

SEMESTER III

(Applicable to the students admitted from the Academic year 2006 – 2007 onwards)

Code No.	Course Title	L	T	P	M
THEORY					
1	Discrete Mathematics	4	0	0	100
2	Data Structures Common for CSE & IT	4	0	0	100
3	Digital Electronics Common for CSE & IT	4	0	0	100
4	System Software	4	0	0	100
5	Object Oriented Programming Common for CSE & IT	4	0	0	100
6	Electron Devices & Circuits	4	0	0	100
PRACTICAL					
1	OOPS & Data Structures Lab (Common for CSE & IT)	0	0	3	100
2	Digital Lab (Common for CSE & IT)	0	0	3	100

. SEMESTER III

DISCRETE MATHEMATICS

- | | | |
|----|---|---|
| 1. | LOGIC
Statements -Truth Tables – connectives - Normal forms - Predicate Calculus - Inference theory for statement calculus and Predicate Calculus. | 9 |
| 2. | COMBINATORICS
Review of Permutation and combination - Mathematical Induction - Pigeon hole principle - Principle of inclusion and exclusion - Generating function - Recurrence relations. | 9 |
| 3. | GROUPS
Semi groups – Monoids – groups - permutation group – Co-sets – Lagrange’s theorem - Group homomorphism – Kernal - Rings and Fields (Definitions and Examples only). | 9 |
| 4. | LATTICES
Partial ordering - Posets - Hasse diagram - Lattices - Properties of Lattices - Sub Lattices - Special Lattices - Boolean Algebra. | 9 |
| 5. | GRAPHS
Introduction to Graphs - Graph terminology - Representation of Graphs - Graph Isomorphism – Connectivity - Euler and Hamilton Paths. | 9 |

Total Hours: 45

Text Book:

1. " Discrete Mathematical Structures with Applications to Computer Science ", by Tremblay J.P, and Manohar R., McGraw Hill Book Company, 1975, International Edition, 1987. Sections: 1-2.1 to 1-2.4; 1-2.6 to 1-2,14; 1-3.1 to 1-3.5; 1-4. 1 to 1-4.3; 1-5.1 to 1-5.5; 1-6.4 and 1-6.5 for Logic. Sections: 3-1.1 to 3-2.3; 3-5.1 to 3-5.5 for Groups Rings and Fields. Sections:2-3.8 and 2-39; 4-1.1 to 4.2.2 for Lattices.
2. " Discrete and Combinatorial mathematics ", Ralph P., Grimaldi, Addison-Wesley Publishing Company, Reprinted in 1985. Sections : 1.1. to 1.3; 4.1 and 4.2.; 5.5; 8.1 to 8.3; 9.1 to 9.5; 10.1, 10.2 and 10.4 for Combinatorics.
3. "Discrete Mathematics and its Applications ", Kenneth H.Rosen, McGraw Hill Book Company, 1999. Sections: 7.1 to 7.5.

REFERENCES

1. Y. Langsam, M. J. Augenstein and A. M. Tenenbaum, "Data Structures using C", Pearson Education Asia, 2004
2. Richard F. Gilberg, Behrouz A. Forouzan, "Data Structures – A Pseudocode Approach with C", Thomson Brooks / COLE, 1998.
3. Aho, J. E. Hopcroft and J. D. Ullman, "Data Structures and Algorithms", Pearson education Asia, 1983.

SEMESTER III
OBJECT ORIENTED PROGRAMMING IN C++
(Common to IT & CSE)

Aim: The subject aims to introduce the salient features of Object Oriented Programming, evolution of c++ as a object oriented programming language with focus on generic programming with templates and Exception Handling.

Objectives: Students will be able to know the fundamentals of object oriented programming and incorporate OOPs' features such as inheritance, polymorphism, templates using c++ as language.

UNIT – I

Introduction to Fundamentals concepts: Object oriented fundamentals- Structured versus object-oriented development, elements of object oriented programming, fundamentals of OO-class, object, and abstraction and its importance, encapsulation, polymorphism, benefits of OOP, structure of C++ program.

UNIT – II

Classes and Objects: Working with classes- Classes and Objects- Class specification, class objects, accessing class members, defining member functions, inline functions, accessing member functions within class, data hiding, class member accessibility, empty classes, constructors, parameterized constructors, constructor overloading, copy constructor, new, delete operators, “this” pointer, friend classes and friend functions.

UNIT – III

Overloading: Overloading-Function overloading, operator overloading- overloadable operators, unary operator overloading, operator keyword, limitations of increment/decrement operators, binary operator overloading, arithmetic operators, concatenation of strings, comparison operators, Generic programming with templates-Function templates, class templates.

