MCA -101
Discrete Mathematics

Subject Code	Subject Name & Title	Total Marks	End Semester Exam Marks		Sessional Marks		Total Credits
			Max	Min	Max	Min	
MCA-101	Discrete Mathematics	100	60	21	40	14	3

Course Objective:

To develop logical thinking and its application to computer science (to emphasize the importance of proving statements correctly and de-emphasize the hand-waving approach towards correctness of an argument). The subject enhances one's ability to reason and ability to present a coherent and mathematically accurate argument.

Syllabus:

Set theory and Counting techniques: Set, Subsets, Operations on set, finite infinite set, Principle of Inclusion – Exclusion, Principle of Mathematical induction, Principle of strong mathematical induction, Permutation and Combination, Logic Theory - Prepositional calculus and Predicate calculus.

Relations, Functions and Lattice: Properties of relations, equivalence relation, Partial order relation, poset, onto function, one–to–one functions, pigeon hole principle, Lattices, Distributive law in lattices, complemented lattice.

Graphs and Trees: Definitions, Algorithms, Euler path and circuit, Hamiltonian path and circuit, Planner and non-planer graphs, Characteristics of tree, theorems, minimum cost spanning tree.

Automata Theory: Finite State Automata: Deterministic, Non – deterministic M/c, regular expressions, regular language and regular grammar. Push down Automata: Deterministic, Non – deterministic push down automata, Context free language and grammar.

Turing Machine: Turing machine and compatibility, Types of turing machine, context sensitive language and grammar, Chomsky Hierarchy.

Books Recommended:

1. C. L. Liu, "Elements of Discrete Mathematics", TMH, 2000.
2. S. Lipschutz & M. Lipson, "Discrete Mathematics", TMH, 1999.
3. Peter Linz, "Introduction to Automata Theory, Languages, and Computation", Narosa Pub., 1997.
3. B. Kolman, R. Busby & S. Ross, "Discrete Mathematical Structures", Pearson Education.
4. J. P. Trembley and R. Manohar, "Discrete Mathematical Structures with Application to Computer Science", TMH, 1997.
5. John C. Martin, "Introduction to Languages and the Theory of Computation", TMH, 1998.
6. Cohen, "Introduction to Computer Theory", John Wiley & Sons, 1996.

Course Outcome:

- Be able to construct simple mathematical proofs and possess the ability to verify them.
- Have substantial experience to comprehend formal logical arguments.
- Be able to specify and manipulate basic mathematical objects such as sets, functions, and relations.

MCA -102
Fundamentals of Computer Programming

Subject Code	Subject Name & Title	Total Marks	End Semester Exam Marks		Sessional Marks		Total Credits
			Max	Min	Max	Min	
MCA-102	Fundamentals of Computer Programming	100	60	21	40	14	3

Course Objective:

1. This course introduces the concepts of computer basics & programming
2. The C programming language is used but the course will stress on fundamental parts of programming language, so that the students will have a basic concept for understanding and using other programming language.

Syllabus :

An overview: Problem identification, analysis, design, coding, testing & debugging, implementation, modification & maintenance; algorithms & flowcharts; Characteristics of a good program - accuracy, simplicity, robustness, portability, minimum resource & time requirement, modularization; Rules/conventions of coding, documentation, naming variables; Top down design; Bottom-up design.

Fundamentals of C Programming: History of C; Structure of a C Program; Data types; Constant & Variable; Operators & expressions; Control Constructs – if-else, for, while, do-while; Case statement; Arrays; Formatted & unformatted I/O; Type modifiers & storage classes; Ternary operator; Type conversion & type casting; Priority & associativity of operators.

Modular Programming: Functions; Arguments; Return value; Parameter passing – call by value, call by reference; Return statement; Scope, visibility and life-time rules for various types of variable, static variable; Calling a function; Recursion – basics, comparison with iteration, tail recursion, when to avoid recursion, examples.

