

UNIVERSITY SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
DEPARTMENT OF INFORMATION TECHNOLOGY

PROGRAMME STRUCTURE

B. Tech. IT

4 YEAR PROGRAMME



GAUTAM BUDDHA UNIVERSITY
Gautam Budh Nagar, Greater Noida,
Uttar Pradesh 201308

SEMESTER I

S.No .	Course Code	Course Name	L	T	P	Credits	Types
1	CS101	Fundamentals of Computer Programming	3	1	0	4	CC1 / FC
2	MA101	Engineering Mathematics-I	3	1	0	4	GE1
3	EE102	Basic Electrical Engineering	3	1	0	4	GE2
4	ME101	Engineering Mechanics	3	1	0	4	GE3
5	ES101	Environmental Studies	3	1	0	4	OE1 / AECC
6	EN151	Language Lab	0	0	2	1	OE L1 / AECC
7	CS181	Computer Programming Lab	0	0	2	1	CC-L1 / SEC
8	EE104	Basic Electrical Engineering Lab	0	0	2	1	GE-L1
9	ME102	Workshop Practice	1	0	2	2	GE-L2 / SEC
10	GP	General Proficiency	Non Credit				
Total Hours and Credits			16	5	8	25	

FUNDAMENTALS OF COMPUTER PROGRAMMING			
Course Code:	CS101	Course Credits:	4
Course Category:	CC	Course (U / P)	U
Course Year (U / P):	1U	Course Semester (U / P):	1U
No. of Lectures + Tutorials (Hrs. /Week):	03 + 02	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):	45 + 15	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1 To develop fundamental understanding of computers, its components and programming environment.			
2 To create programming logics and learn C language programming concepts.			
3 To design and develop algorithms and programs with different data declarations, initialization and related operations.			
4 To develop the ability to define and manage functions, array, structures, pointers etc. based on program objective.			
5 To understand and develop C programs to handle computer files, their usage and perform various operations on files.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Understand the Computer fundamentals.			
2 Understand the use of various programming concepts and techniques.			
3 Understand the C programming fundamentals and applications.			
4 Understand C by using arrays, functions, structures and union.			
5 Develop the Programs in C using its advanced features			

UNIT I INTRODUCTION TO COMPUTER AND PROGRAMMING CONCEPTS

Definition, characteristic, generation of computers, basic components of a computer system, memory, input, output and storage units, high level language and low level language, Soft-ware: system software, application software, hardware, firmware, Operating System, compiler, interpreter and assembler, linker, loader, debugger, IDE. Introduction to algorithm and flow chart; representation of algorithm using flow chart symbol, pseudo code, basic algorithm design, characteristics of good algorithm, development of algorithm.

UNIT II INTRODUCTION TO C PROGRAMMING LANGUAGE

Introduction to C programming language, Declaring variables, preprocessor statements, arithmetic operators, programming style, keyboard input, relational operators, introduction, feature of C language, concepts, uses, basic program structure, simple data types, variables, constants, operators, comments, control flow statement :if, while, for, do-while, switch.

UNIT III DATA TYPES AND STRUCTURES

Bitwise operators, Pre-defined and User defined data types, arrays, declaration and operations on arrays, searching and sorting on arrays, types of sorting, 2D arrays, passing 2D arrays to functions, structure, member accessing, structure and union, array of structures, functions, declaration and use of functions, parameter passing, and recursion.

UNIT IV FUNDAMENTALS OF POINTERS

Introduction to pointers, pointer notations in C, Declaration and usages of pointers, operations that can be performed on computers, use of pointers in programming exercises, parameter passing in pointers, call by value, call by references, array and characters using pointers, dynamic memory allocation

UNIT V FILE HANDLING IN C AND ENUM

Introduction to file handling, file operations in C, defining and opening in file, reading a file, closing a file, input output operations on file, counting: characters, tabs , spaces, file opening modes, error handling in input/output operations, sEnumerated data types, use of Enum, declaration of Enum.

Text Books:

[1] C Programming, Herbert Shield

[2] C Programming Language 2nd Edition by Brian, W Kernighan Pearson Education.

Reference Books:

[3] Programming in ANSI C by E. Balagurusamy, Tata McGraw Hill

[4] C Puzzle Book: Puzzles For The C. Programming Language by Alan R Feuer Prentice Hall- Gale

[5] Expert C Programming: Deep C Secrets (s) by Peter Van Der Linden Dorling Kindersley India.

