

# **NATIONAL ENGINEERING COLLEGE**

*(An Autonomous Institution – Affiliated to Anna University Chennai)*

**K.R.NAGAR, KOVILPATTI – 628 503**

[www.nec.edu.in](http://www.nec.edu.in)

## **REGULATIONS - 2013**



**DEPARTMENT OF  
COMPUTER SCIENCE AND ENGINEERING**

**CURRICULUM AND SYLLABI OF  
M.C.A – MASTER OF COMPUTER APPLICATIONS**

**FIRST YEAR**

**REGULATIONS - 2013**  
**Curriculum and Syllabi of Full Time**  
**M.C.A. – Master of Computer Applications**

**SEMESTER - I**

Sl. No.	Course Code	Course Title	L	T	P	C
<b>Theory</b>						
1.	MCC11	Computer Organization	3	0	0	3
2.	MCC12	Problem Solving and C Programming	3	0	0	3
3.	MCC13	Database Management Systems	3	0	0	3
4.	MCC14	Data Structures	3	0	0	3
5.	MCC15	Mathematical Foundations of Computer Science	3	1	0	4
<b>Practical</b>						
6.	MCC16	C and Data Structures Laboratory	0	0	3	2
7.	MCC17	Database Management Systems Laboratory	0	0	3	2
<b>TOTAL</b>			<b>15</b>	<b>1</b>	<b>6</b>	<b>20</b>

**SEMESTER – II**

Sl. No.	Course Code	Course Title	L	T	P	C
<b>Theory</b>						
1.	MCC21	Object Oriented Programming	3	1	0	4
2.	MCC22	Design and Analysis of Algorithms	3	1	0	4
3.	MCC23	System Software	3	0	0	3
4.	MCC24	Operating Systems	3	0	0	3
5.	MCC25	Software Engineering	3	0	0	3
<b>Practical</b>						
6.	MCC26	Object Oriented Programming Laboratory	0	0	3	2
7.	MCC27	System Programming Laboratory	0	0	3	2
8.	MCC28	Communication Skills Laboratory	1	0	2	2
<b>TOTAL</b>			<b>16</b>	<b>2</b>	<b>8</b>	<b>23</b>

**MCC11          COMPUTER ORGANIZATION****L T P C  
3 0 0 3****OBJECTIVES**

- To study the Digital fundamentals.
- To understand the concepts and issues of Computer Organization.
- To learn the basics of Memory and Processor design.
- To describe the Interrupts and I/O interfaces.

**UNIT I          DIGITAL FUNDAMENTALS          9**

Number Systems and Conversions – Boolean Algebra and Simplification – Minimization of Boolean Functions – Karnaugh Map – Logic Gates – NAND – NOR Implementation.

**UNIT II          COMBINATIONAL AND SEQUENTIAL CIRCUITS          9**

Design of Combinational Circuits – Adder / Subtractor – Encoder – Decoder – MUX / DEMUX – Comparators – Flip Flops – Triggering – Master/Slave Flip Flop – State Diagram and Minimization – Counters – Registers.

**UNIT III          BASIC STRUCTURE OF COMPUTERS          9**

Functional units – Basic operational concepts – Bus structures – Performance and Metrics – Instruction and Instruction sequencing – Addressing modes – Instruction Set: ARM Processor – ALU design – Fixed point and Floating point operation.

**UNIT IV          PROCESSOR DESIGN          9**

Processor basics – Execution of Complete Instruction – Multi bus organization– Hard wired control – Micro programmed control – Pipeline control - Basic concepts – Hazards – Super scalar operation.

**UNIT V          MEMORY AND I/O SYSTEM          9**

Memory technology – Memory systems – Virtual memory – Caches – Design methods Associative memories – Input/Output system – Programmed I/O – DMA and Interrupts – I/O Devices and Interfaces.

**TOTAL: 45****TEXT BOOKS**

1. M. Morris Mano, Michael D. Ciletti , “Digital Design”, 4<sup>th</sup> Edition, Prentice Hall of India, 1997.
2. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, “Computer Organization”, 5<sup>th</sup> Edition, Tata McGraw Hill Education, 2011.

**REFERENCES**

1. Charles H. Roth Jr, Larry L. Kinney, “Fundamentals of Logic Design”, 6<sup>th</sup> Edition, Cengage Learning, 2010.
2. William Stallings, “Computer Organization and Architecture – Designing for Performance”, 6<sup>th</sup> Edition, Pearson Education, 2003.
3. David A. Patterson, John L. Hennessy, “Computer Organization and Design: The Hardware/Software interface”, 2<sup>nd</sup> Edition, Morgan Kaufmann, 2002.
4. John P. Hayes, “Computer Architecture and Organization”, 3<sup>rd</sup> Edition, Tata McGraw Hill, 1998.

