range of applications.*

COURSE CONTENT:

UNIT I

Linear Programming: L P formulation, Graphical methods for LPP with 2 variables, Simplex Algorithm for LPP, Duality theorem in linear programming & applications. Transportation problem: Formulation, methods of selecting initial feasible solutions, Degeneracy & resolution. Assignment problem: Hungarian Method for solving Assignment Problems, Balanced & Unbalanced problems & resolution.

UNIT II

Network Analysis: Shortest routes, Enumeration& applications. Critical Path Method (CPM) & Programme Evaluation and Review Technique(PERT): Use & design of CPM & PERT. Critical Path calculation, Float & its Types, Crashing in Project Management.
Sequencing problems, Johnsons algorithm for processing m jobs through 2, 3 & 'n' machines. Max flow problem, Min Cut & max-flow min-cut theorems

UNIT III

Replacement & Sequencing models: Replacement of items that fail & deteriorate. Group & individual replacement.

Game theory: definition & explanation, saddle points, Dominance mixed strategies, games without saddle points, 2*N games.

Dynamic Programming: Characteristics of dynamic programming problem, Bellman's optimality principles, dynamic programming under certainty, shortest route problem, Inventory models: introduction to inventory problems & their analytical structure

COURSE OUTCOMES:

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

- *Formulate and solve linear Programming Problems*
- *Determine the optimum solution to constrained and unconstrained*
- *Apply dynamic programming principle to Linear programming problems.*
- *Determine the integer solutions to Linear Programming Problems*

Text Books:

1. N.D.Vohra, "Quatitative Techniques in management", TMH
2. Hamdy A. Taha, "Operations Research: An Introduction", Pearson

References:

1. Sharma J. K., "Operations Research: Theory & Applications", Macmillan India
2. Gross Donald, "Fundamentals of Queuing Theory", 3rd Ed., John Wiley
3. Mokhtar S. Bazaraa, "Linear Programming & Network Flows", John Wiley
4. Hiller Lieberman, "Introduction to Operations Research", TMH
5. Laudon, "Decsion Support Systems", PHI

Course Title:	Microprocessor & Assembly Language Programming	**Course Code:**	MCA-2E3-DCE
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA-1T1-C, MCA-1T2-C	**Co-Requisite:**	MCA-2E1-DCE

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To understand the principles of instruction set architecture and assembly language programming.*
- *To understand basic procedures of how a compiler translates C/C++ code to assembly language and perform simple optimizations.*
- *To understand basic principles of interrupt/exception handling.*
- *To explore in detail a simple hardware CPU implementation that supports a small instruction subset.*

COURSE CONTENT:

UNIT I

Software Model of 8088 / 8086 Microprocessor, Memory Add. Space & Data Organization, Data Types, Segment Registers, Memory Segmentation Dedicated, reserved &general use of Memory, generating an Memory Address, Pin-out diagram of 8086 Microprocessor.

UNIT II

The Microcomputer Organization, Assembly Language Programming Development on PC, Instruction Set, Addressing Modes, 8086 Instruction set, Integer Instructions & Computations, Data Transfer, Arithmetic, Logic Shift, Rotate Instruction, Flag Control, Compare, Control Flow & Jump, Subroutine & Subroutine Handling Instructions, Loop & Loop Handling, String & String Handling Instructions. Statement Syntax for a source Program, Assembler Directives, Assembling, Linking, Loading & executing a run Module.

UNIT III

Isolated I / O, Memory Mapped I/O, DMA Controller, Programming Communication Interfaces Controller. Interfacing I/O devices to microprocessor, programmable peripheral interface. Interrupt, Mechanism, Types & Priority, Interrupt Vector table, Real Mode. 8086 / 8088 Microprocessors, 8086 / 8088 Microprocessor's Minimum Mode, Maximum Mode Systems, Bus Cycle & Unit States, Memory Control Signals, Read & Write Bus Cycles, Memory Interface Circuits.

