ANNAMALAI UNIVERSITY

215 - B.Sc. INFORMATION TECHNOLOGY

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
22UINTC13	III	Core Course – I: Introduction to Information Technology	5	4	25	75	100
22UINTC14		Core Course – II : Programming in C	5	4	25	75	100
22UINTP15		Core Practical – I : Programming in C Lab	3	2	40	60	100
		Allied Course - I : Paper -1 Mathematics-1/ Mathematical Foundation -1	5	4	25	75	100
22UENVS18	IV	Environmental Studies	2	2	25	75	100
		Total	**30**	**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
22UINTC23	III	Core Course – III : Programming with C++	5	4	25	75	100
22UINTP24		Core Practical – II : Programming with C++ Lab	3	2	40	60	100
		Allied Course - I : Paper -2 Mathematics-2/ Mathematical Foundation -2	5	4	25	75	100
22UINTE26		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	**30**	**21**			**800**

Internal Elective Courses

22UINTE26-1	Internal Elective - I	Digital Logic Fundamentals
22UINTE26-2		System Software
22UINTE26-3		Software Project Management

SEMESTER – I

SEMESTER: I PART: III	COURSE CODE:22UINTC13 COURSE TITLE: INTRODUCTION TO INFORMATION TECHNOLOGY	CREDIT:3 HOURS:5/W

COURSE OBJECTIVES

1. To introduce IT in a simple language to all undergraduate students, regardless of their specialization.
2. Help them to pursue specialized programs leading to technical and professional careers.
3. Enhances certifications in the IT industry.
4. Introducing skills relating to IT basics, computer applications, programming.
5. A glimpse on various types of software.

Unit I : Introduction to Computers: Hours: 12

Introduction, Definition, .Characteristics of computer, Evolution of Computer, Block Diagram Of a computer, Generations of Computer, Classification Of Computers, Applications of Computer, Capabilities and limitations of computer.

Unit II: Basic Computer Organization: Hours:12

Role of I/O devices in a computer system. Input Units: Keyboard, Terminals and its types. Pointing Devices, Scanners and its types, Voice Recognition Systems, Vision Input System, Touch Screen, Output Units: Monitors and its types. Printers: Impact Printers and its types. Non Impact Printers and its types, Plotters, types of plotters, Sound cards, Speakers.

Unit III: Storage Fundamentals: Hours:12

Primary Vs Secondary Storage, Data storage & retrieval methods. Primary Storage: RAM ROM, PROM, EPROM, EEPROM. Secondary Storage: Magnetic Tapes, Magnetic Disks. Cartridge tape, hard disks, Floppy disks Optical Disks, Compact Disks, Zip Drive, Flash Drives.

Unit IV: Computer Arithmetic: Hours:12

Binary, Binary Arithmetic, Number System: Positional & Non Positional, Binary, Octal, Decimal, Hexadecimal, Converting from one number system to another , Converting from one number system to another , Converting from one number system to another.

Unit V: Hours: 12

Software: Software and its needs, Types of S/W. System Software: Operating System, Utility Programs Programming Language: Machine Language, Assembly Language, High Level Language their advantages & disadvantages. Application S/W and its types: Word Processing, Spread Sheets Presentation, Graphics, DBMS s/w.

Operating System: Functions, Measuring System Performance, Assemblers, Compilers and Interpreters. Batch Processing, Multiprogramming, Multi Tasking, Multiprocessing, Time Sharing, DOS, Windows, Unix/Linux.

COURSE OUTCOMES

1. Understand basic concepts and terminology of information technology.
2. Have a basic understanding of personal computers and their operations.
3. Be able to identify issues related to information security.
4. Knowledge of computers, their operations and applications.
5. Enhancing the concepts of operating systems and software.

Text Books

1. P.K.Sinha,Priti Sinha,Computer Fundamentals,BPB Publications.

Supplementary Readings

1. V.Rajaraman,Fundamentals of Computers,Sixth edition,PHP.
2. Anita Goel,Computer Fundamentals,Pearson.

OUTCOME MAPPING

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

1-LOW 2- MODERATE 3-HIGH

SEMESTER: I PART - III	COURSE CODE: 22UINTC14 COURSE TITLE: POGRAMMING IN C	CREDIT: 4 HOURS:5/WEEK

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: 12**

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:12**

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:12**

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: 12**

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: 12**

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	3	3	2	2	3
CO2	2	2	2	3	2
CO3	3	2	2	2	2
CO4	3	2	3	2	2
CO5	2	3	3	2	2

1-LOW 2- MODERATE 3-HIGH

SEMESTER: I **PART - III**	**COURSE CODE: 22UINTP15** **COURSE TITLE: PROGRAMMIN IN C LAB**	**CREDIT: 2** **HOURS:3/WEEK**

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 EXERCISES: **HOURS: 36**

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	3	3	2	2	3
CO2	2	2	2	3	2
CO3	3	2	2	2	2
CO4	3	2	3	3	3
CO5	2	2	2	2	2

1-LOW 2- MODERATE 3-HIGH

SEMESTER: II PART - III	COURSE CODE: 22UINTC23 COURSE TITLE: PRROGRAMMING WITH C++	CREDIT: 4 HOURS:5/WEEK

LEARNING OBJECTIVES

This course enables the students to understand:

1. Object Oriented concepts, C++ language features.
2. Classes, Objects, Inheritance, Polymorphism, and Exception handling.
3. Functions, Function / Operator overloading, and Constructors.
4. Streams, I/O operations and Files.

