ANNAMALAI UNIVERSITY

214 - B. Sc. Computer Science

Programme Structure and Scheme of Examination (under CBCS)

(Applicable to the candidates admitted in Affiliated Colleges from the academic year 2022 -2023 onwards)

Course Code	Part	Study Components & Course Title	Hours/ Week	Credit	Maximum Marks		
					CIA	ESE	Total
		SEMESTER – I					
22UTAML11	I	**Language Course - I : Tamil/Other Languages**	5	3	25	75	100
22UENGL12	II	**English Course - I : Communicative English I**	5	3	25	75	100
22UCSCC13	III	Core Course -I : Fundamentals of Computers	5	4	25	75	100
22UCSCC14		Core Course – II : Programming in C	5	4	25	75	100
22UCSCP15		Core Practical – I : Programming in C Lab	3	2	40	60	100
		Allied Course - I : Mathematics – I /Mathematical foundation-I	5	4	25	75	100
22UENVS18	IV	**Environmental Studies**	2	2	25	75	100
		Total		**22**			**700**
		SEMESTER – II					
22UTAML21	I	**Language Course - II : Tamil/Other Languages**	5	3	25	75	100
22UENGL22	II	**English Course - II : Communicative English II**	5	3	25	75	100
22UCSCC23	III	Core Course – III : Programming with C++	5	4	25	75	100
22UCSCC24		Core Practical – II : Programming with C++ Lab	3	2	40	60	100
		Allied Course - I : Paper -2 : Mathematics II/Mathematical Foundation-II	5	4	25	75	100
22UCSCE26		Internal Elective – I	3	3	25	75	100
22UVALE27	IV	**Value Education**	2	1	25	75	100
22USOFS28		**Soft Skill**	2	1	25	75	100
		Total		**21**			**800**

Internal Elective Courses

22UCSCE26-1	Internal Elective – I	Digital logic fundamentals
22UCSCE26-2		Fundamental of Algorithms
22UCSCE26-3		System Software

Allied Courses

22UCSCA16	Theory	Mathematics-I/Mathematical Foundation I
22UCSCA25	Theory	Mathematics-II/Mathematical Foundation II

Allied Courses offered by Computer Science Department to Other Departments

22UCSCA01	Theory	Basics of Computers
22UCSCA02	Theory	Web Technology
22UCSCAP1	Practical	Web Technology Lab
22UCSCA03	Theory	Data Mining
22UCSCA04	Theory	Management Information System

SEMESTER: I PART-III	COURSE CODE: 22UCSCC13 COURSE TITLE: FUNDAMENTALS OF COMPUTERS	CREDIT:4 HOURS: 5/W

LEARNING OBJECTIVES

1. An understanding of basic concepts of computer science.
2. An introduction to the fundamentals of hardware, software and programming.
3. To understand the concept of Number System.
4. To know the types of memory for storage purpose.
5. To understand the types of input devices to feed the data for action.

Unit I: Introduction to Computer **Hours:10**

Introduction – Types of computers – Characteristics of Computers. Generations of Computers: First Generation – Second Generation – Third Generation – Fourth Generation – Fifth Generation. Classification of Digital Computers: Introduction – Microcomputers – Personal Computer – Portable Computers – Mini Computers – Super Computers – Main Frames.

Unit II: Number System **Hours:15**

Introduction – Decimal Number System – Binary Number System – Binary-Decimal Conversion – Decimal Binary Conversion – Binary Addition – Binary Subtraction – Complements – 9's Complement – 10's Complement – 1's Complements – 2's Complements – BCD - Bits, Bytes, Words – Octal – Hexadecimal Number System.

Unit III: Anatomy of Digital Computer **Hours:10**

Functions and Components of Computer – Central Processing Unit – Control Unit – Arithmetic – Logic Unit – Memory – Registers – Addresses. Memory Units: RAM, ROM, PROM, EPROM, EEPROM, and Flash Memory.

Unit IV: Input Devices **Hours:10**

Introduction – Keyboard – Mouse – Types of Mice – Connections – Mouse pad – Trackball – joystick – Digitizing Tablet – Scanners – Digital Camera – MICR – OCR – OMR – Bar Code Reader – Speech Input Device- Touch Screen – Touch Pad – Light Pen. Output Devices: Introduction – Monitor – Classification of Monitors – Monochrome – Gray Scale – Color – Digital Monitor – Analog Monitor – Characteristics of monitor – Printers.