[6] Introduction To UNIX System by Morgan Rachel Tata McGraw Hill Education.

[7] C: A Reference Manual (5th Edition) by Samuel P. Harbison&Samuel P. Harbison.

[8] Programming Using the C Language by Hutchison,R.C, McGraw Hill Book Company, New York.

[9] Fundamentals of computers and programming with C, A.K. SHARMA

COMPUTER PROGRAMMING LAB			
Course Code:	CS181	Course Credits:	1
Course Category:	CCL	Course (U / P)	U
Course Year (U / P):	1U	Course Semester (U / P):	1U
No. of Lectures + Tutorials (Hrs. /Week):	02	Mid Sem. Exam Hours:	--
Total No. of Lectures (L + T):	10	End Sem. Exam Hours:	2
COURSE OBJECTIVES			
1 To develops fundamental understanding C programming environment.			
2 To create programming logics and learn C language programming concepts.			
3 To design and develop algorithms and programs with different data declarations, initialization and related operations.			
4 To develop the ability to define and manage functions, array, structures, pointers etc. based on program objective.			
5 To understand and develop C programs to handle computer files, their usage and perform various operations on files.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Understand the C programming fundamentals.			
2 Understand the use of various programming concepts and techniques.			
3 Understand the C data types and operators with their applications.			
4 Understand C by using arrays, functions, structures and union.			
5 Develop the Programs in C using its advance features.			

LIST OF EXPERIMENTS:

- Write a program to find the sum of the digits of a number.
- Write a program to calculate factorial of a number using recursion.
- Write a program to find the reverse of a given number.
- Write a program to check whether the year is leap or not.
- Write a program to take marks of a student of 5 subjects as an input and print the grade.
marks<40 = FAIL
marks>=40 and <=59 =GOOD
marks>=59 and <80 =EXCELLENT
marks>=80 = OUTSTANDING
- Perform program number 5 using switch case statement.
- Write a program to compute the roots of a quadratic equations.
- Write a program to compute the length of a string using While Loop.
- Write a program to print the following pattern: -
 - ```

*
**


```
  - ```

*
* *
* * *
* * * *

```
 - 0

1 2
3 4 5
6 7 8 9

10. Write a program to compute and display the product of two matrices.
11. Write a program to illustrate the difference between call by value and call by reference.
12. Write a program to check whether a given string is palindrome or not.
13. Create a structure called STUDENT having name, reg no., class as its field.
Compute the size of structure STUDENT.
14. Write a program to compute the length of a string using pointers.
15. Write a program to create a file , input data and display its content.

SEMESTER III									
S.No	Course Code	Course Name	L	T	P	Credits		Types	
1	IT201	Animation and Computer Graphics	3	0	0	3		CC4	
2	IT203	Operating System	3	0	0	3		CC5	
3	IT205	Software Engineering	3	0	0	3		CC6 / SEC	
4	IT207	System Design & Analysis Techniques	3	0	0	3		CC7	
5	IT209	Web Technologies	3	0	0	3		CC8	
6	MA201	Engineering Mathematics-III	3	1	0	4		GE7	
7	IT281	Animation & Computer Graphics Lab	0	0	3	2		CC-L3	
8	IT283	Operating System Lab	0	0	3	2		CC-L4 / SEC	
9	IT285	Web Technologies Lab I	0	0	3	2		CC-L5 / SEC	
10	GP	General Proficiency	Non Credit						
Total Hours and Credits			18	1	9	2	5		

ANIMATION & COMPUTER GRAPHICS THEORY			
Course Code:	IT 201	Course Credits:	3
Course Category:	CC	Course (U / P)	U
Course Year (U / P):	2U	Course Semester (U / P):	3U
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
This course will enable students to:			
1. Understand the need of developing graphics application			
2. Learn algorithmic development of graphics primitives like: line, circle, polygon etc.			
3. Learn the representation and transformation of graphical images and pictures.			
4. To be familiar with image fundamentals and animations			
5. Applying efficient graphics technique to solve engineering problems			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Explain fundamental concepts within computer graphics such as geometrical transformations, illumination models, removal of hidden surfaces and rendering.			
2. Explain the ideas in some fundamental algorithms for computer graphics and to some extent be able to compare and evaluate them.			
3. Be able to Compare various graphics algorithm used in 2D and 3D			
4. Be able to understand and identify the performance characteristics of graphics algorithms and animation.			
5. Employ algorithm to model engineering problems, when appropriate.			