UNIT – IV

Inheritance: Inheritance- Base class and derived class relationship, derived class declaration, Forms of inheritance, inheritance and member accessibility, constructors in derived class, destructors in derived class, constructor invocation and data member initialization, data conversion, abstract classes, virtual base classes, virtual functions.

UNIT – V

Exception handling and Files: Files and Streams-Opening and Closing a file, file modes, file pointers and their manipulation, sequential access to a file, ASCII and binary files, random access to a file, error handling during file manipulations, Exception handling-exception handling model, exception handling constructs, lists of exceptions, catching exceptions, handling exceptions.

TEXT BOOK:

1. K.R.Venugopal, T.Ravishankar, and Rajkumar, "Mastering C++", Tata McGraw Hill, 1997

REFERENCE BOOKS:

1. E.Balagurusamy, “Object Oriented Programming with C++”, Tata McGraw Hill, 2nd Edition, 2004,
2. Bjarne stroustrup, “The C++ programming Language”, Addison Wesley, 3rd edition, 1988.

SEMESTER III

DIGITAL ELECTRONICS

1. BASIC CONCEPTS AND BOOLEAN ALGEBRA

9

Number systems - Binary, Octal, Decimal, Hexadecimal, conversion from one to another, complement arithmetic, Boolean theorems of Boolean algebra, Sum of products and product of sums, Minterms and Maxterms, Karnaugh map, Tabulation and computer aided minimization procedures.

2. LOGIC GATES

9

RTL, DTL, TTL, ECL, ICL, HTL, NMOS & CMOS logic gates, Circuit diagram and analysis characteristics and specifications, tri-state gates.

3. COMBINATIONAL CIRCUITS

9

Problem formulation and design of combinational circuits, Adder / Subtractor, Encoder / decoder, Mux / Demux, Code-converters, Comparators, Implementation of combinational logic using standard ICs, ROM, EPROM, EEPROM, Basics of PLD,PAL, PLA and their use in combinational circuit design.

4. SEQUENTIAL CIRCUITS

9

Flipflops - SR, JK, T, D, Master/Slave FF, Triggering of FF, Analysis of clocked sequential circuits - their design, State minimization, state assignment, Circuit implementation, Registers-Shift registers, Ripple counters, Synchronous counters, Timing signal, RAM, Memory decoding, Semiconductor memories.

5. FUNDAMENTAL MODE SEQUENTIAL CIRCUITS

9

Stable, Unstable states, Output specifications, Cycles and Races, Race free Assignments, Hazards, Essential hazards, Pulse mode sequential circuits.

TOTAL HOURS: 45

TEXT BOOKS:

1. Morris Mano, " Digital logic and Computer Design ", Prentice-Hall of India, 1998.
2. William I. Fletcher, " An Engineering Approach to Digital Design ", Prentice-Hall of India, 1980.
3. Floyd T.L., " Digital Fundamentals ", Charles E. Merrill publishing Company, 1982.
4. Tokheim R.L., " Digital Electronics - Principles and Applications ", Tata McGraw Hill, 1999.
5. Jain R.P., " Modern Digital Electronics ", Tata McGraw Hill, 1999.

SEMESTER III

SYSTEM SOFTWARE

(COMMON TO III SEM CSE & IV SEM IT)

AIM

To have an understanding of foundations of design of assemblers, loaders, linkers, and macro processors.

OBJECTIVES

- To understand the relationship between system software and machine architecture.
- To know the design and implementation of assemblers
- To know the design and implementation of linkers and loaders.
- To have an understanding of macroprocessors.
- To have an understanding of system software tools.

UNIT I INTRODUCTION 9

System software and machine architecture – The Simplified Instructional Computer (SIC) - Machine architecture - Data and instruction formats - addressing modes - instruction sets - I/O and programming.

UNIT II ASSEMBLERS 9

Basic assembler functions - A simple SIC assembler – Assembler algorithm and data structures - Machine dependent assembler features - Instruction formats and addressing modes – Program relocation - Machine independent assembler features - Literals – Symbol-defining statements – Expressions - One pass assemblers and Multi pass assemblers - Implementation example - 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 within Macro-Implementation example - MASM Macro Processor – ANSI C Macro language.

Text editors - Overview of the Editing Process - User Interface – Editor Structure. - Interactive debugging systems - Debugging functions and capabilities – Relationship with other parts of the system – User-Interface Criteria.

TEXT BOOK

1. Leland L. Beck, “System Software – An Introduction to Systems Programming”, 3rd Edition, Pearson Education Asia, 2000.

REFERENCES BOOKS

1. D. M. Dhamdhere, “Systems Programming and Operating Systems”, Second Revised Edition, Tata McGraw-Hill, 1999.
2. John J. Donovan “Systems Programming”, Tata McGraw-Hill Edition, 1972.

