

GENERAL COURSE STRUCTURE AND CREDIT DISTRIBUTION

CREDIT DISTRIBUTION

POSTGRADUATE PROGRAMME

**Scheme of Programme Master of Computer Applications (MCA) Integrated
(Scheme PG A1: Postgraduate Programmes (Course work only))**

Semester 1

Course Code	Course Title	Course ID	L	T	P	Credits	TI	TE	PI	PE	Total
Discipline Specific Courses (DSC)											
CC1	Fundamentals of Computer Programming	242/MCA/CC101	3	-	2	4	25	50	5	20	100
CC2	Computer Organization and Architecture	242/MCA/CC102	3	-	2	4	25	50	5	20	100
MIC/Vocational Courses (VOC)											
MIC1	One from pool	242/MCA/VO101	2	-	4	4	15	35	15	35	100
Multidisciplinary Courses (MDC)											
MDC1	One from pool	242/MCA/MD101	2	1	-	3	25	50	-	-	75
Ability Enhancement Course (AEC)											
AEC1	One from pool	242/MCA/AE101	2	0	0	2	15	35	-	-	50
Skill Enhancement Course (SEC)											
SEC1	One from pool	242/MCA/SE101	2	-	2	3	15	35	5	20	75
Value Addition Course (VAC)											
VAC1	One from pool	242/MCA/VA101	2	-	-	2	15	35	-	-	50
Total Credits						22	Total Marks			550	

Semester 2

Course Code	Course Title	Course ID	L	T	P	Credits	TI	TE	PI	PE	Total
Discipline Specific Courses (DSC)											
CC3	Data Structure and Algorithm	242/MCA/CC201	3	-	2	4	25	50	5	20	100
CC4	Operating System	242/MCA/CC202	3	-	2	4	25	50	5	20	100
MIC/Vocational Courses (VOC)											
MIC2	One from pool	242/MCA/VO201	2	-	2	4	15	35	15	35	100
Multidisciplinary Courses (MDC)											
MDC2	One from pool	242/MCA/MD201	2	1	-	3	25	50	-	-	75
Ability Enhancement Course (AEC)											
AEC2	One from pool	242/MCA/AEC201	2	-	-	2	15	35	-	-	50
Skill Enhancement Course (SEC)											
SEC2	One from pool	242/MCA/SE201	2	-	1	3	15	35	5	20	75
Value Addition Course (VAC)											
VAC2	One from pool	242/MCA/VA201	2	-	-	2	15	35	-	-	50
Total Credits						22	Total Marks			550	

Semester 1

Course code	CC1			
Category	Discipline Specific Courses (DSC)			
Course title	Fundamentals of Computer Programming			
Course ID	242/MCA/CC101			
Scheme and Credits	L	T	P	Credits
	3	-	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 hrs			

Note: Examiner will be required to set NINE questions with all questions carrying equal marks. Question Number 1, covering the entire syllabus, will be compulsory. Examiner will set two questions from each Unit with internal choice. Student will be required to attempt FIVE questions in all, selection one question from every unit apart from the Question Number 1.

Course outcomes:

- CO1: Learn the functional units and classify types of computers, their applications and effects.
- CO2: Understand system software's and their working.
- CO3: Understand the logic building used in programming.
- CO4: Design and develop algorithms for solving various real-life problems.
- CO5: Design and develop programs using C.

UNIT-I

Computer Fundamentals: Concept of data and information. Components of Computer. Input and Output Device, Components of CPU, Memory and Storage Devices, Classification of Computers, Advantages and Limitations of Computer, Applications of Computer, Social concerns of Computer Technology: Positive and Negative Impacts, Computer Crimes, Viruses and their remedial solutions.

Computer Software: System and Application Software, Overview of Operating System Programming Languages Machine. Assembly. High Level Language, 4GL. Language Translator, Linker and Loader.

UNIT-II

Problem Solving: Problem Identification. Analysis, Algorithms, Flowcharts. Pseudo codes. Decision Tables. Program Coding. Program Testing and Execution.

C Programming Fundamentals: Keywords, Variables and Constants, Structure of a C program.

UNIT-III

Operators & Expressions: Arithmetic, Unary, Logical. Bit-wise, Assignment & Conditional Operators.

Decision Making: Decision making using if...else. Else If Ladder; Switch, break. Continue and Goto statements.

UNIT-IV

Loops: Looping using while, do...while, for statements. Nested loops.

Functions: Defining & Accessing User defined functions. Library Functions, Function Prototype, Passing Arguments, Passing array as argument. Recursion, Use of Library Functions. Macro vs. Functions, Pointers in C.

