Course Curriculum and Detailed Syllabi

For

Master of Computer Applications (MCA)

Effective for

Students admitted in the

Academic Session 2020-21

Department of Computer Science & Engineering

School of Engineering

Harcourt Butler Technical University, Kanpur

Kanpur-208002

1. About the Department

The Department of Computer Science & Engineering was established in 1984 with a 4-year B. Tech. program in Computer Science & Engineering having an intake of 30 students. A 2-year Post Graduate Program, Master of Computer Application (MCA), with an intake of 60 students was introduced in 1987. Under IT task force recommendations, B. Tech. Information Technology Program with an intake of 60 students was introduced in the year 2000. Currently, the department is running B. Tech. Computer Science & Engineering, B. Tech. Information Technology and MCA with students' intake of 60, 30 and 60 respectively. One of the youngest, but among the most efficient departments, it is reputed for producing the best quality software engineers who serve in leading companies in India and abroad. The students have an in-depth exposure to computing environment consisting of state-of-the-art machines in different laboratories. In order to identify Industrial projects for the students and to expose them to the industrial environment, the department has continuous interaction with the Industries.

2. Vision

To excel in Computer Science & Engineering education, research, innovation and global employability.

3. Mission

1. Achieve academic excellence in Computer Science & Engineering through an innovative teaching-learning process.
2. Inculcate technical competence and collective discipline in students to excel for global employability, higher education and societal needs.
3. Establish focus research groups in leading areas of Computer Science & Engineering.
4. Sustain quality in Computer Science & Engineering education & research through continuous & rigorous assessment.

4. Program Educational Objectives (PEOs)

1. To inculcate professional culture amongst the students to take up technical/ professional positions for design, development, and problem solving in software industries and R&D organizations.
2. To prepare students as technical, ethical, responsible solution providers and entrepreneurs in various areas of computer applications.
3. To provide the necessary competence and capability in students to pursue higher studies in Institutions of International / National repute.
4. To provide analytical and technical ability to develop and innovate software systems and technologies in the leading areas of computer applications.

5. Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

6. Program Specific Outcomes (PSOs)

By the completion of Master of Computer Applications program, students will achieve the following program specific outcomes:-

1. The ability to understand, analyse and develop applications in the field of algorithms, system software, databases, web design, networking and artificial intelligence.
2. The ability to apply standard practices and strategies in software project development using suitable programming environment to deliver quality products.
3. The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.
4. The ability to use research based knowledge to do literature survey, formulate problem, design & carry-out experimentation, analyse & interpret experimental results for complex research problems.

7. Consistency/Mapping of PEOs with Mission of the Department

PEO Statements	M1	M2	M3	M4
PEO1: To inculcate professional culture amongst the students to take up technical/ professional positions for design, development, and problem solving in software industries and R&D organizations.	2	3	2	3
PEO2: To prepare students as technical, ethical, responsible solution providers and entrepreneurs in various areas of computer applications.	3	3	1	2
PEO3: To provide the necessary competence and capability in students to pursue higher studies in Institutions of International / National repute.	2	3	3	2
PEO4: To provide analytical and technical ability to develop & innovate systems and technologies in the leading areas of computer applications.	3	3	3	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High) "-": No correlation

8. Components of the curriculum

(*Program curriculum grouping based on course components)*

Sr. No	Curriculum Content			
	Course Component	**% of total number of credits of the Program**	**Total number of contact hours**	**Total Credits**
1.	Basic Sciences (BSC)	6.00	06	06
2.	Humanities and Social Sciences (HMSC)	4.00	05	04
3.	Program Core (PCC)	58.00	66	58
4.	Program Electives (PEC)	16.00	16	16
5.	Summer Training/Internship & Seminar Course (ISC)	3.00	06	03
6.	Project(s) (PRC)	13.00	26	13
	Total Credits	**100**	**130**	**100**

Department of Computer Science & Engineering

Course Structure (Semester wise)

Master of Computer Applications

(Applicable w.e.f. the Session 2020-21)

Semester-I

Sr.	Course	Course	Course Name	Credits	Details of Sessional Marks				ESM	Total
No.	Type	Code			CT	TA	Lab.	Total		Marks
1	PCC	ECA-451	Computer Concepts & Programming in C	5 (3-1-2)	15	20	15	50	50	100
2	PCC	ECA-453	Computer Organization	4 (3-1-0)	30	20	-	50	50	100
3	PCC	ECA-455	Internet & Java Programming	5 (3-1-2)	15	20	15	50	50	100
4	PCC	ECA-457	Operating Systems	4 (3-1-0)	30	20	-	50	50	100
5	BSC	BMA-451	Discrete Mathematical Structures	3 (3-0-0)	30	20	-	50	50	100
6	HSMC	HHS-454	Fundamentals of Management	4 (3-0-2)	15	20	15	50	50	100
			Total Credits	**25**						

Semester-II

Sr.	Course	Course	Course Name	Credits	Details of Sessional Marks				ESM	Total
No.	Type	Code			CT	TA	Lab.	Total		Marks
1	PCC	ECA-452	Software Engineering	4 (3-1-0)	30	20	-	50	50	100
2	PCC	ECA-454	Design & Analysis of Algorithms	4 (3-1-0)	30	20	-	50	50	100
3	PCC	ECA-456	Database Management Systems	5 (3-1-2)	15	20	15	50	50	100
4	PCC	ECA-458	Data Structures Using C	5 (3-1-2)	15	20	15	50	50	100
5	PCC	ECA-460	Computer Graphics & Animation	4 (3-0-2)	15	20	15	50	50	100
6	BSC	BMA-452	Operations Research	3 (3-0-0)	30	20	-	50	50	100
			Total Credits	**25**						

Semester-III

Sr. No.	Course Type	Course Code	Course Name	Credits	Details of Sessional Marks				ESM	Total Marks
					CT	TA	Lab.	Total		
1	PCC	ECA-551	Computer Networks	5 (3-1-2)	15	20	15	50	50	100
2	PCC	ECA-553	Object Oriented Systems Modeling	5 (3-1-2)	15	20	15	50	50	100
3	PCC	ECA-555	Software Project Management	4 (3-0-2*)	15	20	15	50	50	100
3	PCC	ECA-557	Artificial Intelligence	4 (3-1-0)	30	20	-	50	50	100
5	PEC	ECA-	Programme Elective-I	4 (3-1-0)	30	20	-	50	50	100
6	ISC	ECA-559	Summer Training/ Internship and Seminar	3 (0-0-6)	-	100	-	100	-	100
		Total Credits		**25**						

*During Practical hours students will do a Minor Project which may be extended as Major Project in Semester-IV