**MCC12          PROBLEM SOLVING AND C PROGRAMMING****L T P C  
3 0 0 3****OBJECTIVES**

- To learn and analyze problems and formulate algorithms.
- To learn the fundamentals of C.
- To understand the usage of arrays, functions and structures.
- To study the importance of pointers and files.

**UNIT I          INTRODUCTION TO PROGRAMMING****9**

Introduction to computing – Building blocks for simple programs – Problem to Program – Problem Solving with Decision Structures – Problem Solving with Loop Structures – Problem Analysis – Programming Style – Documentation and Testing – Procedural – Functional – Recursive – Rule-based – Structured programming.

**UNIT II          C PROGRAMMING BASICS****9**

Introduction to C programming – Constants – Variables – Data Types – Storage classes – Operators – Managing Input and Output Operations – Decision Making and Branching – Looping Statements – Solving Simple Scientific and Statistical Problems.

**UNIT III          ARRAYS AND FUNCTIONS****9**

Arrays – One dimensional and two dimensional arrays - String – String operations – String Arrays – Functions – Call by value – Call by reference – Recursion.

**UNIT IV          STRUCTURES AND UNIONS****9**

Introduction – Need for structure – Structure Declaration – Structure Definition – Nested Structures – Unions – Programs using Structures and Unions – Pre-processor Directives.

**UNIT V          POINTERS AND FILE MANIPULATION****9**

Pointers – Definition – Initialization – Pointer Arithmetic – Pointers with Arrays – Dynamic Memory Allocation – FILE Pointers – File Operation: Creation, Copy, Delete, Update – File Type: Text File and Binary File.

**TOTAL: 45****TEXT BOOKS**

1. Ashok N. Kamthane, “Computer programming”, 1<sup>st</sup> Edition, Pearson Education, 2007.
2. Yashavant P. Kanetkar, “Let Us C”, 12<sup>th</sup> Edition, BPB Publications, 2012.
3. Kernigan Brian W, Dennis M. Ritchie, “The C Programming Language”, 2<sup>nd</sup> Edition, Prentice Hall, 2006.

**REFERENCES**

1. Maureen Sprankle, “Problem Solving and Programming Concepts”, 7<sup>th</sup> Edition, Pearson Education, 2009.
2. Deitel and Deitel, “C How to program”, 2<sup>nd</sup> Edition, Prentice Hall, 1994.
3. Cormen, Leiserson, Rivest, Stein “Introduction to algorithms”, 2<sup>nd</sup> Edition, McGraw Hill publishers, 2002.
4. Stephen G.Kochan “Programming in C”, 3<sup>rd</sup> Edition, Pearson Education, 2008.

**MCC13          DATABASE MANAGEMENT SYSTEMS****L T P C  
3 0 0 3****OBJECTIVES**

- To understand the basics of Database Management System.
- To use Standard Query Language.
- To understand normalization techniques.
- To understand importance of backup and recovery techniques.

**UNIT I          INTRODUCTION****8**

Purpose of Database Systems – View of Data – Relational Databases – Database Design – Database Architecture – Database users and Administrators – Database Schema – Keys – Relational algebra – Tuple Relational Calculus – Domain Relational Calculus.

**UNIT II          SQL, PROGRAMMING AND TRIGGERS****11**

SQL Data Definition – Basic Structure of SQL Queries – Basic Operations – Set Operations – Aggregate Functions – Nested Subqueries – Modification of the Database – Join Expressions – Views – Transactions – Integrity Constraints – Authorization – Functions and Procedures – Triggers.

**UNIT III          NORMAL FORMS, STORAGE AND FILE STRUCTURE****9**

Introduction to Normal Forms – Normalization using Functional and Multivalued Dependencies – Overview of Physical Storage Media – Magnetic Disk and Flash Storage – RAID – File Organization – Organization of Records in Files – Ordered Indices – B<sup>+</sup> Tree Index Files.

**UNIT IV          HASHING, QUERY PROCESSING AND TRANSACTION****7**

Static and Dynamic Hashing – Overview of Query Processing and Query Optimization – Measures of Query cost. Transaction: Transaction concepts – Transaction state – Serializability.

**UNIT V          CONCURRENCY CONTROL AND RECOVERY SYSTEM****10**

Concurrency control: Lock-Based Protocols - Deadlock Handling – Multiple granularity – Timestamp-Based Protocols – Thomas' Write Rule – Validation-Based Protocols. Recovery System: Failure Classification – Storage structure – Recovery and Atomicity – Log-Based Recovery – Shadow Paging – Recovery with Concurrent Transactions.