UNIT I : (12 Hrs)

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 : (12 Hrs)

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 : (12 Hrs)

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: (12 Hrs)

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 : (12 Hrs)

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 arugments.

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 Godfery Muganda, Starting Out with C++: Early Objects, Addison-Wesley publication, 8th Edition, 2013.

COURSE OUTCOMES :

1. Able to apply OOPs concepts.
2. Able to use 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.

OUTCOME MAPPING

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

1-LOW 2- MODERATE 3-HIGH

SEMESTER: II PART - III	COURSE CODE: 22UINTP24 COURSE TITLE: PRROGRAMMING WITH C++ LAB	CREDIT: 2 HOURS:3/WEEK

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.

OUTCOMES MAPPING

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

1-LOW 2- MODERATE 3-HIGH

INTERNAL ELECTIVE-I

SEMESTER: II	**COURSE CODE: 22UINTE26-1**	**CREDIT: 3**
PART - III	**COURSE TITLE: DIGITAL LOGIC FUNDAMENTALS**	**HOURS:3/WEEK**

LEARNING OBJECTIVES:

1. To Understand the basic concepts of Digital Circuits and Logic design of Computers
2. To practice and learn the Boolean functions and k-map simplifications.
3. To study the various Circuit models and its use in computers.

Unit-I: Number Systems HOURS: 8

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 : 10

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 (CO):

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, http://www.amazon.in/Norman-Balabanian/e/B001HD3IJY/ref%3Ddp_byline_cont_book_1John Wiley & Sons, Inc 1 Edition 1996

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	3	2	2
CO4	3	3	3	3	3
CO5	2	2	3	2	2

1-LOW 2- MODERATE 3-HIGH

SEMESTER: II PART: III	COURSE CODE: 22UINTE26-2 COURSE TITLE: SYSTEM SOFTWARE	CREDIT: 3 HOURS: 3/WEEK

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	3	3	2	2	3
CO2	2	3	2	3	2
CO3	3	3	2	2	2
CO4	2	2	2	3	2
CO5	2	2	3	3	3

1-LOW 2- MODERATE 3-HIGH

SEMESTER: II PART: III	COURSE CODE: 22UINTE26-3 COURSE TITLE: SOFTWARE PROJECT MANAGEMENT	CREDIT:3 HOURS:3/WEEK

COURSE OBJECTIVES

1. Define and highlight importance of software project management.
2. Describe the software project management activities.
3. Train software project managers and other individuals involved in software project.
4. To ensure the concepts on Planning and tracking .
5. Oversight in the implementation of the software project management process.

Unit I : Introduction and Software Project Planning Hours: 9

Fundamentals of Software Project Management (SPM), Need Identification, Vision and Scope document, Project Management Cycle, SPM Objectives, Management Spectrum, SPM Framework, Software Project Planning, Planning Objectives, Project Plan, Types of project plan, Structure of a Software Project Management Plan, Software project estimation, Estimation methods, Estimation models, Decision process.

Unit II: Project Organization and Scheduling Hours:9

Project Elements, Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Project Life Cycle and Product Life Cycle, Ways to Organize Personnel, Project schedule, Scheduling Objectives, Building the project schedule, Scheduling terminology and techniques, Network Diagrams: PERT, CPM, Bar Charts: Milestone Charts, Gantt Charts.

Unit III: Project Monitoring and Control Hours:9

Dimensions of Project Monitoring & Control, Earned Value Analysis, Earned Value Indicators: 23 Budgeted Cost for Work Scheduled (BCWS), Cost Variance (CV), Schedule Variance (SV), Cost Performance Index (CPI), Schedule Performance Index (SPI), Interpretation of Earned Value Indicators, Error Tracking, Software Reviews, Types of Review: Inspections, Deskchecks, Walk through, Code Reviews, Pair Programming.

Unit IV: Software Quality Assurance and Testing Hours:9

Testing Objectives, Testing Principles, Test Plans, Test Cases, Types of Testing, Levels of Testing, Test Strategies, Program Correctness, Program Verification & validation, Testing Automation & Testing Tools, Concept of Software Quality, Software Quality Attributes, Software Quality Metrics and Indicators, The SEI Capability Maturity Model CMM), SQA Activities,

Unit V: Project Management Hours: 9

Software Configuration Management: Software Configuration Items and tasks, Baselines, Plan for Change, Change Control, Change Requests Management, Version Control, Risk Management: Risks and risk types, Risk Breakdown Structure (RBS), Risk Management Process: Risk identification, Risk analysis, Risk planning, Risk monitoring, Cost Benefit Analysis,

COURSE OUTCOMES

1. Describe and determine the purpose and importance of project management
   from the perspectives of planning, tracking and completion of project.
2. Compare and differentiate organization structures and project structures.
3. Implement a project to manage project schedule, expenses and resources.
4. The application of proper testing tools.
5. Knowledge about Risk Management Process.

Text Books

1. Clifford F. Gray and Erik W. Larson, "Project Management: The Managerial Process with MS" –Mc Graw Hill.

Supplementary Readings

1. M. Cotterell, Software Project Management –Tata McGraw-Hill Publication.
2. Royce, Software Project Management –Pearson Education
3. Kieron Conway, Software Project Management –Dreamtech Press
4. S. A. Kelkar, Software Project Management –PHI Publication.

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	2
CO4	2	3	2	3	2
CO5	2	2	3	3	3

1-LOW 2- MODERATE 3-HIGH