Textbooks & Reference Books:

1. E. Balaguruswamy: Programming in C. Tata McGraw Hill.
2. Rajender Singh Chhillar: Application of IT to Business, Ramesh Publishers, Jaipur.
3. Gill Nasib Singh: Computing Fundamentals and Programming in C, Khanna Books Publishing Co., New Delhi.

Fundamentals of Computer Programming

List of Experiments

1. Write a C program to compute roots of quadratic equation $ax^2+bx+c=0$, where a, b, and c are three coefficients of a quadratic equation are inputs.
2. Design and develop an algorithm to find the reverse of an integer number.
3. Design and develop an algorithm to check whether given number is PALINDROME or NOT. Implement a C program for the developed algorithm that takes an integer number as input and output the reverse of the same with suitable messages. Ex: Num: 2019, Reverse: 9102, Not a Palindrome.
4. Design and develop a c program to implement simple calculator using switch case statement.
5. Develop, implement and execute a C program to search a Number in a list using linear searching Technique.
6. Develop an algorithm, implement and execute a C program that reads N integer numbers and arrange them in ascending order using Bubble Sort.
7. Design and develop a C program to read and print a matrix and check whether a given Matrix is a sparse Matrix or not.
8. Write a C program to implements the following string manipulation functions till the use wishes to continue (infinite loop): (i) strcpy() (ii) strlen() (iii) strrev () (iv) strcmp() (v) strcat(). b. Read a sentence and print frequency of vowels and total count of consonants.
9. Design and develop a C function RightRotate (x, n) that takes two integers x and n as input and returns value of the integer x rotated to the right by n positions. Assume the integers are unsigned.
10. Draw the flowchart and write a recursive C function to find the factorial of a number, n!, define by $fact(n)=1$, if $n=0$. Otherwise $fact(n) = n * fact(n-1)$. Using this function, write a C program to compute the binomial coefficient nCr . Tabulate the results for different values of n and r with suitable messages
11. a. Write a C program to maintain a record of n student details using an array of structures with four fields (Roll number, Name, Marks, and Grade). Assume appropriate data type for each field. Input & Print the members of the structure

- b. Write a C program to take 2 structures HH:MM:SS as T1 & T2 & display the time difference as structure as T3.
12. Write a C program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers.

Course code	CC2			
Category	Discipline Specific Courses (DSC)			
Course title	Computer Organization and Architecture			
Course ID	242/MCA/CC102			
Scheme and Credits	L	T	P	Credits
	3	-	2	4
Theory Internal	25			
Theory External	50			
Practical Internal	05			
Practical External	20			
Total	100			
Duration of Exam	3 hrs			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each unit and the marks of each question from Question no.2 to 9 will be of 20% of total marks of Question paper. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

Course outcomes:

- CO1. Understand the theory and architecture of central processing unit.
- CO2. Analyse some of the design issues in terms of speed, technology, cost, performance.
- CO3. Design a simple CPU with applying the theory concepts.
- CO4. Use appropriate tools to design verify and test the CPU architecture.
- CO5. Learn the concepts of parallel processing, pipelining and inter processor communication.

UNIT-1

Boolean algebra and logic gates: Boolean algebra and Logic gates, Combinational logic blocks(Adders, Multiplexers, Encoders, de-coder), Sequential logic blocks(Latches, Flip-Flops, Registers, Counters) Store program control concept, Flynn's classification of computers (SISD, MISD, MIMD); Multilevel viewpoint of a machine: digital logic, micro architecture, ISA, operating systems, high level language; structured organization; CPU, caches, main memory, secondary memory units & I/O; Performance metrics; MIPS, MFLOPS.

UNIT -2

Instruction set architecture: Instruction Set Architecture Instruction set based classification of processors (RISC, CISC, and their comparison); addressing modes: register, immediate, direct, indirect, indexed; Operations in the instruction set; Arithmetic and Logical, Data Transfer, Control Flow; Instruction set formats (fixed, variable, hybrid); Language of the machine: 8086 ; simulation using MSAM.

UNIT -3

Memory hierarchy: Basic non pipelined CPU Architecture and Memory Hierarchy & I/O Techniques CPU Architecture types (accumulator, register, stack, memory/ register) detailed data path of a typical register based CPU, Fetch-Decode-Execute cycle (typically 3 to 5 stage); microinstruction sequencing, implementation of control unit, Enhancing performance with pipelining. The need for a memory hierarchy (Locality of reference principle, Memory hierarchy in practice: Cache, main memory and secondary memory, Memory parameters: access/ cycle time, cost per bit); Main memory (Semiconductor RAM & ROM organization, memory expansion, Static & dynamic memory types); Cache memory (Associative & direct mapped cache organizations).