Unit V: Computer Software **Hours: 15**

Introduction – Operating System – Utilities – Compiler and Interpreters – Word Processor – Spreadsheets – Presentation Graphics – DBMS – Programming Languages: Machine Language – Assembly Language – High level language – Types of HighLevel Languages. Data Processing: Data VS Information – File Processing – Sequential File Processing – Direct Access file Processing.

COURSE OUTCOMES

1. Explain the needs of hardware and software required for a computation task.
2. Can have the knowledge about the generations of computers.
3. Understand the concept of output device.
4. Having the skill about the various types of languages.
5. Understand the concept of file processing.

Text Books:

1. Alexis Leon and Mathews Leon, ―Fundamentals of Computer Science and Communication Engineering‖, Leon Tech world, 1998.

Supplementary Readings

1. B Ram and Sanjay Kumar, ―Computer Fundamentals‖, 5th Edition, New Age International Publishers, 2014.
2. Pradeep K Sinha, Priti Sinha, ―Computer Fundamentals‖, BPB Publications, 2004. Anita Goel, ―Computer Fundamentals‖, 1st Edition, Pearson Education India, 2010.
3. Anita Goel, Computer Fundamentals, Pearson Publication.

PROGRAMME OUTCOMES AND COURSE OUTCOMES MAPPING TABLE

CO/PO	PO1	PO2	PO3	PO4	PO5
CO1	2	3	2	2	3
CO2	3	2	2	3	2
CO3	3	3	2	2	3
CO4	3	2	3	3	2
CO5	2	2	3	2	2

1-LOW 2- MODERATE 3-HIGH

SEMESTER: I PART-III	COURSE CODE: 22UCSCC14 COURSE TITLE: PROGRAMMING IN C	CREDIT: 4 HOURS: 5/W

LEARNING OBJECTIVES

1. To Provide complete knowledge of C language
2. Students will be able to develop logics which will help them to create programs,applications in C
3. By learning the basic programming constructs they can easily switch over to any other language in future.
4. To understand the concept of function types
5. To acquire knowledge about pointers.

Unit I : Overview of C **Hours: 15**

History of C – Importance of C – Basic Structure of C Programs – Programming Style – Character Set – C Tokens – Keywords and Identifiers – Constants, Variables and Data Types – Declaration of Variables – Defining Symbolic Constants – Declaring a variable as a constant – overflow and underflow of data – Operators and Expressions: Arithmetic, relational, logical, assignment operators – increment and decrement operators, conditional operators, bitwise operators, special operators – Arithmetic Expressions- Evaluation of Expressions – Precedence of Arithmetic Operators – Type Conversions in Expressions – Operator Precedence and Associativity – Mathematical functions.

Unit II: Managing I/O Operations **Hours:10**

Reading and Writing a Character – Formatted Input, Output – Decision Making & Branching: if statement - if else statement - nesting of if else statements - else if ladder – switch statement – the ?: operator – goto statement – the while statement – do statement – the for statement – jumps in loops.

Unit III: Arrays **Hours:10**

One-Dimensional Arrays – Declaration, Initialization – Two Dimensional Arrays – Multi-dimensional Arrays – Dynamic Arrays –Initialization. Strings: Declaration, Initialization of string variables – reading and writing strings – string handling functions

Unit IV: User-defined functions **Hours: 10**

Need – multi-function programs – elements of user defined functions – definition – return values and their types – function calls, declaration, category – all types of arguments and return values – nesting of functions – recursion – passing arrays, strings to functions – scope visibility and life time of variables. Structures and Unions: Defining a structure – declaring a structure variable – accessing structure members – initialization – copying and comparing – operation on individual members – array of structures – arrays within structures – structures within structures – structures and functions – unions – size of structures – bit fields

Unit V: Pointers **Hours: 15**

Understanding Pointers, Accessing the address of a variable – declaring, initialization of pointer variables – accessing a variable through its pointer – chain of pointers – pointer increments and scale factors – pointers and character strings – pointers as function arguments – pointers and structures. Files: Defining, opening, closing a file – IO Operations on files – Error handling during IO operations – command line arguments.