COURSE OUTCOMES:

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

- *Describe the architecture of 8086*
- *Develop simple program using 8086*
- *Describe the basic peripheral devices and its applications*
- *Differentiate various microprocessor architectures*

Text Books:

1. The 8088 and 8086 Microprocessors: Programming, Interfacing, Software, Hardware and Applications (Prentice Hall international editions) Walter A. Triebel; Avtar Singh

References:

1. DOUGLAS HALL "Microprocessors & Interfacing" Tata McGrawHill.
2. LIU, GIBSON et al "Microcomputer system The 8086/8088 Family" PHI.
3. PAL CHAUDHURI "Computer Organization & Designing" PHI.
4. MORRIS MANO "Computer System Architecture" Pearson Education.
5. GILMORE "Microprocessors" Wiley/ Tata McGraw Hill.

Course Title:	Modelling & Simulation	**Course Code:**	MCA-2E4-DCE
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA-1T1-C, MCA-1T2-C	**Co-Requisite:**	MCA-2T1-C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To understand different methods for random number generation*
- *To understand various probability distributions*
- *To have a clear understanding of the need for the development process to initiate the real problem.*
- *To have a clear understanding of principle and techniques of simulation methods informed by research direction.*
- *To be able to describe the components of continuous and discrete systems and simulate them*
- *To be able to model any system from different fields*
- *To be able to implement numerical algorithm to meet simple requirements, expressed in English*

COURSE CONTENT:

UNIT I

Concepts of Systems, Models, & Simulation. Distributed Lag Model, Cobweb Models, The process of a simulation Study, Exponential Growth Models, Exponential Decay Models, Type of simulation, Discrete-Event Simulation: Time-Advance Mechanisms, Components & Organization of a Discrete-Event Simulation Model. Monte Carlo Method. Simulation of Single-Server Queuing System, Simulation of an Inventory System

UNIT II

Continuous Simulation: Pure-pursuit Problem. Monte Carlo Method. Overview of Random numbers and Pseudo random numbers, Random Number Generators: Random Linear Congruently Generators, General Congruences, Composite Random number generator, Testing Random Number Generators. Generating Random Variates: General Approaches, Continuous & Discrete distributions.

UNIT III

Introduction to GPSS, GPSS block-diagrams, General Description, Simulation of a Manufacturing Shop. SNA, Function, Simulation of a Supermarket, GPSS Model of a Simple Telephone System

COURSE OUTCOMES:

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

- *Determine the properties of different types of physical systems and different types of simulations that are suitable to analyze their behaviors;*
- *Analyze data collected from real world and build input models for simulation studies;*
- *Conduct various simulation studies to investigate the behaviors of complex systems;*
- *Conduct statistical analysis of the simulation outputs; and*
- *Analyze discrete event systems through the competent use of computer simulation methods and mathematical modeling techniques.*

Text Books:

1. Averill Law and Averill M. Law, Simulation Modeling and Analysis with Ex-pertÖt Software, Forth Edition, McGraw-Hill Higher Education, 2007
2. Jerry, Banks. Discrete event system simulation. Pearson Education India, 2005.
3. System Simulation – Geoffrey Gordon, 2nd Edition, PHI
4. System Simulation with Digital computer – Narsingh Deo, PHI

References:

1. Law & Kelton, "Simulation Modeling & Analysis", McGraw Hill
2. Fred Maryanski, "Digital Computer Simulation", CBSPD
3. James A. Pyne, "Introduction to Simulation- Programming Techniques & Methods of
4. Analysis", McGraw Hill
5. Zeigler & Kim, "Theory of Modeling & Simulation", Academic Press
6. Banks et al, "Discrete event Simulation", Pearson Education

Course Title:	Computer Graphics	**Course Code:**	MCA-2E5-DCE
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA-1T1-C, MCA-1T3-C	**Co-Requisite:**	MCA-2T1-C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To explain basic primitives like Point, Line, Circle, Ellipse etc. with their mathematical equations coupled with the geometrical interpretations.*
- *To explain process of displaying continuous primitives on Discrete Display Devices.*
- *To explain few transformations like Translation, Rotation, Scaling and composition of these transformations.*
- *To explain Projections from higher dimensional setup to lower dimensions.*
- *To explain drawing of complex scenes using Splines coupled with understanding of spline specifications and few variants of splines besides familiarizing about Beziers.*
- *To acquaint students about Concepts of displaying structures with no inherent regular geometry using Fractals.*

COURSE CONTENT:

UNIT I

An Introduction Graphics System: Computer Graphics & Its Types, Application of computer graphics, Graphics Systems: Video Display Devices, Raster Scan Systems, Random Scan Systems, Display Buffer, Concept of Double Buffering & Segmentation of Display Buffer. Use of Lookup tables. "Introduction to Color models (RGB, CMY, and HSV)."