UNIT I INTRODUCTION

Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices, Output primitives : Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.

UNIT II 2-D GEOMETRICAL TRANSFORMS

Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems, 2-D viewing : The viewing pipeline,

viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

UNIT III REPRESENTATION AND TRANSFORMATION

3-D object representation Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces, basic illumination models, polygon rendering methods, 3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

UNIT IV VISIBLE SURFACE DETECTION METHODS

Classification, back -face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods. Tools of Multimedia: Paint and Draw Applications, Graphic effects and techniques, Image File Format, Anti-aliasing, Morphing, Multimedia Authoring tools, professional development tools.

UNIT V COMPUTER ANIMATION

Introduction and Principles of Animations, Power of Motion, Animation Techniques, Animation File Format, Making animation for Rolling Ball, making animation for a Bouncing Ball, Animation for the web, GIF, Plugins and Players, Animation tools for World Wide Web. Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

REFERENCE BOOKS:

1. Computer Graphics: principals and practice Foley, vanDam, Feiner Hughes Addison Wesley
2. Mathematical Elements of Graphics Roges Tata McGrow Hill
3. Computer Graphics Donald Hearn and M.Pauline Baker Prentice Hall India
4. Procedural Elements-Computer Graphics, David Rogers, TMH
5. Principals of Computer graphics, Shalini Govil-pal, springer

Course Code:	IT203	Course Credits:	3
Course Category:	CC	Course (U / P)	U
Course Year (U / P):	2U	Course Semester (U / P):	3 U
No. of Lectures + Tutorials (Hrs/Week):3	03 + 00	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):45	45 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1 Understanding how Operating System is Important for Computer System.			
2 To make aware of different types of Operating System and their services.			
3 To learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system			
4 To know virtual memory concepts and secondary memory management			
5 Understanding of Security & protection in Operating System			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1Understands the different services provided by Operating System at different level			
2They learn real life applications of Operating System in every field.			
3 Understands the use of different process scheduling algorithm and synchronization techniques to avoid deadlock.			
They will learn different memory management techniques like paging, segmentation and demand paging etc.			
5 Perform implementation of protection mechanisms in operating system			

UNIT I INTRODUCTION TO OPERATING SYSTEM

Importance of operating systems, basic concepts and terminology about operating system, memory management, processor management, device management, information management functions.

UNIT II PROCESS MANAGEMENT

Elementary concept of process, job scheduler, process scheduling, operation on process, threads, overview, scheduling criteria, scheduling algorithms, algorithm evaluation process synchronization, synchronization hardware, semaphores, classical problem of synchronization, monitors and atomic transaction deadlocks: system model, deadlock characterization, deadlocks prevention, deadlocks avoidance, deadlocks detection, recovery from deadlock.

UNIT III MEMORY & STORAGE MANAGEMENT

Basic Memory Management: Definition, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, partition, Fragmentation, Compaction, Paging, Segmentation.

Virtual Memory: Basics of virtual memory, Hardware and control structures-Locality of reference, Page fault, Demand paging, page replacement policies: First In First Out (FIFO), second chance (SC), Not recently used (NRU) and Least recently used (LRU).

UNIT IV UNIX/LINUX OPERATING SYSTEM: Development Of Unix/Linux, Role & Function Of Kernel, System Calls, Elementary Linux command & Shell Programming, Directory Structure, System Administration, Case study: Linux, Windows Operating System

UNIT V SECURITY & PROTECTION: Security Environment, Design Principles of Security, User authentication, Protection Mechanism: Protection Domain, Access Control List

Text Books:

- [1]. Galvin, Wiley, Operating Systems Concepts, 8th edition, 2009.
- [2]. James L Peterson, Operating Systems Concept, John Wiley & Sons Inc, the 6Rev edition, 2007.

Reference Books:

- [3]. Deitel H. M., An Introduction to Operating Systems, Addison-Wesley, 1990.
- [4]. Stallings William, Operating Systems, PHI, New Delhi, 1997.
- [5]. S. Tanenbaum Modern Operating Systems, Pearson Education, 3rd edition, 2007.
- [6]. Nutt, Operating System, Pearson Education, 2009.
- [7]. S. Tanenbaum, Distributed Operating Systems, Prentice Hall, 2nd edition, 2007.