**TOTAL : 45****TEXT BOOK**

1. Abraham Silberschatz, Henry. F. Korth and S. Sudharshan, "Database system Concepts", 6<sup>th</sup> Edition, McGraw-Hill Publishers, 2010.

**REFERENCES**

1. Raghu RamaKrishnan, Johannes Gehrke, "Database Management Systems", 3<sup>rd</sup> Edition, McGraw Hill Publishers, 2003.
2. C. J. Date, "An Introduction to Database Systems", 7<sup>th</sup> Edition, Addison Wesley Publishers, 1997.

**MCC14****DATA STRUCTURES****L T P C**  
**3 0 0 3****OBJECTIVES**

- To learn about the basic linear data structures, their implementations and applications.
- To make the students to study about non linear data structures.
- To learn about various sorting types and its efficiency.
- To make the students to learn about hashing techniques.

**UNIT I LINEAR DATA STRUCTURES****9**

Elementary Data Structures – ADT – List ADT: Array based Implementation – Doubly and Circularly Linked Lists – Cursor based Linked List – Applications of List – The stack ADT: Model – Implementation – Applications – The Queue ADT: Model – Implementation – Applications.

**UNIT II TREE STRUCTURES AND BALANCED TREES****9**

Trees - Preliminaries – implementation of Tree – Tree Traversals – Binary tree Implementation – Expression Tree – The Search Tree ADT – Binary Search Tree Operations – AVL tree – Splay tree – B-Tree – Binary Heap: Structure Property – Heap Order Property – Heap Operations – d-heaps.

**UNIT III GRAPH****9**

Graph - Representation of Graphs – Bi-connectivity – Topological sort – Graph Traversal: Breadth-First Traversal – Depth-First Traversal – Shortest path Algorithms: Minimum Spanning Tree – Prim's and Kruskal's Algorithms – Applications of Graph.

**UNIT IV SORTING****9**

Sorting – Lower Bound for Simple Sorting Algorithms – Internal and External Sorting: Bubble Sort – Insertion Sort – Shell Sort – Merge Sort – Heap Sort – Quick Sort – Bucket Sort – Radix Sort – Distribution Sort – Shuffle Sort.

**UNIT V HASHING AND SET****9**

Hashing: Hash function – Open hashing – Closed hashing – Double hashing – Extendible hashing – Rehashing. The Disjoint set ADT: Equivalence Relations – Dynamic Equivalence Problem – Smart Union Algorithms – Path compression – Applications of Set.

**TOTAL: 45****TEXT BOOK**

1. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2<sup>nd</sup> Edition, Pearson Education, 2012.

**REFERENCES**

1. Clifford A. Shaffer, “Data Structures and Algorithm Analysis”, 3<sup>rd</sup> Edition, Dover Publications, 2013.
2. Richard F. Gilberg, Behrouz A. Forouzan, “Data Structures: A Pseudocode Approach with C”, 2<sup>nd</sup> Edition, Cengage Learning, 2004.
3. Narasimha Karumanchi, “Data Structures and Algorithms Made Easy”, 2<sup>nd</sup> Edition, Career Monk Publications, 2011.

<b>MCC15</b>	<b>MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE</b>	<b>L T P C</b>
		<b>3 1 0 4</b>

**OBJECTIVES**

- To learn about the basic linear algebra concepts.
- To understand the relations and logics of computer science.
- To make the students to understand the basics of network theory using graph.

**UNIT I MATRIX ALGEBRA 12**

Matrices – Rank of Matrix – Solving System of Equations – Eigen Values and Eigen Vectors – Inverse of a Matrix – Cayley Hamilton Theorem.

**UNIT II SETS, RELATIONS AND FUNCTIONS 12**

Sets – Basic Definitions – Set operations – Laws of set theory – Principles of inclusion and exclusion – Relations – Properties of relations – Matrices of relations – Closure operations on relations – Functions: injective - surjective – bijective.

**UNIT III MATHEMATICAL LOGIC 12**

Propositions and logical operators – Truth table – Propositions generated by a set – Equivalences and implications – Basic laws – Some more connectives – Functionally complete set of connectives – Normal forms – Proofs in Propositional calculus.

**UNIT IV FORMAL LANGUAGES 12**

Languages and Grammars – Phrase Structure Grammar – Classification of Grammars – Pumping Lemma for Regular Languages – Context Free Languages.

**UNIT V GRAPH THEORY 12**

Graph – Special types of graphs – Sub graph – Graph isomorphism – Euler graph – Hamiltonian graph.