COURSE OUTCOMES

1. To understand the concepts of data types and operators
2. To analyze the usages of the various programming constructs and functions
3. To interpret the importance of arrays and pointers
4. To identify the purpose of structures, unions, macros and bit fields
5. To develop programs using dynamic memory allocation and data file operations

Text Books:

1. E.Balagurusamy, Programming in ANSI C, 7 the Edition, Tata McGraw Hill Pub,2017

Supplementary Readings:

1. Ashok N.Kamthane , Programming with ANSI and Turbo C , Pearson Education, 2006
2. Kanetkar Y., Let us C, BPB Pub., New Delhi, 1999.
3. T.Prabhu, C Programming Made Easy, Kanthimathi Publications

PROGRAMME OUTCOMES AND COURSE OUTCOMES MAPPING TABLE

CO/PO	PO1	PO2	PO3	PO4	PO5
CO1	2	3	2	2	3
CO2	3	2	2	3	2
CO3	3	2	2	2	2
CO4	3	2	3	3	3
CO5	2	2	3	2	2

1-LOW 2- MODERATE 3-HIGH

SEMESTER: I PART-III	COURSE CODE: 22UCSCP15 COURSE TITLE: PROGRAMMING IN C LAB	CREDIT: 2 HOURS: 3/W

LEARNING OBJECTIVES

1. To Develop Programs In C Using Basic Constructs.
2. Familiarize The Different Control And Decision Making Statements In "C"
3. Build Programs Using Arrays And Strings.
4. Provide Knowledge On Working With Files And Functions.
5. To Understand The Concepts Of Structures.

LIST OF PROGRAMS **HOURS:45**

1. C Program Swap Numbers in Cyclic Order Using Call by Reference
2. C Program to Remove all Characters in a String Except Alphabets
3. C Program to Sort Elements in Lexicographical Order (Dictionary order)
4. C Program to Calculate Standard Deviation of 10 numbers stored in an array.
5. C Program to Add Two Matrices Using Multi-dimensional Arrays.
6. C Program to Find Largest Number Using Dynamic Memory Allocation
7. C Program To Convert Binary Number To Decimal
8. C Program to Add Two Distances (in inch-feet system) using Structures
9. C Program to Check Whether a Number can be Expressed as Sum of Two Prime Numbers.
10. C Program to Make a Simple Calculator Using switch...case.
11. C Program to Display (i) Fibonacci Sequence (ii) Factorial of a given number.
12. C Program to find odd or even numbers using files.

COURSE OUTCOMES:

1. Demonstrate knowledge on C programming constructs.
2. Study all the Basic Statements in C Programming.
3. Practice the usage of branching and looping statements.
4. Apply string functions and arrays usage.
5. Analysis the use of files and structures.

PROGRAMME OUTCOMES AND COURSE OUTCOMES MAPPING TABLE

CO/PO	PO1	PO2	PO3	PO4	PO5
CO1	2	3	2	2	3
CO2	2	2	2	3	2
CO3	3	3	2	2	3
CO4	2	2	3	3	2
CO5	2	2	3	2	2

1-LOW 2- MODERATE 3-HIGH

YEAR- I SEMESTER - II PART-III	COURSE CODE:22UCSCC23 COURSE TITLE:PROGRAMMING WITH C++	HRS/WK – 5 CREDIT – 4

LEARNING OBJECTIVES:

This course enables the students to know about:

1. Object Oriented concepts,C++ language features.
2. Classes, Objects,Inheritance, and Polymorphism.
3. Functions, Constructors, Streams and Files.

UNIT I: (15Hrs)

Principles of object oriented programming: Basic concepts of object oriented programming – Benefits of OOPs – Applications of OOPs – Beginning with C++: C++ introduction – Applications of C++ – C++ statements – Structure of C++ program. Tokens, Expressions and Control structures: Tokens – Keywords – Identifiers – Constants – Operators in C++ - Manipulators – Expressions and their types – Basic and user defined data types – operators in C++ – Operator overloading – Operator precedence – Control structures.

UNIT II : (10Hrs)

Functions in C++: The main functions – Function prototyping – Call by reference – Return by reference – Inline functions– Default arguments - Function overloading – Friend & Virtual Functions – Math Library functions. Classes and Objects: Specifying a class– Defining member function– Nesting of member functions– Private member functions– Arrays within a class – Static data members – Static member functions – Array of objects – Objects as function arguments – Friendly functions – Returning objects - Pointers to members.

UNIT III : (10Hrs)

Constructors and Destructors: Constructors – Parameterized constructors – Multiple constructors in a class – Constructors with default arguments – Copy constructors – Dynamic constructors – Destructors. Operator overloading: Defining – Overloading Unary, Binary operators – Manipulation of strings using operators - Type conversions.