UNIT II

Output Primitives & Attributes of Output Primitives: Output Primitives Points & Lines, Line Drawing Algorithms, Circle Generating Algorithms, Scan-Line Polygon Fill Algorithm, Inside-Outside tests, Boundary-Fill Algorithm, Flood Fill Algorithm, Cell Array, Character Generation, Attributes of Output Primitives: Line attributes, Color & Grayscale Levels, Area fill Attributes, Character Attributes, Bundled Attributes. Anti-aliasing.

Two-dimensional Geometric Transformations: Basic Transformations, Matrix Representations & Homogeneous Coordinates, Composite Transformations, Reflection & Shearing.

UNIT III

Two-Dimension Viewing: The viewing Pipeline, Window to view port coordinate transformation, Clipping Operations, Point Clipping, Line Clipping, Polygon Clipping, Text Clipping, Exterior Clipping.

Three–Dimensional Concepts: Three Dimensional Display Methods, 3D Transformations, Parallel Projection & Perspective Projection

Curves & Surfaces, Splines, Spline specification, Interpolated & Approximated Splines. Bezier Splines, Bezier Curves, Cubic Bezier Curves, Bezier Surfaces. B-Splines curves & surfaces. Fractals - Fractal Generation Procedure.

COURSE OUTCOMES:

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

- *Understand concepts of computer graphics.*
- *Understand 2D and 3D transformation, clipping, splines, objects modeling, colour modeling, lighting, textures, visible surface detection.*
- *Understand algorithms to design and create computer graphics scenes.*

Text Books:

1. Hughes, John F., et al. Computer graphics: principles and practice. Pearson Education, 2014.

References:

1. W.M.Newman & Sproull. "Principles of interactive Computer Graphics" ,TMH
2. Steven Harrington." Computer Graphics a Programming Approach" McGraw Hill.
3. Plastock & Kelley. "Schaums outline of theory & problems of computer Graphics"
4. David F Frogers & J Alan Adams. "Procedural Elements of Computer Graphics" McGraw Hill
5. David F Rogers & J Alan Adams. "Mathematical Elements of Computer Graphics" McGraw Hill

Course Title:	Numerical & Statistical Computing	**Course Code:**	MCA-2E6-DCE
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks::**	75
Pre Requisite:	MCA-1T1-C, MCA-1T2-C	**Co-Requisite:**	MCA-2T1-C, MCA-2E2-DCE

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To explain concept of Polynomials, Equations, Zeros and Roots.*
- *To explain few iterative techniques for computation of roots together with their geometrical interpretation and implementation in C.*
- *To explain solving system of equation using known methods along with their geometrical interpretation and Implementation in c.*
- *To explain concepts of Curve Fitting with specific focus on Lagrange's interpolating Polynomial with a prior explanation of using linear and nonlinear curves for curve fitting.*
- *To explain few statistical concepts measures of central tendency, Dispersion etc.*
- *To explain concept of Hypothesis, Null hypothesis besides giving students knowledge about few tests which include T test, F test etc.*
- *To make students ready for courses like Artificial Intelligence and Machine Learning by giving them prior knowledge about Linear Algebra, Probability and statistics.*

COURSE CONTENT:

UNIT I

Approximations & Errors – Types of Programming Errors, Data Errors, Computer & Arithmetic Errors, Round Off and Truncation Errors. Accuracy and Precision, Measures of Accuracy, Error Propagation. Non-Linear Equations, Types of Methods to find solutions to nonlinear equations, Algorithms to Compute Roots of Equation – Methods of Tabulation or Brute Force Method, Method of Bisection, Secant Method, Newton-Raphson Method, Method for False Position. Derivation of mathematical formulas, geometric interpretation and implementation of these methods.