SOFTWARE ENGINEERING			
Course Code:	IT205	Course Credits:	3
Course Category:	CC	Course (U / P)	U
Course Year (U / P):	2U	Course Semester (U / P):	3U

No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1 Knowledge of basic SW engineering methods and practices and application.			
2 A general understanding of software process models.			
3 Understanding of software requirements and the SRS documents.			
4 Understanding of software design process.			
5 Understanding of software coding, testing and maintenance.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Basic knowledge and understanding of the analysis and design of complex systems.			
2 Ability to apply software engineering principles and techniques.			
3 Ability to design, develop, maintain and evaluate large-scale software systems.			
4 To produce efficient, reliable, robust and cost-effective software solutions.			
5 Ability to perform independent research and analysis.			

UNIT I SOFTWARE ENGINEERING

Introduction to software engineering: definitions, role of software engineering, planning a software project, defining the problem, developing a solution strategy, planning the development process, software engineering process paradigms, principles of software engineering, software engineering activities.

UNIT II REQUIREMENT ANALYSIS AND DESIGN

Software Requirement Specification (SRS): Introduction, need of SRS, significance, characteristics of SRS, Structure of SRS, IEEE standards for SRS design, functional and non-functional requirements, Requirement gathering and analysis, requirement engineering and management.

UNIT III SOFTWARE DESIGN PROCESS

Software Design: Introduction, design process activities: architectural design, Abstract specification, Interface design, component design, data structure design, algorithm design modular approach, top-down design,

bottomup design, design methods: data-flow model: data flow diagram, entity-relation-attribute model: E-R diagram, structural model: structure charts, context diagrams, object models: use case modeling, use case diagrams, sequence diagrams, cohesion and coupling.

UNIT IV SOFTWARE LIFE CYCLE MODELS

Software Development Life Cycle (SDLC), SDLC models, waterfall model and its variations, prototype model, iterative enhancement model, spiral model, RAD model, comparison of these models, software development teams, software development environments, validation and traceability, maintenance, prototyping requirements, Software project management.

UNIT V SOFTWARE CODING, TESTING AND MAINTENANCE

Coding, Testing Methods: unit testing, integration testing, system testing, acceptance testing, testing techniques: white box testing, black box testing, thread testing, regression testing, alpha testing, beta testing, static testing, dynamic testing, Evolution of software products, economics of maintenance, category of software maintenance, Role of product development life cycle, deployment model, adaptive maintenance, corrective maintenance, perfective maintenance, enhancement request, proactive defect prevention, problem reporting, problem resolution, software maintenance from customers' perspective, maintenance standard: IEEE-1219, ISO-12207.

REFERENCE BOOKS:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House, New Delhi 1997.
2. Ian Sommerville, Software Engineering, Pearson Education, 2009.
3. Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
4. Software Engineering: Software Reliability, Testing and Quality Assurance, Nasib S. Gill, Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.

SYSTEM DESIGN AND ANALYSIS TECHNIQUES			
Course Code:	IT207	Course Credits:	3
Course Category:	CC	Course (U / P)	U
Course Year (U / P):	2U	Course Semester (U / P):	3 U
No. of Lectures + Tutorials (Hrs/Week):3	03 + 00	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):45	45 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1 Explain what systems are and how they are developed.			
2 Identify and describe the phases of the systems development life cycle.			
3 Describe the role and responsibilities of the systems analyst in the development and management of systems.			
4 Use tools and techniques for process and data modeling.			
5 Develop and deliver a Requirements Definition Proposal for a new system in a well-structured business proposal			

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Understand the basis for understanding the life cycle of a systems development project;
2 Experience in developing information systems models
3 Experience in developing systems project documentation
4 Able to analyze business problems and develop a requirements° document, written in clear and concise business language.
5 An understanding of the object-oriented methods models as covered by the Unified Modelling Language.

UNIT-1

DATA AND INFORMATION

Types of information: operational, tactical, strategic and statutory, why do we need information systems, management structure, requirements of information at different levels of management, functional allocation of management, requirements of information for various functions, qualities of information, small case study.

UNIT-2

SYSTEMS ANALYSIS AND DESIGN LIFE CYCLE

Requirements determination, requirements specifications, feasibility analysis, final specifications, hardware and software study, system design, system implementation, system evaluation, system modification, role of systems analyst, attributes of a systems analyst, tools used in system analysis.

UNIT-3

INFORMATION GATHERING

Strategies, methods, case study, documenting study, system requirements specification. from narratives of requirements to classification of requirements as strategic, tactical, operational and statutory.