UNIT- 4

Parallelism and interrupts: Introduction to Parallelism and Computer Organization [80x86] Goals of parallelism (Exploitation of concurrency, throughput enhancement); Amdahl's law; Instruction level parallelism (pipelining, super scaling –basic features); Processor level parallelism (Multiprocessor systems overview). Instruction codes, computer register, computer instructions, timing and control, instruction cycle, type of instructions, memory reference, register reference. I/O reference, Basics of Logic Design, accumulator logic, Control memory, address sequencing, micro-instruction formats, micro-program sequencer, Stack Organization, Instruction Formats, Types of interrupts; Memory Hierarchy.

TEXT AND REFERENCE BOOKS:

1. Computer Organization and Design, 2nd Ed., by David A. Patterson and John L. Hennessy, Morgan 1997, Kauffmann.
2. Computer Architecture and Organization, 3rd Edi, by John P. Hayes, 1998, TMH.
3. Operating Systems Internals and Design Principles by William Stallings, 4th edition, 2001, Prentice-Hall Upper Saddle River, New Jersey
4. Computer Organization, 5th Edi, by Carl Hamacher, Zvonko Vranesic, 2002, Safwat Zaky.
5. Structured Computer Organisation by A.S. Tanenbaum, 4th edition, Prentice-Hall of India, 1999, Eastern Economic Edition.
6. Computer Organisation & Architecture: Designing for performance by W. Stallings, 4th edition, 1996, Prentice-Hall International edition.
7. Computer System Architecture by M. Mano, 2001, Prentice-Hall.
8. Gill, Nasib Singh and Dixit J.B.: Digital Design and Computer Organisation, University Science Press (Laxmi Publications), New Delhi

Practical List:

- To study and verify the truth table of logic gates
- To simplify the given expression and realize it using basic logic gates and universal gates.
- Design and implement Half-Adder and Full-Adder circuits.
- Design and implement Half-subtractor and Full- Subtractor circuits.
- Design and simulate 2:1, 4:1 multiplexers using logic gates.
- Design and simulate encoders (4-to-2) and decoders (2-to-4).
- Implement SR, D, JK, and T Flip-Flops and verify truth tables.
- Design and simulate 4-bit shift registers (left and right shift).

Semester 2

Course code	CC3			
Category	Discipline Specific Courses			
Course title	Data Structure and Algorithm			
Course ID	242/MCA/CC201			
Scheme and Credits	L	T	P	Credits
	3	-	2	4
Theory Internal	15			
Theory External	35			
Practical Internal	15			
Practical External	35			
Total	100			
Duration of Exam	3 HRS			

Note: The examiner will set nine questions in total. Question one will have seven parts from all units and the marks of first question will be of 20% of total marks of Question Paper and the remaining eight questions to be set by taking two questions from each unit and the marks of each question from Question no.2 to 9 will be of 20% of total marks of Question paper . The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

Course outcomes

CO1. List or describe types of data structures and operations that can be implemented on these data structures.

CO2. Demonstrate the use of various data structure and their related operations

CO3. Apply appropriate data structures with respect to effective storage of data and efficiency of the required operations on data for solving real world problems.

CO4. Analyse the time complexity of searching and algorithms.

CO5. formulate data structures and prescribe operations for given real world situations.

UNIT – I

Introduction: Elementary data organization, Data Structure definition, Data type vs. data structure, Categories of data structures, Data structure operations, Applications of data structures, Algorithms complexity and time-space tradeoff, Big-O notataion. Strings: Introduction, Storing strings, String operations, Pattern matching algorithms.

UNIT – II

Arrays: Introduction, Linear arrays, Representation of linear array in memory, address calculations, Traversal, Insertions, Deletion in an array, Multidimensional arrays, Parallel arrays, Sparse arrays.

Linked List: Introduction, Array vs. linked list, Representation of linked lists in memory, Traversal, Insertion, Deletion, Searching in a linked list, Header linked list, Circular linked list, Two-way linked list, Threaded lists, Garbage collection, Applications of linked lists.

UNIT – III

Stack: Introduction, Array and linked representation of stacks, Operations on stacks, Applications of stacks: Polish notation, Recursion. Queues: Introduction, Array and linked representation of queues, Operations on queues, Deques, Priority Queues, Applications of queues.

UNIT – IV

Tree: Introduction, Definition, Representing Binary tree in memory, Traversing binary trees, Traversal algorithms using stacks. Graph: Introduction, Graph theory terminology, Sequential and linked representation of graphs.