Programme Elective-I

1. Theory of Automata & Formal Languages (ECA-561)
2. Simulation & Modeling (ECA-563)
3. Information Security & Cyber Laws (ECA-565)
4. E_Commerce (ECA-567)
5. Internet of Things (ECA-569)

Semester-IV

Sr. No.	Course Type	Course Code	Course Name	Credits	Details of Sessional Marks				ESM	Total Marks
					CT	TA	Lab.	Total		
1	PEC	ECA-	Programme Elective-II	4 (3-1-0)	30	20	-	50	50	100
2	PEC	ECA-	Programme Elective-III	4 (3-1-0)	30	20	-	50	50	100
3	PEC	ECA-	Programme Elective-IV	4 (3-1-0)	30	20	-	50	50	100
4	PRC	ECA-592	Project	13 (0-0-26)	-	50	-	50	50	100
			Total Credits	**25**						

Programme Elective-II

1. Compiler Design (ECA-552)
2. Machine Learning (ECA-554)
3. Advanced Database Management Systems (ECA-556)
4. ERP Systems (ECA-558)
5. AI Programming using Pythan (ECA-560)

Programme Elective-III

1. Cloud Computing (ECA-572)
2. Software Quality Engineering (ECA-574)
3. Digital Image Processing (ECA-576)
4. Data Warehousing & Mining (ECA-578)
5. Mobile Application Development (ECA-580)

Programme Elective-IV

1. Cryptography & Network Security (ECA-582)
2. Soft Computing (ECA-584)
3. Embedded System (ECA-586)
4. Virtual Reality (ECA-588)
5. Data Analytics (ECA-590)

Detailed Syllabus

Ist Year

COMPUTER CONCEPTS & PROGRAMMING IN 'C' (ECA-451)

Type	L	T	P	Credits
ESC	3	1	2	5

Prerequisite: NIL

Course Content:

Unit-1:

Introduction to Computers: Computer hardware Components, peripherals and their functions, Number Systems and conversion methods, Concept of an algorithm; termination and correctness. Algorithms to programs: specification, top-down development and stepwise refinement, Introduction to programming environment, use of high level programming language for the systematic development of programs. Introduction to the design and implementation of correct, efficient and maintainable programs, Structured Programming, Trace an algorithm to depict the logic.

Unit-2:

Basic operating System Concepts: Introduction of MS-DOS, WINDOWS, and LINUX Operating Systems, Functional Knowledge of these operating systems, Introduction of basic commands of LINUX and Editors, Managing Files and Directories in LINUX, Programming Environment in LINUX, Writing and executing programs in LINUX.

Unit-3:

Programming in C: History, Introduction to C Programming Languages, Structure of C programs, compilation and execution of C programs, Debugging Techniques, Data Types and Sizes, Declaration of variables, Modifiers, Identifiers and keywords, Symbolic constants, Storage classes (automatic, external, register and static), Enumerations, command line parameters, Macros, The C Preprocessor.

Unit-4:

Operators: Unary operators, Arithmetic & logical operators, Bit wise operators, Assignment operators and expressions, Conditional expressions, Precedence and order of evaluation. Control statements: if-else, switch, break, and continue, the comma operator, goto statement. Loops: for, while, do-while. Functions: built-in and user-defined, function declaration, definition and function call, and parameter passing: call by value, call by reference, recursive functions, Multi-file programs. Arrays: linear arrays, multidimensional arrays, passing arrays to functions, Arrays and strings.

Unit-5:

Structure and Union: definition and differences, self-referential structure. Pointers: value at (*) and address of (&) operator, pointer to pointer, Dynamic Memory Allocation, calloc and malloc functions, array of pointers, function of pointers, structures and pointers. File Handling in C: opening and closing a data file, creating a data file, read and write functions, unformatted data files.

Lab Work:

1. Write C program to find largest of three integers.
2. Write C program to check whether the given string is palindrome or not.
3. Write C program to find whether the given integer is

(i). a prime number
(ii). an Armstrong number.

4. Write C program for Pascal triangle.
5. Write C program to find sum and average of n integer using linear array.
6. Write C program to perform addition, multiplication, transpose on matrices.
7. Write C program to find Fibonacci series of iterative method using user-defined function.
8. Write C program to find factorial of n by recursion using user-defined functions.
9. Write C program to perform following operations by using user defined functions:
   (i) Concatenation
   (ii) Reverse
   (iii) String Matching
10. Write C program to find sum of n terms of series: n – n*2/2! + n*3/3! - n*4/4! + ...
11. Write C program to interchange two values using
    (i). Call by value.
    (ii). Call by reference.
12. Write C program to sort the list of integers using dynamic memory allocation.
13. Write C program to display the mark sheet of a student using structure.
14. Write C program to perform following operations on data files:
    (i) Read from data file.
    (ii) Write to data file.
15. Write C program to copy the content of one file to another file using command line argument.

Text and References Books

1. Kernighan, Ritchie, "The C Programming Language", PHI
2. V. Rajaraman, "Fundamentals of Computers", PHI
3. Peter Norton's, "Introduction to Computers", TMH
4. Gottfried, "Programming in C", Schaum's Series, Tata McGraw Hill
5. Yashwant Kanitkar, "Working with C", BPB
6. E. Balagurusamy, "Programming in ANSI C", TMH

Course Outcomes

1. Identify the parts of the computer system and explain the functioning of its components alongwith the process of problem solving. (Remember, Understand)
2. Design an algorithmic solution for a given problem and translate it into a program. (Design)
3. Understand different operating systems, related concepts and their functions. (Understand)
4. Use the appropriate control statements to solve the given problem. (Apply)
5. Implement different Operations on arrays and use functions to solve the given problem. (Apply)
6. Understand pointers, structures and unions & Implement file Operations in C programming. (Understand, Apply)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	0	0	0	0	0	2	0	0	2
CO2	3	3	3	0	0	0	0	0	2	0	0	2
CO3	3	3	1	0	0	0	0	0	2	0	0	2
CO4	3	3	3	0	0	0	0	0	2	0	0	2
CO5	3	3	3	0	0	0	0	0	2	0	0	2
CO6	3	3	3	0	0	0	0	0	2	0	0	2

CO and PSO Mapping

CO/PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	1
CO2	3	2	1	1
CO3	3	2	1	1
CO4	3	2	1	1
CO5	2	1	-	-

COMPUTER ORGANIZATION (ECA-453)

Type	L	T	P	Credits
PCC	**3**	**1**	**0**	**4**

Prerequisite: NIL

Course Content:

Unit-1:

Representation of Information and Basic Building Blocks: Introduction to Computer, Computer hardware generation, Number System: Binary, Octal, Hexadecimal, Character Codes (BCD, ASCII, EBCDIC), Logic gates, Boolean Algebra, K-map simplification, Half Adder, Full Adder, Subtractor, Decoder, Encoders, Multiplexer, De-Multiplexer, Carry look ahead adder, Combinational logic Design, Flip-Flops, Registers, Counters (synchronous & asynchronous), ALU, Micro-Operation, ALU Chip, Faster Algorithm and Implementation (Multiplication & Division).