**TOTAL: 60****TEXT BOOKS**

1. Kenneth H.Rosen, “Discrete Mathematics and Its Applications”, 4<sup>th</sup> Edition, Tata McGraw Hill, 2002
2. J.P.Trembly, R.Manohar “Discrete Mathematical Structures with Applications to Computer Science”, Tata McGraw-Hill, 1997

**REFERENCES**

1. Hopcroft, Ullman, “Introduction to Automata Theory, Languages and Computation”, Narosa Publishing House, 2002.
2. Narsingh Deo, “Graph Theory with applications to Engineering and Computer Science”, Reprint, Prentice-Hall of India Pvt.Ltd, 2006.
3. A.Tamilarasi, A.M.Natarajan, “Discrete Mathematics and its Application”, 2<sup>nd</sup> Edition, Khanna Publishers, 2005.
4. M.K.Venkataraman “Engineering Mathematics”, Volume II, 2<sup>nd</sup> Edition, National Publishing Company, 1989.

**MCC16 C AND DATA STRUCTURES LABORATORY****L T P C**  
**0 0 3 2****OBJECTIVES**

- To implement various operations on linear data structures.
- To perform the various tree traversals in binary tree.
- To implement various internal and external sorting techniques.
- To implement shortest path algorithms using Floyd's algorithm and Warshall's algorithm.
- To implement minimum spanning tree of graph using Prim's algorithm and Kruskals algorithm.

**List of Experiments:**

1. Write a program to perform the following operations on single linked list:
  - i) Creation ii) Insertion iii) Deletion iv) Reversal v) Sorting vi) Traversal
2. a) Write a program that uses functions to perform the following operations on Double linked list:
  - i) Creation ii) Insertion iii) Deletion iv) Traversal
 b) Write a program to create and traverse a circular single linked list
3. a) Write a program to create a stack using array and linked list and perform the following operations:
  - i) push ii) pop iii) display
 b) Write a program to create a queue using array and linked list and perform the following operations:
  - i) insert ii) delete iii) display
4. Write a program to perform the following stack applications:
  - i) Convert infix expression into postfix expression
  - ii) Evaluation of postfix expression
5. Write the program to perform the following traversal in Binary tree
  - i) Inorder traversal ii) Preorder traversal iii) Postorder traversal
6. Write a program to perform the following operations in Binary Search Tree (BST):
  - i) Creation ii) Insertion iii) Deletion iv) Traversal
7. Write a program that implement the following sorting techniques:
  - i) Bubble sort ii) Insertion sort iii) Selection sort
8. Write a program that implement the following sorting techniques:
  - i) Quick sort ii) Merge sort iii) Heap sort
9. Write a program to perform following Traversal in Graph
  - i) Breadth First Search, ii) Depth first search
10. Write a program to find out the shortest path between every vertex in Graph using the following algorithms
  - i) Floyd's algorithm ii) Warshall's algorithm
11. Write a program to find out the shortest path in Graph using the following algorithms
  - i) Prim's algorithm ii) Kruskal's Algorithm
12. Write a program to implement Huffman's Algorithm.

Required Software: C/C++



**MCC17      DATABASE MANAGEMENT SYSTEMS LABORATORY****L T P C  
0 0 3 2****OBJECTIVES**

- To implement the Basic Queries.
- To implement the SQL program.
- To implement the GUI program.

**List of Experiments**

1. Execute DDL commands for tables like Bank Management System.
2. Execute DML commands for table for Student Processing System.
3. Execute and manipulate the views for Bank Management System.
4. Apply different joins like equi-join and outer join by relating two or more tables in Payroll processing system.
5. Manipulate student processing system using SET operations and Aggregate functions.
6. Execute Built-in functions for Characters, Numbers and Date.
7. Write basic PL/SQL programming for the following:
  - a) Write a programme to find the factorial of a number.
  - b) Find odd and even numbers from 100 to 1000.
  - c) Write a program to input 2 numbers if the 1<sup>st</sup> no > 2<sup>nd</sup> no then swap it, else if 1<sup>st</sup> no < 2<sup>nd</sup> no doubles it else multiply 10 with both numbers?
  - d) Write a program to input 2 numbers and an operator, and display the result.
  - e) Print multiplication table from 1 to a given Number.
8. Do Advanced PL/SQL programming using
  - a) Cursors   b) Procedures   c) Triggers
9. Create Forms, Menus and Report for
  - a) Payroll Processing System
  - b) Bank Management System
  - c) Library Management System

Required Software: Oracle, Visual Basic.

**MCC21 OBJECT ORIENTED PROGRAMMING****L T P C**  
**3 1 0 4****OBJECTIVES**

- To understand the fundamentals of OOP's Concepts.
- To understand the C++ Concepts.
- To understand the file operations in C++.