UNIT IV: (10Hrs)

Inheritance: Defining derived classes – Single Inheritance – Multilevel Inheritance – Multiple Inheritance – Hierarchical Inheritance – Hybrid Inheritance – Virtual Base Classes – Abstract classes – Constructors in Derived Classes. Pointers, Virtual functions and Polymorphism: Pointers – Pointers to Objects, this Pointer – Virtual functions.Exception handling: Basics – Exception handling mechanism – Throwing and catching exception.

UNIT V : (15Hrs)

Managing console I/O operations: C++ streams – C++ stream classes – Unformatted I/O operations – Formatted console I/O operations – Managing output with manipulators. Working with files: classes for file stream operations – opening and closing a file – Detecting End – of - File – File Modes – File pointers and manipulation - Sequential I/O operations – Random access - Error handling during file operations, Command-line arguments.

Course outcomes:

1. Able to understand OOPs concept,C++ language features.
2. Able to understand and apply the concepts of Classes &Objects,friend function, constructors and destructors in program design.
3. Able to design & implement various forms of inheritance, and String classes.
4. Able to apply and analyze operator overloading, and runtime polymorphism.
5. Able to analyze and explore various Stream classes, I/O operations and Exception handling.

Text Book:

1. E.Balagurusamy, Object Oriented Programming with C++, Tata McGraw Hill Publications, 8th Edition, 2020.

Supplementary Readings

1. BjarneStroustrup, The C++ Programming Language, Pearson Education, 4th Edition, 2014.
2. Rajesh K.Shukla, Object Oriented Programming in C++, Wilsey India Pvt. Ltd., 1st edition, 2008.
3. Robert Lafore, Object Oriented Programming in C++, Galgotia Publications Pvt. Ltd., 4th edition, 2001.
4. Tony Gaddis, Judy Walfers, and GodferyMuganda, Starting Out with C++: Early Objects, Addison-Wesley publication, 8th Edition, 2013.

PROGRAMME OUTCOMES AND COURSE OUTCOMES MAPPING TABLE

CO/PO	PO1	PO2	PO3	PO4	PO5
CO1	2	3	2	2	3
CO2	3	2	3	3	2
CO3	2	3	2	2	2
CO4	2	2	3	3	2
CO5	2	2	3	2	3

1-LOW 2- MODERATE 3-HIGH

YEAR-I SEMESTER -II PART-III	COURSE CODE:22UCSCP24 COURSE TITLE: PROGRAMMING WITH C++ LAB	HRS/WK – 3 CREDIT – 2

LEARNING OBJECTIVES:

The objectives of the course are to have students :

1. Identify and practice the object-oriented programming concepts and techniques,
2. Practice the use of C++ classes and class libraries, arrays, vectors, inheritance and file I/O stream concepts.

LIST OF PROGRAMS **HOURS: 45**

1. Write a C++ program to find sum of digits of a given number.
2. Write a C++ program to demonstrate the use of Constructors.
3. Write a C++ program to perform Overloading of a Binary Operator.
4. Write an OOP Program to demonstrate the importance of Multilevel inheritance.
5. Write an OOP program to demonstrate the Function overloading.
6. Write a C++ program to find the sum of the given variables using Function with Default arguments.
7. Write a C++ program to demonstrate the use array of Objects.
8. Write a C++ program to handle the Exceptions.
9. Write a C++ program to perform Formatted console operations.
10. Write a C++ program to copy the content of one Text file into another text file.

COURSE OUTCOMES:

Ability to:

1. Creating simple programs using classes and objects in C++.
2. Implement Object Oriented Programming Concepts in C++.
3. Develop applications using stream I/O and file I/O.
4. Implement simple graphical user interfaces.
5. Implement Object Oriented Programs using templates and exceptional handling concepts.

PROGRAMME OUTCOMES AND COURSE OUTCOMES MAPPING TABLE

CO/PO	PO1	PO2	PO3	PO4	PO5
CO1	3	3	2	2	3
CO2	3	2	3	3	2
CO3	2	3	2	2	2
CO4	3	2	3	3	2
CO5	2	2	3	2	3

1-LOW 2- MODERATE 3-HIGH

INTERNAL ELECTIVE – I

YEAR-I SEMESTER -II PART-III	COURSE CODE:22UCSCE26A COURSE TITLE: DIGITAL LOGIC FUNDAMENTALS	HRS/WK – 3 CREDIT – 3

OBJECTIVE:

To Understand the basic concepts of Digital Circuits and Logic design of Computers.