Unit-2:

Basic Organization: Von Neumann Architecture, Operational flow chart, Instruction Cycle, Organization of Central Processing Unit, Hardwired & micro programmed control unit, Single Organization, General Register Organization, Stack Organization, Addressing modes, Instruction formats, data transfer & Manipulation, I/O Organization, Bus Architecture, Programming Registers

Unit-3:

Memory Organization: Memory Hierarchy, Main memory (RAM/ROM chips), Auxiliary memory, Associative memory, Cache memory, Virtual Memory, Memory Management Hardware, hit/miss ratio, magnetic disk and its performance, magnetic Tape etc.

Unit-4:

I/O Organization: Peripheral devices, I/O interface, Modes of Transfer, Priority Interrupt, Direct Memory Access, Input Output Processor and Serial Communication. I/O Controllers, Asynchronous data transfer, Strobe Control, Handshaking.

Unit-5:

Process Organization: Basic Concept of 8-bit micro Processor (8085) and 16-bit Micro Processor (8086), Assembly Instruction Set, Assembly language program of (8085): Addition of two numbers, Subtraction, Block Transfer, find greatest number, Table search, Numeric Manipulation, Introductory Concept of pipeline, Flynn's and Feng's Classification, Parallel Architectural classification, Concept of Pipelining and Multi-Core Architecture.

Text and References Books

1. William Stalling, "Computer Organization & Architecture", Pearson education Asia
2. Mano Morris, "Computer System Architecture", PHI
3. Zaky & Hamacher, "Computer Organization", McGraw Hill
4. B. Ram, "Computer Fundamental Architecture & Organization",
5. New Age, A.S. Tannenbaum, "Structured Computer Organization", PHI.

Course Outcomes

1. Understand Number systems, Logic Gates, Boolean algebra, Design of Combinational and sequential circuits. (Understand)
2. Understand Von Neumann architecture, instruction cycle and the concept of Hardwired and Micro programmed control unit, addressing modes, register organization. (Understand)
3. Apply the concepts of memory organization in calculating hit-miss ratio and access time of magnetic disks. (Apply)
4. Understand the working of various I/O devices, buses, interrupt and interfaces etc. (Understand)
5. Understand the basics of pipelining and Multicore architecture. (Understand)
6. Design and implement systems using 8085 and 8086 microprocessor with the knowledge of pin diagram, interrupts and instruction format by writing assembly language programming. (Analyze)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	1	-	-	-	-	-	-	-	-	-
CO2	2	2	1	1	-	-	-	-	-	-	-	-
CO3	2	1	1	1	-	-	-	-	-	-	-	-
CO4	2	1	1	-	-	-	-	-	-	-	-	-
CO5	1	1	1	-	-	-	-	-	-	-	-	-
CO6	2	1	3	2	-	-	-	-	-	-	-	2

CO and PSO Mapping

CO/PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	-	2	-
CO2	2	-	1	-
CO3	2	2	1	-
CO4	2	-	1	-
CO5	2	-	1	-
CO6	2	2	2	1

INTERNET & JAVA PROFRAMMING (ECA-455)

Type	L	T	P	Credits
PCC	3	1	2	5

Prerequisite:

Course Content:

Unit-1

Introduction to Internet and Internet Services, HTML: Formatting tags, Links, Lists, Tables, Frames, Forms, Comments in HTML, DHTML

Core Java: Introduction, Operator, Data types, Variables, Arrays, Control Statements, Methods & Classes, Inheritance, Packages and Interface.

Unit-2

Core Java: Exception Handling, Multithread Programming, I/O, Applet, String handling, Networking, Event Handling, Introduction to AWT, AWT controls, Layout managers, Menus, Images, Graphics, JDBC

Unit-3

Java Servelets: Servelet Life Cycle, HTTP Servelet Class, Request Interface, Response Interface, Session Tracking (Cookies VRL)

JSP: Overview, Relation of Applet and Servelet with JSP, Scripting Element, JSP Expressions, JSP Scriplets, Predefined Variables, Creating Custom JSP Tag Libraries, Using Nested Tags, Structuring Generated Servelet in JSP Pages, Including Files and Applets in JSP Documents, Integrating Servelet and JSP.

Unit-4

Java Swing: Creating a Swing Applet and Applications, Programming using Panes, Pluggable Look and Feel, Labels, Text Fields, Buttons, Toggle Buttons, Checkboxes, Radio Buttons, View Port, Scroll Panes, Scroll Bars, Lists, Combo Box, Progress Bar, Menus and Toolbars, Layered Panes, Tabbed Panes, Split Panes, Layouts, Windows, Dialog Boxes, Inner Frame.

Unit-5

Java Beans: Application Builder Tools, The Bean Developer Kit (BDK), JAR Files, Introspection, Developing a Simple Bean, Using Bound Properties, The Java Bean API, Session Beans, Entity Beans, Introduction to Enterprise Java Beans (EJB), Introduction to Remote Method Invocation (RMI): A Simple Client-Server Application using RMI.

Lab Work

1. Design a HTML page to display your CV
2. Design a HTML form to reserve a railway ticket.
3. Write a Java Script program that finds the greatest common divisor of two numbers.
4. In the form mentioned in problem 2 to reserve a railway ticket add the following validations using java script.
   - From city and to city are two different cities.
   - Age of passengers should not be greater than 150.
   - Name of the passenger should be a string of a maximum length 20.
5. Write a program for illustrating client/server side scripting with help of ASP.
6. Write a piece of code in XML for creating DTD, which specifies set of rules.
7. Create style sheet in CSS/XSL and display the document in Internet Explorer.