**UNIT I FUNDAMENTALS****10**

Object Oriented Programming Concepts: Objects - Classes – Methods – Messages – Abstraction and Encapsulation – Inheritance – Abstract Classes – Polymorphism. Introduction to C++: Classes – Access Specifiers – Function and Data Members – Default Arguments – Static Members – Class Objects – Pointers and Objects – Constant Objects – Nested Classes – Local Classes.

**UNIT II FUNCTIONS, CONSTRUCTORS AND DESTRUCTORS****15**

Function: Inline Function – Difference between member and non-member function – Call by reference and Return by reference – Function Overloading – Friend Function – Const and Volatile Functions – Static Function – Private and Public functions – Function Pointer. Constructor: Default and Explicit Constructors – Parameterized Constructors – Constructor Overloading – Dynamic Constructor – Copy Constructor – Destructors.

**UNIT III INHERITANCE AND POLYMORPHISM****15**

Need of Inheritance - public, private, and protected derivations – Access control. Types of inheritance: Single inheritance, Multilevel inheritance, Multiple inheritance, Hierarchical inheritance and Hybrid inheritance – Virtual base classes – Constructors in derived class. Pointers – Pointers to Objects – this Pointer – Pointers to derived classes - Virtual and Pure Virtual Functions – Virtual Constructors and Destructors.

**UNIT IV I/O FORMATTING AND FILE HANDLING****9**

Streams and Formatted I/O: C++ stream for console I/O operations – Classes Hierarchy – Formatted and unformatted I/O – Formatting I/O. File handling: Classes for file stream operations - Opening and Closing a file – End of file detection – File Pointers and their manipulations - Random access – Error handling during file operations.

**UNIT V TEMPLATES****11**

Drawbacks of Macros - Function Templates – Function Templates with Multiple / Two Generic Arguments – Overloading of Template Function – Member Function Templates – Class Templates – Class Template with Multiple Arguments – Exception Handling.

**L : 45 T : 15****TOTAL : 60****TEXT BOOKS**

1. Bhushan Trivedi, "Programming with ANSI C++- A Step-by-Step Approach", 4<sup>th</sup> Impression, Oxford University Press, 2012.
2. E.Balagurusamy, "Object Oriented Programming with C++", 5<sup>th</sup> Edition, McGraw Hill Education, 2012.

**REFERENCES**

1. Robert Lafore, "Object oriented Programming in C++", 4<sup>th</sup> Edition, Sams Publishing, 2002.
2. Stanley B. Lippman, Josee Lajoie, Barbara E. Moo, "C++ Primer", 4<sup>th</sup> Edition, Addison Wesley Professional Publisher, 2005.

**MCC22      DESIGN AND ANALYSIS OF ALGORITHMS****L T P C  
3 1 0 4****OBJECTIVES**

- To study about the fundamentals of problem solving and algorithm analysis.
- To understand the problem using Divide and conquer methods and Greedy technique.
- To learn about Dynamic programming techniques to solve Knapsack problem.
- To study about N Queens problem, sum of subset problem using Backtracking method.
- To learn about approximation algorithm for NP-hard and NP-complete problems.

**UNIT I      INTRODUCTION****12**

Fundamentals of algorithmic problem solving – Important problem types – Fundamentals of analysis of algorithm efficiency – Analysis frame work – Asymptotic notations – Mathematical analysis for recursive and non-recursive algorithms.

**UNIT II      DIVIDE AND CONQUER METHOD AND GREEDY METHOD****12**

Divide and conquer methodology – Merge sort – Quick sort – Binary search – Binary tree traversal – Multiplication of large integers – Strassen's matrix multiplication – Greedy method – Prim's algorithm – Kruskal's algorithm – Dijkstra's algorithm.

**UNIT III      DYNAMIC PROGRAMMING****12**

Computing a binomial co-efficient – Warshall's and Floyd's algorithm – Optimal binary search tree – Knapsack problem – Memory functions.

**UNIT IV      BACKTRACKING AND BRANCH AND BOUND****12**

Backtracking – N-Queens problem – Hamiltonian circuit problem – Subset sum problem – Branch and bound – Assignment problem – Knapsack problem – Traveling salesman problem.

**UNIT V      NP-HARD AND NP-COMPLETE PROBLEMS****12**

P & NP problems – NP-complete problems – Approximation algorithms for NP-hard problems – Traveling salesman problem – Knapsack problem.

**L: 45      T: 15      TOTAL: 60****TEXT BOOKS**

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", 2<sup>nd</sup> Edition, Pearson Education, 2011.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of computer algorithms", 2<sup>nd</sup> Edition, Prentice Hall, 2008.