Unit-I: Number Systems **HOURS:9**

Digital Computers and Digital Systems - Binary Numbers – Number Base Conversions – Octal and Hexadecimal Numbers – Complements – Signed Binary Numbers – Binary Codes – Binary Storage and Registers – Binary Logic.

Unit-II: Boolean Algebra and Logic Gates **HOURS:9**

Axiomatic Definition of Boolean algebra - Basic Theorems and Properties of Boolean Algebra – Boolean Functions – Canonical and Standard Forms – Other Logic operations – Digital Logic Gates – Integrated Circuits.

Unit-III: Simplification of Boolean Functions **HOURS:9**

The Map Method – Two and Three Variable Maps – Product of Sums Simplification - NAND and NOR Implementation - Other Two-Level Implementations - Don't Care Conditions - The Tabulation Method - Determination of Prime Implicants - Selection of Prime Implicants.

Unit-IV: Combinational Logic **HOURS:9**

Design Procedure – Adders – Subtractors – Code Conversion – Analysis Procedure – Multilevel NAND Circuits – Multilevel NOR Circuits – Exclusive OR Functions.

UNIT-V: Sequential Circuits **HOURS:9**

Flip Flops – Triggering of Flip-Flops – Analysis of Clocked Sequential Circuits – State Reduction and Assignment – Flip-Flop Excitation Tables – Design Procedure – Design of Counters.

COURSE OUTCOMES:

1. To Learn the basic design of Computers, Number Systems and Binary Codes.
2. To understand the Boolean algebra and the Logic Gates Operations.
3. To Learn and practice the K-Map Simplifications.
4. To study the Design Procedure of Adders, Subtractors and Multilevel Circuits.
5. To understand Flipflops, its types and the design of Counters.

Text Books:

1. M. Morris Mano, Digital Logic and Computer Design - PHI, 2ndEdition -2006

Supplementary Readings

1. Louis Neshelsky, Introduction to Digital Technology , John Wiley & Sons, Third Edition, 1983.
2. Dr. K. Meena, Principles of Digital Electronics, PHI Learning Private Limited, New Delhi - 1st Edition-2009.
3. Norman Balabanian, Bradley Carlson **,"**Digital Logic Design Principles" **-** -John Wiley & Sons, Inc 1 Edition1996

PROGRAMME OUTCOMES AND COURSE OUTCOMES MAPPING TABLE

CO/PO	PO1	PO2	PO3	PO4	PO5
CO1	2	3	2	2	3
CO2	3	2	3	3	2
CO3	3	3	2	2	3
CO4	2	2	3	3	2
CO5	2	2	3	2	2

1-LOW 2- MODERATE 3-HIGH

YEAR-I SEMESTER -II PART-III	COURSE CODE:22UCSCE26B COURSE TITLE: FUNDAMENTALS OF ALGORITHMS	HRS/WK – 3 CREDIT – 3

LEARNING OBJECTIVES:

1. To know how to analyze the performance of algorithms.
2. To understand how the choice of data structures and algorithm design methods impacts the performance of programs.
3. To solve problems using algorithm design methods such as the greedy method, divide and conquer, dynamic programming, and backtracking.

Unit I: (9 Hrs)

Introduction to algorithm, reason for the analysis algorithms, Goal, Running time analysis, Compare Algorithms, Rate of Growth, Commonly Used Rate of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega-Ω Notation, Theta-Θ Notation, Asymptotic Analysis, Properties of Notations, Commonly used Logarithms and Summations, Amortized analysis.

Unit II: (9 Hrs)

Recursion and Back tracking: Recursion – importance – Format of a Recursive function – Recursion and Memory – Recursion versus Iteration – Algorithms for Recursion – Backtracking – Algorithms for Back tracking.

Tree algorithms: Tree – Binary tree – Types and properties of binary tree – Binary tree traversals – Threaded Binary tree traversals – Binary search trees – Balanced Binary search trees – AVL Trees.

Unit III: (9 Hrs)

Graph Algorithms: Introduction - Applications of Graphs - Graph Representation - Graph Traversals - Topological Sort - Shortest Path Algorithms - Minimal Spanning Tree.