Text and Reference Books

1. Margaret Levine Young, "The Complete Reference Internet", TMH
2. Naughton Schildt, "The Complete Reference JAVA2", TMH
3. Balagurusamy E, "Programming in JAVA", TMH
4. Dustin R. Callway, "Inside Servlets", Addison Wesley
5. Mark Wutica, "Java Enterprise Edition", QUE
6. Steven Holzner, "Java2 Black Book", Dreamtech Media

Course Outcomes

1. Understand the basics of web and apply the web concepts for web application development. (Apply)
2. Understand, apply and analyze mark-up languages like HTML, DHTML, and XML for development of different web applications. (Apply, Analyze)
3. Develop interactive web applications using client-side scripting languages. (Apply)
4. Develop and deploy web services to build the server side components in web applications. (Apply)
5. Understand and develop applications using EJB and RMI concepts. (Understand, Apply)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2	-	-	-	-	-	-	-	-	-
CO2	2	2	2	-	2	-	-	-	-	-	-	-
CO3	2	1	3	1	2	-	-	-	-	-	-	-
CO4	2	1	3	1	2	-	-	-	-	-	-	-
CO5	2	1	3	1	2	-	-	-	-	-	-	-

CO and PSO Mapping

CO/PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	-
CO2	3	2	3	-
CO3	3	2	3	1
CO4	3	2	3	1
CO5	3	2	3	1

OPERATING SYSTEMS (ECA-457)

Type	L	T	P	Credits
PCC	3	1	0	4

Prerequisite:

Course Content:

Unit-1:

Introduction: Operating System and its functions, Evolution of Operating System, Batch, Interactive, Time Sharing and Real Time Operating System, System Protection. Operating System Structure: System Components, System Structure, Operating System Services.

Unit-2:

Process Management: Process Concept, Process State, Process Control Block, Threads. Concurrent Processes: Principle of Concurrency, Mutual Exclusion, Inter Processes Communication, Critical Section Problem, Semaphores, Classical Problems in Concurrency, Producer / Consumer Problem, Readers-Writers Problem, Dining Philosophers Problem.

Unit-3:

CPU Scheduling: Scheduling Concept, Scheduling Techniques, Performance Criteria for Scheduling Algorithm, Evolution, Multiprocessor Scheduling. Deadlock: System Model, Deadlock Characterization, Prevention, Avoidance and Detection, Recovery From Deadlock Combined Approach.

Unit-4:

Memory Management: Basic Machine, Resident Monitor, Multiprogramming with Fixed Partition, Multiprogramming With Variable Partition, Multiple Base Register, Paging, Segmentation, Paged Segmentation. Virtual Memory: Virtual Memory Concept, Demand Paging, Performance, Paged Replaced Algorithm, Allocation of Frames, Thrashing, Cache Memory Organization, Impact on Performance.

Unit-5:

I/O Management & Disk Scheduling: I/O Devices, Organization of I/O Function, Operating System Design Issues, I/O Buffering, Disk Scheduling (FCFS, SCAN, CSCAN). File Management: File Concept, File Organization, File Directories, File Sharing, Allocation Methods, Free Space Management, and Implementation Issues.

Text and References Books:

1. Milenekovik, "Operating System Concept", McGraw Hill.
2. Petersons, "Operating Systems", Addison Wesley.
3. Dietal, "An Introduction to Operating System", Addison Wesley.
4. Tannenbaum, "Operating System Design and Implementation", PHI.
5. Gary Nutt, "Operating System, A Modern Perspective", Addison Wesley.
6. Stalling, Williams, "Operating System", Maxwell Macmillan
7. Silveschatz, Peterson J., "Operating System Concepts", Willey.
8. Crowley, "Operating System", TMH.

Course Outcomes:

1. Understand types and structure of operating systems. (Understand)
2. Construct solutions for problems related to process scheduling, deadlocks and synchronization in a multi-programmed operating system. (Apply)
3. Develop appropriate solutions for memory management considering challenges due to multi-programming and virtual memory. (Apply)
4. Apply knowledge of various software and hardware synchronization tools for solving critical section problem in concurrent processes. (Apply)
5. Construct solutions for problems related to secondary storage management with an understanding of file systems and disk scheduling. (Apply)
6. Design various system protection and security mechanisms in order to design efficient software system. (Apply)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	-	-	-	-	-	-	-	-	-	-	-
CO2	2	2	3	-	-	-	-	-	-	-	-	-
CO3	2	2	3	-	-	-	-	-	-	-	-	-
CO4	2	2	3	1	2	-	-	-	-	-	-	2
CO5	2	2	3	-	-	-	-	-	-	-	-	-
CO6	2	2	3	1	-	-	-	-	-	-	-	2

CO and PSO Mapping

CO/PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	-	-	-
CO2	3	3	2	-

CO3	3	3	2	-
CO4	3	3	3	1
CO5	3	2	2	-
CO6	3	2	2	1

DISCRETE MATHEMATICAL STRUCTURES (BMA-451)

Type	L	T	P	Credits
BSC	3	0	0	3

Prerequisite:

Course Content:

UNIT I: Fundamentals of Logic

Propositional Logic: Propositions, Basic logic operations and truth tables, Tautologies, Contradictions, Contingency, Algebra of propositions, Logical equivalence: the laws of logic, Logical implication: Rules of inference, Logical analysis of arguments, Some computing applications (Normal forms), Functionally complete set of operations, Formal proofs.

First Order Logic: Predicates & quantifiers, Nested quantifiers, Use of quantifiers, Rules of inference, Validity of arguments.

Notion of Proofs: Proof by counter example, the contraposition, proof by contradiction, inductive proofs.

UNIT II: Set Theory, Relations and Functions

Set Theory: sets & subsets, Venn diagrams, Set operations and laws, countable and uncountable sets, Cartesian product, Cardinality, Principle of inclusion- exclusion.

Relations: Relation, Representation & properties, n-ray relations and applications, Composition of relations, Closures of relations, Equivalence relation & partitions, partial orders, compatibility relation.

Functions: Functions and its types, Inverse function, Composition of functions, Special functions, Recursively defined functions, Computational Complexity, Analysis of algorithms.

Theorem Proving Techniques: Mathematical induction, strong induction, and well ordering, structural induction, Pigeonhole principle.

UNIT III: Algebraic Structures and Coding Theory

Algebraic Structures: Definition, Properties, Semi group, Monoid, Group, Properties of groups, Subgroup, Cyclic group, Cosets and Lagrange's theorem, Permutation groups, Normal subgroup, Homomorphism and isomorphism of groups, Congruence relation, Rings and Fields. Examples and standard results.

Coding Theory: Elements of coding theory, Hamming matric, Parity-check and generator matrices, Coding and error detection, Group codes: decoding with coset leaders and error correction, Hamming matrices.

UNIT IV: Partially Ordered Structures

Posets: Definitions, ordered set, Hasse diagram, isomorphic ordered set, well ordered set, Minimal and Maximal elements, LUB & GLB etc.

Lattices: Definition & Properties, Product Lattices, Isomorphic Lattices, Applications, Types of Lattices.