**REFERENCES**

1. Horowitz, Sahni, Anderson-Freed, "Fundamentals of Data Structures in C", 2<sup>nd</sup> Edition, University Press, 2007.
2. G. A.V.PAI, "Data structures and algorithms, concepts, Techniques and Applications", 1<sup>st</sup> Edition, Tata McGraw Hill, 2008.
3. Jean-Paul Tremblay, Paul G. Sorenson, "An Introduction to Data Structures with Applications", 2<sup>nd</sup> Edition, Tata McGraw Hill, 2007.
4. Parag Dave & Himanshu Dave, "Design and Analysis of Algorithms", 2<sup>nd</sup> Edition, Pearson Education, 2008.

**MCC23      SYSTEM SOFTWARE****L T P C**  
**3 0 0 3****OBJECTIVES**

- To discuss the machine architecture of SIC & SIC/XE.
- To elaborate the functionality of an assembler, loader and linker with necessary data structures.
- To differentiate between machine dependent and machine independent assembler features.
- To design assemblers, loaders, linkers and text editors.
- To describe the functionality of macro processors.

**UNIT I      INTRODUCTION****9**

Introduction – System software and machine architecture – The Simplified Instructional Computer (SIC) – Machine Architectures (SIC and SIC/XE) – Data and Instruction Formats – Addressing Modes – Instruction sets – I/O.

**UNIT II      ASSEMBLERS****9**

Basic assembler functions – A simple SIC assembler – Assembler algorithms and Data Structures – Machine dependent assembler features: Instruction formats and addressing modes – Program relocation. Machine independent assembler features: Literals – Symbol-defining statements – Expressions – Program Blocks – Control Sections and Program Linking. One-Pass Assembler and Multi-pass Assembler – Implementation examples: MASM assembler.

**UNIT III      LOADERS AND LINKERS****9**

Basic loader functions: Design of an Absolute Loader – A Simple Bootstrap Loader - Machine dependent loader features: Relocation – Program Linking – Algorithm and Data Structures for Linking Loader. Machine-independent loader features: Automatic Library Search – Loader Options - Loader design options: Linkage Editors – Dynamic Linking – Bootstrap Loaders - Implementation example: MSDOS linker.

**UNIT IV      MACRO PROCESSORS****9**

Basic macro processor functions – Macro Definition and Expansion – Macro Processor Algorithm and Data Structures – Machine independent macro processor features: Concatenation of Macro Parameters – Generation of Unique Labels – Conditional Macro Expansion – Keyword Macro Parameters. Macro Processor Design Options: Recursive Macro Expansion – General Purpose macro Processors – Macro Processing within Language Translators - Implementation examples: MASM Macro Processor – ANSI C macro language.

**UNIT V      TEXT EDITOR AND DEBUGGER****9**

Text editors – Overview of Editing Process – User Interface – Editor Structure – Interactive Debugging Systems – Debugging functions and capabilities – User Interface Criteria.

**TOTAL: 45****TEXT BOOK**

1. Leland L. Beck, “System Software – An Introduction to Systems Programming”, 3<sup>rd</sup> Edition, Pearson Education, 1999.

**REFERENCE**

1. D. M. Dhamdhare, “Systems Programming and Operating Systems”, 1<sup>st</sup> Edition, Tata McGraw-Hill Education, 2011.

**MCC24 OPERATING SYSTEMS****L T P C  
3 0 0 3****OBJECTIVES**

- Understand how the operating system abstractions can be used in the development of application programs or to build higher level abstractions.
- Understand how the operating system abstractions can be implemented.
- Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software.
- Understand basic resource management techniques and principles and how they can be implemented. These also include issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection.

**UNIT I INTRODUCTION**  
**9**

Definition and types of operating systems – Batch Systems – multi programming – Time-sharing – parallel – Distributed and real-time systems – Operating system structure – Operating system components and services – System calls – System programs – Virtual machines.

**UNIT II PROCESS MANAGEMENT**  
**9**

Process concept – Process scheduling – Cooperating processes – Threads – Interprocess communication – CPU scheduling criteria – Scheduling algorithms – Multiple-processor scheduling.

**UNIT III PROCESS SYNCHRONIZATION AND DEADLOCKS**  
**9**

The Critical-Section problem – Synchronization hardware – Semaphores – Classical problems of synchronization – Critical regions – Monitors – Deadlocks – System model – Characterization – Deadlock prevention – Avoidance – Detection – Recovery from deadlock – Combined approach to deadlock handling.

**UNIT IV STORAGE MANAGEMENT**  
**9**

Memory Management – Logical and Physical Address Space – Swapping – Contiguous Allocation – Paging – Segmentation – Virtual Memory – Demand paging and its performance – Page replacement algorithms – Thrashing – Demand segmentation – File systems – File concept – Access methods – Directory implementation – Efficiency and performance – Recovery – Disk structure – Disk scheduling methods.