UNIT II

Linear Equations, Types of Methods to find solutions to linear equations. Algorithms to Solve Linear Algebraic Equations: Gauss Elimination, Gauss Jordan, Gauss Seidel, L.U. Decomposition, Lagrange Interpolated Polynomial, Newton'sMethods of INTERPOLATION – Forward difference, Backward difference. Derivation of mathematical formulasand implementation of these methods.

UNIT III

Differential Equations – Concepts and Terminology, Algorithms to solve Ordinary Differential Equations – Euler Method and Modification. The trapezoidal Rule, Simpson's Rule. 4th order R-K Method.

Derivation of mathematical formulas and implementation of these methods.

COURSE OUTCOMES:

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

- *Understand the iterative methods to find solution of polynomial and transcendental equations.*
- *Discuss methods of interpolation and curve fitting.*
- *Find the solution of linear equations using matrices.*

Text Books:

1. Rajaraman, Vaidyeswaran. Computer oriented numerical methods. PHI Learning Pvt. Ltd., 1993.

References:

1. S.C.ChapraandR.P.Canale: "Numerical methods for Engineering". Tata McGraw Hill.
2. Krishenmurty & Sen : "Numerical Algorithms"
3. V. Rajaraman "Computer oriented numerical methods." Prentice Hall of India.
4. McCalla, Thomas Richard: "Introduction to Numerical Methods & FORTRAN Programming", John Wiley & Sons, Inc.
5. Grewal, B. S.: "Higher Engineering Mathematics", Hindustan Offset Problems Series.
6. "SCHAUM'S Solved Problems Series".
7. Sharma, K. D.:"Programming in Fortran".
8. Jain, M. K., Iyengav, S. R. K., Jain, R. K.: "Numerical Methods for Scientific & Engineering Computation"+, Wiley Eastern Ltd, New Delhi.

Course Title:	Java Programming	**Course Code:**	MCA-2E7-DCE
Semester:	2nd	**Paper Type:**	DCE
Credits:	03	**Max Marks:**	75
Pre Requisite:	MCA-1T1-C, MCA-1T3-C	**Co-Requisite:**	MCA-2T1-C

Marks Distribution: (Mid Term:25, End Term:35, Viva:05, Assignment / Presentations:10)

COURSE OBJECTIVES:

- *To cover OOPS concepts like inheritance, polymorphism, overriding etc.*
- *To introduce Lambda expressions, inner classes and interfaces.*
- *To learn about threads, assertions, logging and exception handling.*
- *To know about collection framework and introduction of spring framework.*

COURSE CONTENT:

UNIT I

Fundamental Programming structures in Java: A simple java program, comments, data type, variables, operators, string, input& output, control flow, big numbers, arrays. Objects & Classes: introduction to Object Oriented programming, using predefined classes, defining your own classes, static fields & methods, method parameters, object construction, packages, the class path, documentation comments.

UNIT II

Inheritance Classes superclasses, subclasses, object: the cosmic superclass, generic array list, object wrapping & autoboxing, method with a variable number of parameters, enumeration class, reflection, design hints for inheritance. Interfaces lambda expressions & inner classes. Interfaces, examples of interfaces, lambda expressions, inner classes, proxies. Exceptions, assertion & logging dealing with errors, catching exception, using assertion, logging.

UNIT III

Generic Programming why generic programming, defining a simple generic class, generic methods, bound for type variables, generic code & virtual machine restriction & limitation, inheritance rules for generic types, wildcard types, reflection & generics. Collections Java collection framework, concrete frameworks, Maps, Views & wrappers, Algorithms, Legacy collections

Concurrency, Introduction to threads, Interrupting threads, thread states, thread properties, synchronization, blocking queues, thread safe collections, callable & futures, executors, synchronizers

COURSE OUTCOMES:

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

- *Describe the features of Java*
- *Design classes with object-oriented features*
- *Describe advanced features of Java like exception handling, multi-threading etc.*
- *Write programs in JAVA featuring its core capabilities*

Text Books:

1. Core Java Volume 1-Fundamentals by Cay S.horstman 10th edition Publisher: Prentice Hall

References:

1. Java the complete reference by Herbert Schield 10th edition Publisher:Tata Mc Graw Hill
2. Java How to program Dietel and Dietel