Boolean Algebras: Definitions & Properties, SOP & POS forms, Logic gates and minimization of circuits, Karnaugh maps, Quine-McClusky method.

Trees: Definition & Examples and Properties, Rooted tree, Binary tree, Tree traversal, application in computer science and engineering.

UNIT V: Combinatorics and Graph Theory:

Combinatorics: Basic counting techniques, Discrete numeric functions and properties, Recurrence relations and their applications (modelling), various methods of solutions, system of recurrence relations, OGF & EGF, properties, applications: solution of recurrence relations and combinatorial problems. Polya's enumeration theorem and applications.

Graphs: Graphs and graph models, terminology, matrices associated with graphs, Isomorphism, Special types of graphs, connectedness, Euler and Hamilton graphs with their applications, trees with properties, MST, planer graphs and applications, criteria of planarity, Graph coloring and coloring models, directed graphs.

Text and Reference Books:

1. Trembley, J.P. & R. Manohar, "Discrete Mathematical Structures with applications to Computer Science", McGraw Hill.
2. Kenneth H. Rosen, "Discrete Mathematics and its Applications", McGraw Hill.
3. Ralph, P. Garimaldi, "Discrete& Combinatorial Mathematics" Pearson Publication, Asia.
4. Deo, narsingh, "Graph Theory with applications to Engineering & Computer Science", PHI.
5. Krishnamurthy, V., "Combinatorics Theory & Application", East-West Press Pvt. Ltd., New Delhi.

Course Outcomes:

1. Understand concepts of Logic and various inference mechanisms using logic. (Understand)
2. Understand Set theory, functions, relations and the concepts of theorem proving. (Understand)
3. Explain algebraic structure and coding theory. (Understand)
4. Understand and apply concepts of partially ordered structures, Boolean algebra and trees in various application of computer science domain. (Understand, Apply)
5. Understand and apply graph theory and concepts of recurrence relation in system modeling. (Understand, Apply)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	3	3	2	3	1	2	-	3	3
CO2	3	3	3	3	3	2	3	1	2	-	3	3
CO3	3	3	3	3	3	2	3	1	3	-	3	3
CO4	3	3	3	3	3	2	3	1	3	-	3	3
CO5	3	3	3	3	3	2	3	-	3	-	3	3

FUNDAMENTAL OF MANAGEMENT (HSS-454)

Type	L	T	P	Credits
HSMC	3	0	2	4

Prerequisite:

Course Content:

Unit 1: Principles of Management
Introduction, Functions of Management, Planning, Organising, Directing, Controlling, Coordinating

Unit 2: People Management
Introduction, Functions, Human Resource Planning, Recruitment, Selection, Performance Appraisal, Training, Salary Management

Unit 3: Finance and Marketing Management
Functions of Finance Management, Financing, Investment and Dividend Decisions; Functions of Marketing Management, E-retailing, Interactive advertising and IT

Unit 4: Behavioural Management
Organisational Structure, Motivation, Leadership, Team Management, Change Management, Conflict, Stress Management

Unit 5: Business Communication
Fundamentals of communication, Elements of written communication, Business letters, Technical Reports, Business Presentations, Listening.

Text Books

1. Robbins, S. P., Management, Prentice hall of India, 1998
2. Newstrom,J. , Organisational Behaviour: Human Behaviour at Work, McGraw Hill Education
3. L. M. Prasad, Management,
4. Business Communication, Monipally, Tata McGraw-Hill Publication

Reference Books

1. Fred Luthans, 'Organizational Behaviour', McGraw Hill Education, Asia, 2007.
2. Mamoria, C.B., Personnel Management, Himalayan Publishing, India
3. Dwivedi, R S , 'Human Relations and Organizational Behaviour: a Global Perspective', Macmillan India Ltd., Delhi
4. Krishna Mohan & Meera Banerjee , Developing Communication Skills, Macmillan India

Course Outcomes (CO)

1. Understand the fundamental principles of management.
2. Learn people management in an organization.
3. Learn basics of finance and marketing concept of businesses
4. Understand human behavior and the role in organizational context
5. Communicate their ideas in the contemporary global competitive environment effectively.

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	1	-	1	1	1	2	2	1	3
CO2	-	-	-	1	-	1	1	3	3	3	1	3
CO3	-	-	-	1	-	2	1	3	3	3	2	3
CO4	-	-	-	2	-	2	2	3	3	3	2	3
CO5	-	-	-	2	-	1	2	2	2	2	2	3

SOFTWARE ENGINEERING (ECA-452)

Type	L	T	P	Credits
PCC	3	1	0	4

Prerequisite:

Course Content:

Unit-1:

Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, Selection of Software Development Models,

Unit-2:

Software Requirement Specifications (SRS) Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modeling, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS, Estimation of various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.

Unit-3:

Software Design Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.

Unit-4:

Software Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing

(Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards.

Unit-5:

Software Maintenance: Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re-Engineering, Reverse Engineering, Software Configuration Management Activities, Change Control Process, Software Version Control, Defect Detection and Removal: Defect Amplification Model, An Overview of CASE Tools.

Lab Work:

Lab exercises or a Mini Project (as per list given below) to be carried out using languages like C++, Java, C# and tools like Visio, ARGOUML, Rational Rose etc. Design and Implementation of an Object based application using any one of the above languages/tools is desirable.

- Hotel Automation System
- Book Shop Automation Software
- Word processing Software
- Software Component Cataloguing Software
- Payroll System
- Banking System
- Purchase Order System
- Library Management System
- Railway Reservation System
- Bill Tracking System
- University Admission System
- Estate Management System.

Text and References Books:

1. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.
3. Ian Sommerville, Software Engineering, Addison Wesley.
4. Pankaj Jalote, Software Engineering, Narosa Publication
5. Pfleeger, Software Engineering, Macmillan Publication.