SUGGESTED READINGS

1. Seymour Lipschutz, “Data Structure”, Tata-McGraw-Hill
2. Horowitz, Sahni & Anderson-Freed, “Fundamentals of Data Structures in C”, Orient Longman.
3. Trembley, J.P. And Sorenson P.G., “An Introduction to Data Structures With Applications”, Mcgrraw-Hill International Student Edition, New York.
4. Mark Allen Weiss Data Structures and Algorithm Analysis In C, Addison- Wesley, (An Imprint Of Pearson Education), Mexico City.Prentice- Hall Of India Pvt. Ltd., New Delhi.
5. Yedidyan Langsam, Moshe J. Augenstein, and Aaron M. Tenenbaum, “Data Structures Using C”, Prentice- Hall of India Pvt. Ltd., New Delhi

Data Structure and Algorithm

List of Experiments

Note: At least 8 experiments are to be performed by the students.

List of Subject related Experiments:

1. Write a program to search an element in a two-dimensional array using linear search.
2. Using iteration & recursion concepts write programs for finding the element in the array using Binary Search Method
3. Write a program to perform following operations on tables using functions only
(a) Addition (b) Subtraction (c) Multiplication (d) Transpose
4. Using iteration & recursion concepts write the programs for Quick Sort Technique

5. Write a program to implement the various operations on string such as length of string concatenation, reverse of a string & copy of a string to another.
6. Write a program for swapping of two numbers using 'call by value' and 'call by reference strategies.
7. Write a program to implement binary search tree.
8. (Insertion and Deletion in Binary search Tree)
9. Write a program to create a linked list & perform operations such as insert, delete, update, reverse in the link list
10. Write the program for implementation of a file and performing operations such as insert, delete, update a record in the file.
11. Create a linked list and perform the following operations on it
 - (a) add a node
 - (b) Delete a node

Course code	CC4			
Category	Discipline Specific Courses			
Course title	Operating System			
Course ID	242/MCA/CC202			
Scheme and Credits	L	T	P	Credits
	3	-	2	4
Theory Internal	15			
Theory External	35			
Practical Internal	15			
Practical External	35			
Total	100			
Duration of Exam	3 HRS			

Note: The examiner will set nine questions in total. Question one will be compulsory. Question one will have seven parts of 2 marks each from all units, and the remaining eight questions of 14 marks each to be set by taking two questions from each unit. The students have to attempt five questions in total, the first being compulsory and selecting one from each unit.

COURSE OUTCOMES:

At the end of this course, students will demonstrate the ability to

CO1. Experiment with Unix commands and shell programming.

CO2. Able to build shell program for process and file system management with system calls.

CO3. Able to implement and analyse the performance of different algorithm of Operating Systems like CPU scheduling algorithm,

CO4. Able to implement and analyse the performance of different algorithm of page replacement algorithms, deadlock avoidance, detection algorithm and so on.

CO5. Able to design and develop a course project that can have positive impact on environment or society or mankind.

UNIT – I

Fundamentals of Operating system: Introduction to Operating System, its need and operating System services, Early systems, Structures - Simple Batch, Multi programmed, timeshared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems. Process Management: Process concept, Operation on processes, Cooperating Processes, Threads, and Inter-process Communication.

UNIT-II

CPU Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms : FCFS, SJF, Round Robin & Queue Algorithms. Deadlocks: Deadlock characterization, Methods for handling deadlocks, Banker's Algorithm.

UNIT-III

Memory Management: Logical versus Physical address space, Swapping, Contiguous allocation, Paging, Segmentation. Virtual Memory: Demand paging, Performance of demand paging, Page replacement, Page replacement algorithms, Thrashing.

UNIT-IV

File management: File system Structure, Allocation methods: Contiguous allocation, Linked allocation, Indexed allocation, Free space management: Bit vector, Linked list, Grouping, Counting. Device Management: Disk structure, Disk scheduling: FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK.

Suggested Readings

1. Abraham Silberschatz, Peter B. Galvin, "Operating System Concepts", Addison-Wesley publishing. Co., 7th. Ed., 2004.
2. Nutt Gary, "Operating Systems", Addison Wesley Publication, 2000.
3. Andrew S. Tannenbaum, "Modern Operating Systems", Pearson Education Asia, Second Edition, 2001.
4. William Stallings, "Operating Systems, "Internals and Design Principles", 4th Edition, PH, 2001.
5. Ekta Walia, "Operating Systems Concepts", Khanna Publishes, New Delhi, 2002

LAB: List of Experiments

1. Basics of UNIX
2. UNIX commands
3. Implementation of FCFS and SJF CPU scheduling algorithms
4. Implementation of Round Robin and Priority CPU Scheduling
5. Implementation of Producer-Consumer problem using semaphores
6. Implementation of FIFO Page Replacement Algorithms

7. Implementation of LRU Page Replacement Algorithms

8. Implementation of Sequential File Allocation Strategies

9. Implementation of Indexed File Allocation Strategies