Sorting algorithms: Sorting – importance – Classification of Algorithms – Bubble sort – Selection sort – Insertion sort – Merge sort – Heap sort – Quick sort – External sorting.

Unit IV: (9 Hrs)

Searching: Importance – types – Unordered linear search – Ordered linear search – Binary search – comparing basic searching algorithms – String searching algorithms.

Greedy Algorithms: Introduction, Greedy Strategy, Elements of Greedy Algorithms, Advantages and disadvantages of Greedy Method, Greedy Applications, Understanding Greedy Technique.

Unit V: (9 Hrs)

Divide and Conquer Algorithms: Introduction - Divide and Conquer Strategy - Divide and Conquer Visualization - Understanding Divide and Conquer - Advantages of Divide and Conquer - Disadvantages of Divide and Conquer - Divide and Conquer Applications.

Dynamic Programming: Introduction - Dynamic Programming Strategy - Properties of Dynamic Programming Strategy - Problems which can be solved using Dynamic Programming - Dynamic Programming Approaches - Examples of Dynamic Programming Algorithms - Understanding Dynamic Programming - Longest Common Subsequence.

COURSE OUTCOMES:

1. 1.To learn the method of analysing algorithms.
2. To understand Recursion and backtracking principles.
3. To gain knowledge on the tree and graph algorithms.
4. To understand the sorting and searching algorithms.
5. To learning the working principles of Greedy, Divide-and-Conquer and Dynamic programming algorithms.

Text book:

1. Narasimha Karumanchi, Data Structures and Algorithms Made Easy: Data Structure and Algorithmic Puzzles, CareerMonk Publications, 2017.

Supplementary Readings

1. Cormen, Thomas H, and Thomas H. Cormen. Introduction to Algorithms. Cambridge, Mass: MIT Press, 2001.
2. Aho, Ullman & Hopcroft, Data Structures and Algorithms, Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. Data Structures and Algorithmus. Addison-Wesley, 2009.
3. Ellis Horowitz and Sartaj Sahni, Fundamentals of Computer Algorithms, Computer Science Press, 2007.

PROGRAMME OUTCOMES AND COURSE OUTCOMES MAPPING TABLE

CO/PO	PO1	PO2	PO3	PO4	PO5
CO1	2	3	2	2	3
CO2	3	2	2	3	2
CO3	3	2	2	2	3
CO4	2	2	3	3	2
CO5	2	2	3	3	2

1-LOW 2- MODERATE 3-HIGH

SEMESTER: II PART: III	COURSE CODE:22UCSCE26C COURSE TITLE: SYSTEM SOFTWARE	CREDIT: 3 HOURS: 3/W

COURSE OBJECTIVES

1. To understand the relationship between system software and machine architecture.
2. To know the design and implementation of assemblers
3. To know the design and implementation of linkers and loaders.
4. To have an understanding of macroprocessors.
5. To have an understanding of system software tools.

Unit I INTRODUCTION **Hours: 8**

System software and machine architecture – The Simplified Instructional Computer (SIC) – The SIC/XE Machine Architecture – SIC Programming Examples.

Unit II ASSEMBLERS **Hours: 10**

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 **Hours: 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 **Hours: 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.

Unit V SYSTEM SOFTWARE TOOLS **Hours: 9**

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.

COURSE OUTCOMES

1. Understand the relationship between system software and machine architecture.
2. Know the design and implementation of assemblers
3. Know the design and implementation of linkers and loaders.
4. Understanding of macroprocessors and its implementation.
5. Understanding of system software tools

Text Books

1. Leland L. Beck (2006). System Software – An Introduction to Systems Programming (3rd Edition). Pearson Education Asia.

Supplementary Readings

1. D. M. Dhamdhere (2000). Systems Programming and Operating Systems (2nd Revised Edition). Tata McGraw-Hill.
2. John J. Donovan (2000). Systems Programming. Tata McGraw-Hill Edition.
3. John R. Levine (2000). Linkers & Loaders – Harcourt India Pvt. Ltd., Morgan Kaufmann Publishers.

PROGRAMME OUTCOMES AND COURSE OUTCOMES MAPPING TABLE

CO/PO	PO1	PO2	PO3	PO4	PO5
CO1	2	3	2	2	3
CO2	2	2	2	3	2
CO3	2	3	2	2	3
CO4	3	2	3	3	2
CO5	2	2	3	2	2

1-LOW 2- MODERATE 3-HIGH