Course Outcomes:

1. Understand and explain various concepts of software engineering and software life cycle development models. (Understand)
2. Prepare SRS and Compute cost and effort required to complete a given project, using various estimation techniques and models. (Apply)

3. Understand various concepts of Software design and Construct Data Flow Diagrams, Data Dictionaries and UML diagrams for a given software requirement specification. (Understand, Apply)
4. Understand various testing techniques and use these concepts to design optimal test cases. (Understand, Apply, Analyze)
5. Understand software configuration management, version control, reverse engineering, defect tracking etc. (Understand)
6. Build a project report as a team which contains the requirement specification, plan, schedule and design documents based on the knowledge of software development lifecycle. (Apply)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	2	-	-	-	-	-	-	-	-	1
CO2	2	2	1	1	1	-	-	-	-	-	3	1
CO3	2	2	3	1	3	1	1	1	3	3	3	1
CO4	2	2	1	-	1	-	-	-	-	-	3	1
CO5	2	2	1	-	1	-	-	-	-	-	3	1
CO6	3	3	3	3	3	1	1	1	3	3	3	-

CO and PSO Mapping

CO/PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	3	1	1
CO2	3	3	1	2
CO3	3	3	1	2
CO4	3	3	1	1
CO5	2	2	-	1
CO6	3	3	2	2

DESIGN & ANALYSIS OF ALGORITHMS (ECA-454)

Type	L	T	P	Credits
PCC	3	1	0	4

Prerequisite:

Course Content:

Unit-1:
Algorithms definition and introduction, Analysis of algorithms, Growth of Functions, Master's Theorem, Designing of Algorithms, Partitioning Algorithms, Divide and Conquer design and analysis techniques: Merge Sort and Quick Sort, Sorting and order Statistics: Heap sort, Sorting in linear time, Medians and Order Statistics.

Unit-2:
Advanced Data Structures: Introduction of Red-Black Trees, Augmenting Data Structure, B-Trees, Binomial Heaps, Fibonacci Heaps, Data Structure for Disjoint Sets, Amortized Analysis.

Unit-3:
Advanced Design and Analysis Techniques: Dynamic Programming, Greedy Algorithms, Back Tracking, Branch and Bound with their applications.

Unit-4:
Graph Algorithms: Elementary Graphs Algorithms, Minimum Spanning Trees, Single-source Shortest Paths, All-Pairs Shortest Paths, Traveling Salesman Problem and Maximum Flow

Unit-5:
Selected Topics: Randomized Algorithms, String Matching, Non-deterministic Algorithms: P, NP, NP Hard and NP Completeness, Approximation Algorithms, PRAM Algorithms.

Text and References Books:

1. Coreman, Rivest, Lisserson: “Algorithm", PHI.
2. Basse, "Computer Algorithms: Introduction to Design & Analysis", Addison Wesley.
3. Horowitz & Sahni, "Fundamental of Computer Algorithm", Universities Press

Course Outcomes:

1. Understand and apply mathematical preliminaries to the analysis and design stages of different types of algorithms. (Understand, Apply)
2. Analyze worst-case time complexity of various algorithms using asymptotic methods. (Analyze)
3. Understand and apply the divide-and-conquer paradigm and synthesize divide-and-conquer algorithms on problems of Sorting, Searching, finding MST etc. (Understand, Apply)
4. Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms. (Apply, Analyze)
5. Apply the dynamic-programming paradigm to model engineering problems using graph and write the corresponding algorithm to solve the problems. (Apply)
6. Explain the ways to analyze randomized and approximation algorithms (Apply, Analyze)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	1	-	-	-	-	-	-	-	-	-
CO2	3	3	1	-	-	-	-	-	-	-	-	-
CO3	3	2	3	-	-	-	-	-	-	-	-	-
CO4	3	2	3	-	-	-	-	-	-	-	-	-
CO5	3	2	3	-	-	-	-	-	-	-	-	-
CO6	2	2	-	-	-	-	-	-	-	-	-	-

CO and PSO Mapping

CO/PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	-	1
CO2	2	1	-	1
CO3	2	2	1	1
CO4	2	2	1	1
CO5	2	2	1	1
CO6	2	1	-	2

DATABASE MANAGEMENT SYSTEMS (ECA-456)

Type	L	T	P	Credits
PCC	3	1	2	5

Prerequisite:

Course Content:

Unit-1:

Introduction: An overview of database management system, database system Vs file system, Database system concepts and architecture, data models schema and instances, data independence and database language and interfaces, Data definitions language, DML, Overall Database Structure. Data Modeling using the Entity Relationship Model: ER model concepts, notation for ER diagram, mapping constraints, keys, Concepts of Super Key, candidate key, primary key, Generalization, aggregation, reduction of an ER diagrams to tables, extended ER model, relationships of higher degree.

Unit-2:

Relational data Model and Language: Relational data model concepts, integrity constraints: entity integrity, referential integrity, Keys constraints, Domain constraints, relational algebra, relational

calculus, tuple and domain calculus, Introduction to SQL: Characteristics of SQL, Advantage of SQL.SQL data types and literals, Types of SQL commands, SQL operators and their procedure, Tables, views and indexes. Queries and sub queries, Aggregate functions. Insert, update and delete operations, Joins, Unions, Intersection, Minus, Cursors in SQL.

Unit-3:

Data Base Design & Normalization: Functional dependencies, normal forms, first, second, third normal forms, BCNF, inclusion dependences, loss less join decompositions, normalization using FD, MVD, and JDs, alternative approaches to database design.

Unit-4:

Transaction Processing Concepts: Transaction system, Testing of serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures, log based recovery, checkpoints, deadlock handling.

Unit-5:

Concurrency Control Techniques: Concurrency control, locking Techniques for concurrency control, Time stamping protocols for concurrency control, validation based protocol, multiple granularity, Multi version schemes, Recovery with concurrent transaction.

Lab Work:

1. Write the queries for Data Definition and Data Manipulation language.
2. Write SQL queries using Logical operators (=, <, >, etc.).
3. Write SQL queries using SQL operators (Between…. AND, IN(List), Like, ISNULL and with negating expressions).
4. Write SQL query using character, number, date and group functions.
5. Write SQL queries for Relational Algebra (UNION, INTERSECT, and MINUS, etc.).
6. Write SQL queries for extracting data from more than one table (Equi-Join, Non-Equi-Join, Outer Join)
7. Write SQL queries for sub queries, nested queries.
8. Write programs by the use of PL/SQL.
9. Concepts for ROLL BACK, COMMIT & CHECK POINTS.
10. Create VIEWS, CURSORS, and TRIGGRS & write ASSERTIONS.
11. Create FORMS and REPORTS.

*Students are advised to use **Developer 2000/Oracle-9i** version or other latest version for above listed experiments. However depending upon the availability of software's, students may use **Power Builder /SQL SERVER**. Students may also work on a Mini Project to understand the important concepts of Database.

Text and References Books:

1. Date C J, "An Introduction to Database System", Addision Wesley
2. Korth, Silbertz, Sudarshan, "Database Concepts", McGraw Hill
3. Elmasri, Navathe, "Fundamentals of Database Systems", Addision Wesley
4. Leon & Leon, "Database Management System", Vikas Publishing House.
5. Bipin C. Desai, "An introduction to Database Systems", Galgotia Publication
6. Majumdar & Bhattacharya, "Database Management System", TMH
7. Ramakrishnan, Gehrke, "Database Management System", McGraw Hill
8. Kroenke, "Database Processing: Fundamentals, Design and Implementation", Pearson Education.