SEMESTER III
ELECTRONIC DEVICES AND CIRCUITS
(COMMON FOR III SEM CSE AND III SEM IT)

- | | |
|---|----|
| 1. SEMI CONDUCTOR DEVICES | 7 |
| Semiconductor-PN Junction diode –BJT-FET-SCR-VI characteristics (qualitative Treatment only) | |
| 2. AMPLIFIERS | 12 |
| Transistor biasing – self biasing –DC and AC analysis of CE, CB and CC amplifiers. Power amplifiers-Efficiency. Tuned Amplifiers-Frequency Response. | |
| 3. FEED BACK AMPLIFIERS AND OSCILLATORS: | 10 |
| Negative feed back- Types feedback with examples for each type. Effect of feedback on ac characteristic of amplifiers. Positive Feedback – oscillators-Analysis of RC Phase shift Oscillator and LXC oscillators – Hardly and colpitt. | |
| 4. MULTIVIBRATORS AND TIMERS | 7 |
| Bistable, Monostable and a stable multivibrators using Transistors-triggering delay and frequency Calculation.555 Timer-Internal Block-Application. | |
| 5. OPERATIONAL AMPLIFIER AND APPLICATION | 9 |
| Operational amplifier- Characteristics-Block diagram only application of op-amp-Current to voltage, Voltage to current converters, Arithmetic circuits-Adder, Subtractor, multiplier, differentiator And Integrator. Active Filters - Butter worth and Chebyshev. | |

Total Hours 45

TEXT BOOKS

1. Electronic Device – Floyd -Fifth Edition – Addison Wesley Long man Pt. Ltd Branch 2001.
2. David A. Bell Electronic Devices and Circuits –Prentice Hall of India.

REFERENCES

1. Milman and Halkiad, Integrated Electronic McGraw Hill publishers 1985.
2. Boyle stad Nashelsky - Electronic devices and Circuit Theory-Prentice hall of India Pvt. 6th edition.

SEMESTER III

OBJECT ORIENTED PROGRAMMING AND DATA STRUCTURES LAB

(i) OBJECT ORIENTED PROGRAMMING LAB

C++

1. Programs Using Functions
 - Functions with default arguments
 - Implementation of Call by Value, Call by Address and Call by Reference
2. Simple Classes for understanding objects, member functions and Constructors
 - Classes with primitive data members
 - Classes with arrays as data members
 - Classes with pointers as data members – String Class
 - Classes with constant data members
 - Classes with static member functions
3. Compile time Polymorphism
 - Operator Overloading including Unary and Binary Operators.
 - Function Overloading
4. Runtime Polymorphism
 - Inheritance
 - Virtual functions
 - Virtual Base Classes
 - Templates
5. File Handling
 - Sequential access
 - Random access

(ii) DATA STRUCTURES LAB

AIM

To teach the principles of good programming practice and to give a practical training in writing efficient programs in C

OBJECTIVES

- To teach the students to write programs in C
- To implement the various data structures as Abstract Data Types
- To write programs to solve problems using the ADTs

Implement the following exercises using C:

1. Array implementation of List Abstract Data Type (ADT)
2. Linked list implementation of List ADT
3. Cursor implementation of List ADT
4. Array implementations of Stack ADT
5. Linked list implementations of Stack ADT

The following three exercises are to be done by implementing the following source files

- (a) Program for 'Balanced Paranthesis'
- (b) Array implementation of Stack ADT
- (c) Linked list implementation of Stack ADT
- (d) Program for 'Evaluating Postfix Expressions'

An appropriate header file for the Stack ADT should be #included in (a) and (d)

- 6. Implement the application for checking 'Balanced Paranthesis' using array implementation of Stack ADT (by implementing files (a) and (b) given above)
- 7. Implement the application for checking 'Balanced Paranthesis' using linked list implementation of Stack ADT (by using file (a) from experiment 6 and implementing file (c))
- 8. Implement the application for 'Evaluating Postfix Expressions' using array and linked list implementations of Stack ADT (by implementing file (d) and using file (b), and then by using files (d) and (c))
- 9. Queue ADT
- 10. Search Tree ADT - Binary Search Tree
- 11. Heap Sort
- 12. Quick Sort

SEMESTER III

Electronic Devices and Digital Lab

(COMMON FOR III SEM CSE AND III SEM IT)

1. PN junction diode Characteristics
2. CE Transistor Characteristics
3. FET Characteristics
4. Frequency Response of CE Amplifiers with self bias and fixed bias.
5. Differential Amplifiers, CMRR measurements
6. Astable and monostable Multivibrator
7. IC 555 Timer
8. Differentiator/Integrator
9. Study of logic gates.
10. Adder/Subtractor
11. Study of Flip – Flops – using gates – Flip – Flop ICs.
12. Multiplexer/Demultiplexer
13. Counters
14. Shift Registers