Advanced Programming Techniques: Special constructs – Break, continue, exit(), goto & labels; Pointers - & and * operators, pointer expression, pointer arithmetic, dynamic memory management functions like malloc(), calloc(), free(); String; Pointer v/s array; Pointer to pointer; Array of pointer & its limitation; Function returning pointers; Pointer to function, Function as parameter; Structure – basic, declaration, membership operator, pointer to structure, referential operator, self referential structures, structure within structure, array in structure, array of structures; Union – basic, declaration; Enumerated data type; Typedef; command line arguments.

Miscellaneous Features: File handling and related functions; printf & scanf family; C preprocessor – basics, #Include, #define, #undef, conditional compilation directive like #if, #else, #elif, #endif, #ifdef and #ifndef; Variable argument list functions.

Books:

1. Kerninghan & Ritchie: The C programming language, PHI
2. Cooper Mullish: The Spirit of C, Jaico Publishing House, Delhi
3. Programming with C- Gottfried
4. Kanetkar Y.: Let us C
5. Kanetkar Y.: Pointers in C
6. Mastering C- K R Venugopal

Course Outcome:

On completion of the course students will be able to
1. Understanding the concept of input and output devices of Computers and how it works and recognize the basic terminology used in computer programming
2. Write, compile and debug programs in C language and use different data types for writing the programs.
3. Design programs connecting decision structures, loops and functions.
4. Explain the difference between call by value and call by address.
5. Understand the dynamic behaviour of memory by the use of pointers.
6. Use different data structures and create / manipulate basic data files and developing applications for real world problems.

MCA -103
Computer Organization & Architecture

Subject Code	Subject Name & Title	Total Marks	End Semester Exam Marks		Sessional Marks		Total Credits
			Max	Min	Max	Min	
MCA-103	Computer Organization & Architecture	100	60	21	40	14	3

Course Objectives:

- To understand the structure, function and characteristics of computer systems.
- To understand the design of the various functional units and components of computers.
- To identify the elements of modern instructions sets and their impact on processor design. • To explain the function of each element of a memory hierarchy,
- To identify and compare different methods for computer I/O.

Syllabus

Unit 1: Introduction: Evolution of digital computer, multilevel model of a Computer, Von-Neumann model, ALU, Control Unit, System bus, Memory, I/O Devices, Concept of instruction execution, Number systems, Data and number representation, Character codes, Binary arithmetic. Logic gates, Flip Flops, Latches, Registers, Boolean algebra, Combinational and sequential circuits, Arithmetic circuits.

Unit 2: Memory: Secondary and main memory; Main memory organization, properties and technologies; Associative memory, Cache memory. Control Unit operation: Micro operations, Control of the CPU, Hardwired and Micro programmed control. Machine Level Instructions: Instruction formats, Addressing modes, Instruction types, Instruction cycle, Flow of control.

Unit 3: Input Output : I/O devices, their characteristic, Interfacing, I/O ports, Memory mapped and I/O mapped I/O, Programmed I/O, Concept of interrupts, Interrupt driven and DMA based I/O, I/O processors, Device controllers, I/O device interfaces, Device Drivers, I/O and system buses, Serial and Parallel Communication.

Unit 4: Introduction to Advanced Architectures: Pipelining, Vector Processing, RISC v/s CISC, Multiprocessors, Fault Tolerant architectures.

Unit 5: Introduction to System Software: Assemblers, Assembly language Programming: Use of imperatives, declaratives, directives and macros; Compiler: Overview of compilation process; Interpreters; Loading, linking and relocation; Functions of editors and debuggers.

Books Recommended:

1. William Stalling, “Computer Organization and Architecture”, Pearson Education.
2. Douglas V. Hall, “Microprocessors and Interfacing”, TMH, 1991.
3. M. Morris Mano, “Computer System Architecture”, Pearson Education.
4. Andrew S. Tannenbaum, “Structured Computer Organization”, Pearson Education.
5. Dhamdhere, “System Programming & Operating System”, TMH, 1997.
6. Govindrajalu, “IBM PC and Clones, Hardware, Troubleshooting & Maintenance”, TMH, 1996.
7. Peter Abel, “IBM PC Assembly Language and Programming”, Pearson Education.