UNIT- 4

FEASIBILITY ANALYSIS

Deciding project goals. examining alternative solutions, cost benefit analysis, quantifications of costs and benefits, payback period, system proposal preparation for managements, parts and documentation of a proposal, tools for prototype creation.

UNIT-5

TOOLS FOR SYSTEMS ANALYSTS

Data flow diagrams, case study for use of DFD, good conventions, leveling of DFDs, leveling rules, logical and physical DFDs, software tools to create DFDs, decision tables for complex logical specifications, specification oriented design vs procedure oriented design

Text Books:

1. Elias M.Awad., System Analysis and Design.

WEB TECHNOLOGIES			
Course Code:	IT209	Course Credits:	3
Course Category:	CC	Course (U / P)	U
Course Year (U / P):	2U	Course Semester (U / P):	3U
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1 Explain the history of the internet and related internet concepts that are vital in understanding web development.			
2 Discuss the insights of internet programming and implement complete application over the web			
3 Demonstrate the important HTML tags for designing static pages and separate design from content using Cascading Style sheet.			
4 Describe the concepts of client side scripting like JavaScript.			
5 Understanding the concept of web hosting and seo			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Students are able to develop a dynamic webpage by the use of java script and HTML.			
2 Understand the concept of JAVA SCRIPTS.			
3 Able to build interactive web applications using CSS			
4 Understand the basic concepts of websites.			
5 To develop and deploy real time web applications in web servers			

UNIT-1

Introduction: Basic principles involved in developing a web site, Planning process , Domains and Hosting, Responsive Web Designing , Types of Websites (Static and Dynamic Websites), Web Standards and W3C recommendations.

UNIT-2

Introduction to HTML: What is HTML, HTML Documents, Basic structure of an HTML document, creating an HTML document, Mark up Tags, Heading-Paragraphs, Line Breaks.

Elements of HTML: HTML Tags., Working with Text, Working with Lists, Tables and Frames, Working with Hyperlinks, Images and Multimedia, Working with Forms and controls.

UNIT-3

Concept of CSS: Creating Style Sheet, CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), Working with block elements and objects, Working with Lists and Tables, CSS Id and Class, Box Model (Introduction, Border properties, Padding Properties, Margin properties).

UNIT-4

Introduction to Client Side Scripting: Introduction to Java Script , Javascript Types , Variables in JS, Operators in JS , Conditions Statements , Java Script Loops, JS Popup Boxes , JS Events , JS Arrays, Working with Arrays, JS Objects, JS Functions, Using Java Script in Real time, Validation of Forms, Related Examples.

UNIT-5

Web Hosting: Web Hosting Basics, Types of Hosting Packages, Registering domains, Defining Name Servers, Using Control Panel, Creating Emails in Cpanel , Using FTP Client, Maintaining a Website.

Concepts of SEO: Basics of SEO, Importance of SEO, on-page Optimization Basics and off-page SEO.

Text Books:

1. Web Technologies, Uttam Roy, OXFORD University press
2. Web programming with HTML, XHTML and CSS, 2e, Jon Duckett, Wiley India
3. Steven M. Schafer, "HTML, XHTML, and CSS Bible, 5ed", Wiley India
4. Ian Pouncey, Richard York, "Beginning CSS: Cascading Style Sheets for Web Design", Wiley India
5. SEO for Dummies, 6th Edition, by Peter Kent
6. Teach Yourself Javascript in 24 Hours, 5th Edition, by Ballard and Moncur

REFERENCE BOOKS:

1. Web programming Bai, Michael Ekedahl, CENAGE Learning , India edition.
2. An Introduction to Web Design + Programming, Paul S.Wang, India Edition

ANIMATION & COMPUTER GRAPHICS LAB			
Course Code:	IT 281	Course Credits:	2
Course Category:	CCL	Course (U / P)	U
Course Year (U / P):	2U	Course Semester (U / P):	3U
No. of Lectures + Tutorials (Hrs/Week):3	03 + 00	Mid Sem. Exam Hours:	
Total No. of Lectures(L + T): 10	10 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
This course will enable students to:			
1. Learn algorithmic development of graphics primitives like: line, circle, polygon etc.			
2. Learn the representation and transformation of graphical images and pictures.			
3. Learn algorithmic development of graphics primitives like: line, circle, polygon etc			
4. Learn the representation and transformation of graphical images and pictures.			
5. To familiarize the students with various approaches, methods and techniques of Animation Technology.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Explain fundamental concepts within computer graphics such as geometrical transformations, illumination models, removal of hidden surfaces and rendering			
2. Explain the ideas in some fundamental algorithms for computer graphics and to some extent be able to compare and evaluate them			
3. Be able to Compare various graphics algorithm used in 2D and 3D.			
4. To be familiar with image fundamentals and animations.			
5. Employ algorithm to model engineering problems, when appropriate.			