**UNIT V CASE STUDY**  
**9**

Case study – Introduction – Process – File system – Storage management (WINDOWS, UNIX and LINUX).

**TOTAL: 45****TEXT BOOK**

1. Abraham Siberschatz, Peter Baer Galvin, “Operating System Concepts”, 7<sup>th</sup> Edition, John Wiley & Sons, Reprint 2009.

**REFERENCES**

1. Richard Peterson, “Linux: The Complete Reference”, 6<sup>th</sup> Edition, McGraw-Hill, 2003.
2. Maurice J. Bach, “Design of the Unix Operating System”, Pearson Education, 1996.
3. William Stallings, “Operating systems: internals and design principles”, 6<sup>th</sup> Edition, Prentice Hall, 2008.

**MCC25 SOFTWARE ENGINEERING****L T P C  
3 0 0 3****OBJECTIVES**

- To gain knowledge of basic software engineering methods and practices, and their appropriate application.
- A general understanding of software process models such as the waterfall and evolutionary models.
- An understanding of the role of project management software requirements SRS document, implementation issues, verification and validation reviews, software testing approaches, software evolution, version management and how to ensure good quality software.

**UNIT I INTRODUCTION****9**

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

**UNIT II SOFTWARE REQUIREMENT SPECIFICATIONS (SRS)****9**

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. Software Quality Assurance (SQA): Verification and Validation – SQA Plans – Software Quality Frameworks – ISO 9000 Models – CMM Model.

**UNIT III SOFTWARE DESIGN****9**

Basic Concept of Software Design – Architectural Design – Low Level Design: Modularization – Design Structure Charts – 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 – Halstead's Software Science – Function Point (FP) Based Measures – Cyclomatic Complexity Measures – Control Flow Graphs.

**UNIT IV SOFTWARE TESTING****9**

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 V SOFTWARE MAINTENANCE****9**

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 – An Overview of CASE Tools.

**TOTAL: 45****TEXT BOOK:**

1. R. S. Pressman, "Software Engineering - A Practitioners Approach", 7<sup>th</sup> Edition, McGraw Hill, 2009.

**REFERENCES:**

1. K. K. Aggarwal and Yogesh Singh, "Software Engineering", 3<sup>rd</sup> Edition, New Age International Publishers, 2008.
2. Ian Sommerville, "Software Engineering", 9<sup>th</sup> Edition, Addison Wesley, 2010.

**MCC26                      OBJECT ORIENTED PROGRAMMING LABORATORY****L T P C  
0 0 3 2****OBJECTIVES**

- To implement the basic concepts of C++.
- To implement the polymorphism and template concepts.
- To implement the file operations.

**List of Experiments:**

1. Write a program to perform a student's internal mark calculation.
2. Write a program to calculate the area of circle, rectangle and triangle using function overloading
3. Write a class Square which has a field for side. It must have a constructor to initialize the side. Add methods to the Square class to calculate area and perimeter.
4. Write a class Circle which has a field for radius. It must have a constructor to initialize the radius. Add methods to the Circle class to calculate area and perimeter.
5. Write a class CheckoutCalculator which behaves somewhat like the machine at the checkout counter in supermarkets. This calculator should ask for the number of items for which the total is to be calculated and then allow you to enter the price for every item. After entering all the items it displays all the prices entered and the total amount. (Hint: Here you need to use dynamic memory allocation since you do not know how many items will be there. Use an array to store the prices of items)
6. Write a class Results which stores all the results in an array. Assume that we need to store only the results of a single semester which is four results. Each result will be final marks for a course between 0-100 and is stored in an integer array. Use dynamic memory allocation and the four methods mentioned above. Add separate methods to calculate the total marks for all courses and the average marks. Add a field for student ID.
7. Modify the Circle class which you have written earlier to overload the + operator so that you can add two Circle objects. Adding two Circle object should give another Circle whose radius is the sum of the radius of the two Circle objects.
8. Modify the Rectangle class which you have written earlier to overload the + operator so that you can add two Rectangle objects. Adding two Rectangle objects should give another Rectangle object whose length is the sum of the lengths of the two Rectangle objects and whose breadth is the sum of the breadths of the two Rectangle objects.
9. Write a class Time which represents time. The class should have three fields for hours, minutes and seconds. It should have constructor to initialize the hours, minutes and seconds. A method printTime() to print the current time. Overload the following operators: plus operator (+) (add two time objects based on 24 hour clock) and < (compare two time objects)

10. Modify the Rectangle class which you have written earlier to overload the < (less than) operator so that you can compare two Rectangles as to which is bigger (or smaller). The logic you will use is to compare the areas of the Rectangle objects and decide which is bigger.
11. Modify the Circle class which you have written earlier to overload the > (greater than) operator so that you can compare two Circles as to which is bigger (or smaller). The logic you will use is to compare the radii of the Circle objects and decide which is bigger.
12. Write a program to perform Stack and Queue operations using template.
13. Write a function calculateAverage() which takes four int arguments which are marks for four courses in the semester and returns their average as a float. The calculateAverage() function should take only valid range for marks which is between 0 - 100. If the marks are out of range throw an OutOfRangeException - define this exception as a class.
14. Write a program to perform a bank management system using file.