Course Outcome:

On completion of the course, student will be able to :

- Demonstrate computer architecture concepts related to design of modern processors, memories and I/Os.
- Analyze the performance of commercially available computers.
- To develop logic for assembly language programming

MCA -104
Operating Systems

Subject Code	Subject Name & Title	Total Marks	End Semester Exam Marks		Sessional Marks		Total Credits
			Max	Min	Max	Min	
MCA-104	Operating Systems	100	60	21	40	14	3

COURSE OBJECTIVE:

1. To understand the main components of an OS & their functions.
2. To study the process management and scheduling.
3. To understand various issues in Inter Process Communication (IPC) and the role of OS in IPC.
4. To understand the concepts and implementation Memory management policies and virtual memory.
5. To understand the working of an OS as a resource manager, file system manager, process manager, memory manager and I/O manager and methods used to implement the different parts of OS

Syllabus :

UNIT - I Introduction to OS.

Operating system functions, evaluation of O.S., Different types of O.S.: batch, multi-programmed, time-sharing, real-time, distributed, parallel. System Structure Computer system operation, I/O structure, storage structure, storage hierarchy, different types of protections, operating system structure (simple, layered, virtual machine), O/S services, system calls.

UNIT - II Process Management Processes

Concept of processes, process scheduling, operations on processes, co-operating processes, interprocess communication. Threads : overview, benefits of threads, user and kernel threads. CPU scheduling : scheduling criteria, preemptive & non-preemptive scheduling, scheduling algorithms (FCFS, SJF, RR, priority), algorithm evaluation, multi-processor scheduling. Process Synchronization : background, critical section problem, critical region, synchronization hardware, classical problems of synchronization, semaphores.

UNIT - III Deadlocks

Deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock. Storage Management Memory Management: background, logical vs. physical address space, swapping, contiguous memory allocation, paging, segmentation, segmentation with paging.

UNIT - IV Virtual Memory

Background, demand paging, performance, page replacement, page replacement algorithms (FCFS, LRU), allocation of frames, thrashing. File Systems : file concept, access methods, directory structure, file system structure, allocation methods (contiguous, linked, indexed), free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency & performance.

UNIT - V I/O Management

I/O hardware, polling, interrupts, DMA, application I/O interface (block and character devices, network devices, clocks and timers, blocking and nonblocking I/O), kernel I/O subsystem (scheduling, buffering, caching, spooling and device reservation, error handling), performance. Disk Management : disk structure, disk scheduling (FCFS, SSTF, SCAN,C-SCAN) , disk reliability, disk formatting, boot block, bad blocks. Protection & Security Goals of protection, domain of protection, security problem, authentication, one time password, program threats, system threats, threat monitoring, encryption.

REFERENCE BOOKS :

1. Milenkovie M., “Operating System : Concept & Design”, McGraw Hill.
2. Tanenbaum A.S., “Operating System Design & Implementation”, Practice Hall NJ. 3. Silbersehatz A. and Peterson J. L., “Operating System Concepts”, Wiley.
3. Dhamdhere: Operating System TMH
4. Stalling, William, “Operating Systems”, Maxwell McMillan International Editions, 1992.
5. Dietel H. N., “An Introduction to Operating Systems”, Addison Wesley.

COURSE OUTCOME:

1. Describe the general architecture of computers, contrast and compare differing structures for operating systems
2. Understand and analyse theory and implementation of: processes, resource control (concurrency etc.),
3. Physical and virtual memory, scheduling, I/O and files.

MCA -105
Programming Lab I (C, C++)

Subject Code	Subject Name & Title	Total Marks	End Semester Exam Marks		Sessional Marks		Total Credits
			Max	Min	Max	Min	
MCA-105	Programming Lab -I	**100**	**60**	**21**	**40**	**14**	**3**

This course is designed to familiarize students with the basic components of programming, so as to be able to initiate into the discipline of programming. It aims to start of the development of program solving ability using computer programming like C, C++ or programming assignments based on the subject present in current semester. Student will be able to design, develop, test and document structured programs in languages.