9. Maheshwari Jain, “DBMS: Complete Practical Approach”, Firewall Media, New Delhi.

Course Outcomes

1. Understand and Develop Entity Relationship (ER) and Relational Models for a given application. (Understand, Apply)
2. Develop and manipulate relational database using Structured Query Language and relational languages. (Apply)
3. Develop a normalized database for a given application by incorporating various constraints like integrity and value constraints. (Apply)
4. Understand and apply transaction processing concepts and convert schedules to serializable schedules. (Understand, Apply)
5. Illustrate different concurrency control mechanisms to preserve data consistency in a multi-user environment. (Apply)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	-	-	-	-	-	-	-	-
CO2	3	2	2	-	-	-	-	-	-	-	-	-
CO3	3	3	2	-	-	-	-	-	-	-	-	-
CO4	3	3	2	-	-	-	-	-	-	-	-	-
CO5	3	3	2	-	-	-	-	-	-	-	-	-

CO and PSO Mapping

CO/PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	2
CO2	2	2	-	1
CO3	2	2	-	1
CO4	2	3	-	1
CO5	2	3	-	1

DATA STRUCTURE USING C (ECA-458)

Type	L	T	P	Credits
PCC	3	1	2	5

Prerequisite: Computer Concepts & Programming in 'C' (ECA-451)

Course Content:

Unit -1:

Introduction: Basic Terminology, Elementary Data Organization, Structure operations, Algorithm Complexity and Time-Space trade-off.

Arrays: Array Definition, Representation and Analysis, Single and Multidimensional Arrays, address calculation, application of arrays, Character String in C, Character string operation, Array as Parameters, Ordered List, Sparse Matrices and Vectors.

Stacks: Array Representation and Implementation of stack, Operations on Stacks: Push & Pop, Array Representation of Stack, Linked Representation of Stack, Operations associated with Stacks, Applications of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack, Applications of recursion in problems like 'Tower of Hanoi'.

Unit-2:

Queues: Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty, Circular queues, D-queues and Priority Queues**.**

Linked list: Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly linked list, Linked List in Array, Polynomial representation and addition, Generalized linked list, Garbage Collection and Compaction.

Unit-3:

Trees: Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree, Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees, Traversing Threaded Binary trees, Huffman algorithm**.**

Searching and Hashing: Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation.

Unit-4:

Sorting: Insertion Sort, Bubble Sort, Quick Sort, Two Way Merge Sort, and Heap Sort, Sorting on Different Keys, Practical consideration for Internal Sorting**.**

Binary Search Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees, B-trees.

Unit-5:

Graphs: Terminology & Representations, Graphs & Multi-graphs, Directed Graphs, Sequential Representations of Graphs, Adjacency Matrices, Traversal, Connected Component and Spanning Trees, Minimum Cost Spanning Trees.

File Structures: Physical Storage Media File Organization, Organization of records into Blocks, Sequential Files, Indexing and Hashing, Primary indices, Secondary indices, B+ Tree index Files, B Tree index Files, Indexing and Hashing Comparisons.

Text and Reference Books:

1. Horowitz and Sahani, "Fundamentals of data Structures", Galgotia Publication Pvt. Ltd., New Delhi.
2. R. Kruse et. al, "Data Structures and Program Design in C", Pearson Education Asia, Delhi-2002
3. A. M. Tenenbaum, "Data Structures using C & C++", Prentice-Hall of India Pvt. Ltd., New Delhi.
4. K Loudon, "Mastering Algorithms with C", Shroff Publisher & Distributors Pvt. Ltd.
5. Bruno R Preiss, "Data Structures and Algorithms with Object Oriented Design Pattern in C++", Jhon Wiley & Sons, Inc.
6. Adam Drozdek, "Data Structures and Algorithms in C++", Thomson Asia Pvt. Ltd.(Singapore)

Lab Work:

Write Program in C or C++ for the following

1. Array implementation of Stack, Queue, Circular Queue, List.
2. Implementation of Stack, Queue, Circular Queue, List using Dynamic memory Allocation.
3. Implementation of Tree Structures, Binary Tree, Tree Traversal, Binary Search Tree, Insertion and Deletion in BST.
4. Implementation of Searching and Sorting Algorithms.
5. Graph Implementation, BFS, DFS, Min. cost spanning tree, shortest path algorithm.

Course Outcomes:

1. Analyze the algorithms to determine the time and computation complexity and justify the correctness. (Analyze)
2. Implement Arrays, Stacks, Queues and linked list based problems and analyze the algorithm to determine the time complexity. (Apply, Analyze)
3. Implement search and traversal algorithms on Trees and Graphs and determine the time complexity. (Apply, Analyze)
4. Algorithms for Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort and compare their performance in term of space and time complexity. (Apply, Analyze, Evaluate)
5. Understand file structures and file handling. (Understand)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	1	-	-	-	-	-	-	-	-	-
CO2	2	1	-	-	-	-	-	-	-	-	-	-
CO3	3	3	2	-	-	-	-	-	-	-	-	-
CO4	3	3	2	-	-	-	-	-	-	-	-	-
CO5	2	-	-	-	-	-	-	-	-	-	-	-

CO and PSO Mapping

CO/PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	1	1
CO2	3	2	1	1
CO3	3	2	1	1
CO4	3	2	1	1
CO5	2	1	-	-

COMPUTER GRAPHICS & ANIMATION (ECA-460)

Type	L	T	P	Credits
PCC	3	0	2	4

Prerequisite: NIL

Course Content:

Unit-1:
Line generation: Points and Lines, Planes, Pixels and Frame buffers, vector and character generation. Graphics Primitives: Display devices, Primitive devices, Display File Structure, Display control text, Line-drawing Algorithms**:** DDA Algorithm Bresenham's line Algorithm, Circle-generating Algorithm**:** Midpoint Circle of Algorithm, Polygon Filling Algorithm.

Unit-2:
2-D Viewing and Clipping: Point Clipping, Line Clipping, Cohen-Sutherland Line Clippings, Cyrus-Beck Line Clipping Algorithm, Polygon Clipping: Sutherland Hodgman Algorithm, Polygon: Polygon Representation, Entering polygons, Filling polygons, Segments: Segments table, Creating deleting and renaming segments, Visibility.

Unit-3:
2-D and 3-D Transformations: Basic Transformations: Translation, Rotation, Scaling, Shear, Composite Transformations: Rotations about a point, Reflection about a line, Homogeneous Coordinate Systems, 3-D Transformations, 3-D geometry primitives, Viewing Transformation, Projections: Parallel Projection, Orthographic & Oblique Projections, Perspective Projections. Interaction: Hardware input devices handling algorithms, Event handling echoing, Interactive techniques.