Required Software: C++



**MCC27      SYSTEM PROGRAMMING LABORATORY****L T P C  
0 0 3 2****OBJECTIVES**

- To develop and analyze the data structures needed for developing an assembler, loader and linker.
  - To perform file handling using file manipulation system calls
  - To implement process management
  - To implement IPC techniques
  - To analyze scheduling algorithms and demonstrate page replacement policies
1. Write a C program to create symbol table and intermediate file using the algorithm for pass1 of the two pass assembler.
  2. Write a C program to create object file and list file with the output of the previous program using the algorithm for pass2 of the two pass assembler.
  3. Write a C program to implement the absolute loader.
  4. Write a C program to implement relocating loader.
  5. Write a C Program to implement the pass1 of the linking loader.
  6. Write a C program to implement the pass2 of the linking loader.
  7. Develop a text editor with features like insertion / deletion of a character, word, and sentence.
  8. Write a C program to perform file management tasks using file manipulation system calls in UNIX creat( ), open( ), read( ), write( ), close( )
  9. Write a C program to create a new process using fork( ). Make the child process to execute a new program using exec( ). Terminate the execution of the child process using exit( ). Make the parent to read the termination status of the child process using wait( ) / waitpid( ).
  10. Write a C program to implement IPC using pipe( ).
  11. Write a C program to implement IPC using message queue for unrelated process.
  12. Write a program to implement the following process scheduling algorithms
    - a. First Come First Serve
    - b. Shortest Remaining Job First (preemptive & non preemptive)
    - c. Round Robin
    - d. Priority Scheduling
  13. Write a program that demonstrates how two processes can share a variable using semaphore.
  14. Write a program to implement producer consumer problem using semaphore.
  15. Write a program to demonstrate page replacement policies like
    - a. Optimal
    - b. Least Recently Used (LRU)
    - c. First-In-First-Out

Required Software: LINUX &amp; C

**MCC28 COMMUNICATION SKILLS LABORATORY****L T P C**  
**1 0 2 2****OBJECTIVES**

- To help the learners to improve their communicative skill.
- To facilitate the learners to improve the pronunciation of words with proper stress.
- To help the learners acquire the soft skills and interpersonal skills which will help the students to excel in their workplace.
- To inculcate the habit of reading and to improve the active vocabulary among the learners.
- To enhance the performance of students in placement, interviews and Group discussion.

**Unit I****15**

1. Vocabulary Building
2. Splitting Syllables
3. Stress and Shift of words and sentences
4. Common errors in Speaking
5. Letter writing
6. Writing Application, Bio-data, Resume, Curriculum Vitae.
7. Reading Comprehension and Answering Multiple Choice questions and Fill ups.

**Unit II****10**

1. Listening to audio files and answering questions
2. Planning for an event
3. Extempore Speech - On the spot topics for speech Practice
4. Identifying tonal variations expressing
5. E-Mail writing

**Unit III****10**

1. Listening to Presentation Skills (GD & Debate)
2. Group Discussion
3. Reading Practice: Dr.Abdul Kalam's "Wings of Fire"
4. Report Writing
5. Paper Presentation

**Unit IV****10**

1. Listening to Telephonic Conversation & Situational Conversation
2. Debate
3. Note Taking
4. Interview Skills

**RECORD LAY OUT**

Every student has to maintain a record in which he / she have to incorporate the following details.

1. Students have to collect materials related to topics for Group Discussion / Debate
2. 10 assignments of Lab observations related to Presentation Skills about 200 words each
3. Covering letter with Bio data / Resume / Curriculum Vitae
4. Paper Presentation Topics with source materials to be pasted in the record

**MODE OF EVALUATION****INTERNAL ASSESSMENT (30 Marks)**

1. Writing Skill (**10 marks**)
2. Communication Skill (**15 marks**)
3. Attendance (**5 Marks**)

**EXTERNAL ASSESSMENT (70 Marks)**

1. Reading a Passage with proper stress and intonation (**10 Marks**)
2. Language Focus – (**10 marks**)
3. Writing Skill – (**20marks**)
4. Group discussion / Debate – (**30 marks**)

**TOTAL: 45**

Required Software: Oral Digital Language Lab & Globarena