Unit-4:
Hidden Line and Surface: Back face removal algorithms, hidden line methods, Rendering and Illumination: Introduction to curve and Surfaces generation, Bezier, Hermite and B-spline algorithms and their comparisons.

Unit-5:
Multimedia and Animation: Basic of Animation, Types of Animation, Simulating, Accelerations, Computer Animation Tools, Multimedia Applications, Concepts of Hypertext/Hypermedia, Images, Audio and Video**,** Multimedia Tools.

Text and Reference Books:

1. Rogers, "Procedural Elements of Computer Graphics", McGraw Hill
2. Baker and Hearn, "Computer Graphics", PHI Publication.
3. Newman and Sproul, "Principle of Interactive Computer Graphics", McGraw Hill
4. Steven Harrington, "Computer Graphics", A Programming Approach, 2nd Edition
5. Rogar and Adams, "Mathematical Elements of Computer Graphics", McGraw Hill.

Lab Work:

Write Program in C or C++ for the following.

1. Implementation of line generation using slope's method, DDA and Bresenham's algorithms.
2. Implementation of circle generation using Mid-point method and Bresenham's algorithm.
3. Implementation of ellipse generation using Mid-point method.
4. Implementation of polygon filling using Flood-fill, Boundary-fill and Scan-line algorithms.
5. Implementation of 2-D transformation: Translation, Scaling, Rotation, Mirror Reflection and Shearing (write a menu driven program).
6. Implementation of Line Clipping using Cohen-Sutherland algorithm and Bisection Method.
7. Implementation of Polygon Clipping using Sutherland-Hodgman algorithm.
8. Implementation of 3-D geometric transformations: Translation, Scaling and rotation.
9. Implementation of Curve generation using Interpolation methods.
10. Implementation of Curve generation using B-spline and Bezier curves.
11. Implementation of any one of Back face removal algorithms: Depth-Buffer algorithm, Painter's algorithm, Warnock's algorithm, Scan-line algorithm.

Course Outcomes:

1. Understand and use various mathematical concepts and supporting composite 2-D & 3-D graphics transformations for hidden surface detection/ removal and various graphical algorithms. (Understand, Apply)
2. Design algorithms for various graphics shapes like ellipse, hyperbola, triangle etc. (Apply)
3. Use of various graphical tools and software in 3D Graphics API (e.g. OpenGL or DirectX). (Apply)
4. Understand and apply geometrical transformation and computer graphics in multidisciplinary field of engineering. (Apply)
5. Understand the hardware system architecture for computer graphics - graphics pipeline, frame buffers, and graphic accelerators/co-processors. (Understand)
6. Analyze and implement interactive graphics applications using programming language and graphics application programming interfaces. (Apply, Analyze)

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	-	-	-	-	-	-	-	-	-
CO2	2	1	3	-	-	-	-	-	-	-	-	-
CO3	2	2	3	-	3	-	-	-	-	-	-	-
CO4	2	1	1	2	-	2	-	-	-	-	-	2
CO5	2	1	-	-	-	-	-	-	-	-	-	-
CO6	2	2	3	2	-	-	-	-	2	-	-	-

CO and PSO Mapping

CO/PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	-
CO2	2	3	1	-
CO3	2	3	3	-
CO4	2	3	3	2
CO5	2	1	-	-
CO6	2	3	3	1

OPERATIONS RESEARCH (BMA-452)

Type	L	T	P	Credits
BSC	3	0	0	3

Prerequisite: NIL

Course Content:

Unit-1: Linear Programming Problems (LPP)

OR model, Formulation of LPP. model, Graphical LPP solution and sensitivity analysis, simplex method, M-method, Two-phase method, Special cases in simplex method application, Duality theory, Dual simplex method, Revised simplex method, Degeneracy, Sensitivity analysis, Various industrial application of LP.

Unit-2: Transportation Models, Assignment Models and Integer Programming

Formulation and Optimal solution of transportation models, Assignment models, Trans shipment models, Degeneracy in TP model, Industrial application, Formulation and Solution of integer linear

programming problems; Cutting-plane algorithm, Branch and Bound algorithm, 0-1 ILPP, applications, Knapsack problem, facility-location problem.

Unit-3: Sequencing and Scheduling Model

Sequencing problems- Travelling salesman problem, Machine-scheduling problem (Job shop), Network based planning models, Objectives of CPM and PERT, Characteristics of CPM/PERT projects, Network diagram, Terminology, Critical path, Project duration, PERT Network, Activity time, Probabilities of project completion, Optimal crashing of project activities.

Unit-4: Replacement and Inventory models

Replacement Problems: Optimal age of equipment replacement, capital equipment discounting cost, Replacement of items that fail, Individual and group replacement policies.

Inventory Models: Deterministic inventory models, Classic EOQ model, EOQ with price breaks, Multi-term, stochastic inventory models under probabilistic demand and lead times.

Unit-5: Dynamic Programming and Genetic Algorithms

Dynamic programming: Bellman's principle of optimality, computations in DP, Forward and Backward recursions, Dynamic Programming formulations, Investment problem, General allocation problem, Storage coach problem, Production scheduling.

Genetic Algorithms: Working principles, similarities and differences between Gas and Traditional methods, Gas for constrained optimization, Applications of Gas to solve simple problems.

Text and Reference Books:

1. S. S. Rao, "Optimization: Theory and Applications" Willey Eastern Limited.
2. H.A. Taha, "Operations Research- AN Introduction", Macmillan.
3. Hiller, F. S., G.J. Lieberman, "Introduction to Operations Research", Hoiden-Day.
4. Kalyanmoy Deb, "Optimization for Engineering Design: Algorithms & Examples "Prentice-Hall of India.
2. B. E. Gillet, Introduction Operations Research- A Computer Oriented Algorithmic Approach, McGraw Hill 1989.

Course Outcomes:

1. Understand and solve Linear Programming Problems.
2. Formulate and solve Transportation Models, Assignment Models and Integer Linear Programming Problems.
3. Formulate and solve Sequencing and Scheduling Models.
4. Formulate and solve Replacement and Inventory Models.
5. Learn and use Dynamic Programming and Genetic Algorithms.

CO and PO Mapping

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	2	3	2	2	-	1	2	2	3
CO2	3	3	3	2	3	2	2	-	1	2	2	3
CO3	3	3	3	3	2	2	2	-	1	2	2	3
CO4	3	3	3	3	2	2	2	-	1	2	2	3