List of Experiments:

1. Procedure to create an animation to represent the growing moon.
2. Procedure to create an animation to indicate a ball bouncing on steps.
3. Procedure to simulate movement of a cloud.
4. Procedure to draw the fan blades and to give proper animation.
5. Procedure to display the background given (filename: tulip.jpg) through your name.
6. Procedure to display the background given (filename: garden.jpg) through your name using mask.
7. Procedure to create an animation with the following features.
WELCOME (Letters should appear one by one .The fill color of the text should change to a different colour after the display of the full word.)

8. Procedure to simulate a ball hitting another ball.
9. Procedure to design a visiting card containing at least one graphic and text information.
10. Procedure to take a photographic image. Give a title for the image. Put the border. Write your names. Write the name of institution and place.
11. Procedure to prepare a cover page for the book in your subject area. Plan your own design.
12. Procedure to extract the flower only from given photographic image and organize it on a Background, Selecting your own background for organization.

Course Code:	IT283	Course Credits:	2
Course Category:	CCL	Course (U / P)	U
Course Year (U / P):U	2U	Course Semester (U / P):	3 U
No. of Lectures + Tutorials (Hrs/Week):3	03 + 00	Mid Sem. Exam Hours:	
Total No. of Lectures(L + T):10	10 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1To learn about file management and different types of permission setup			
2To understand how system processes work and how to manage them			
3 To learn & implement different Operating system algorithm			
4 Apply concepts of Deadlock and its prevention.			
5 Apply concept of OS to develop Producer Consumer problem & real scenario problems			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Able to implement and analyse the performance of different algorithm of Operating Systems like CPU scheduling algorithm, page replacement algorithms, deadlock avoidance, detection algorithm and so on.			
2 Able to design and develop a course project that can have positive impact on environment or society or mankind.			
3 Demonstrate the various operations of file system.			
4 Apply the various methods in memory allocation and page replacement algorithm.			
5 Apply the process synchronous concept using message queue, shared memory, semaphore and Dekker's algorithm for the given situation			

1. Program for file handling.
2. Program for Dining Philosophers Problem.
3. Program for Producer – Consumer Problem concept.
4. Program for First Come First Serve Algorithm.
5. Program for Shortest Job First Scheduling Algorithm.
6. Program for Round Robin Scheduling Method.
7. Program for Priority Scheduling Algorithm.
8. Implement the concept of Fragmentation and Defragmentation.

9. Simulate Bankers Algorithm for Dead Lock Avoidance

10. Simulate Bankers Algorithm for Dead Lock Prevention

WEB TECHNOLOGIES LAB			
Course Code:	IT285	Course Credits:	2
Course Category:	CCL	Course (U / P)	U
Course Year (U / P):	2U	Course Semester (U / P):	3U
No. of Labs + Tutorials (Hrs/Week):	03+00	Mid Sem. Exam Hours:	
Total No. of Labs (L) :	10	End Sem. Exam Hours:	3

COURSE OBJECTIVES
1 Students will learn to analyze a web page and identify its elements and attributes
2 Create web pages using XHTML and Cascading Style Sheets
3 Students will learn the ordered list and unordered list
4 Student will learn the inline CSS, internal CSS and external CSS
5 Students will learn that how to create the login form using HTML.
COURSE OUTCOMES
At the end of the course the students should be able to:
1 Learn the how to display attributes and their values using HTML
2 Learn the how to embed images in a web pages
3 Learn to create the Registration form, login form.
4 Learn the Inline CSS, Internal CSS, External CSS
5 Learn the how to create the websites.

NOTE: Suggested list of experiments but not limited to these only.

List of Experiments: In this lab programs related to HTML etc. are suggested.

1. Write an HTML code to display your education details.
2. Write an HTML code to display attributes and their values for HTML elements.
3. Write an HTML code to create a web page having website links.
4. Write HTML code to embed images in a web page.
5. Write an HTML code to create a Registration Form. On submitting the form, the user should be asked to login with this new credentials.
6. Write an HTML code to create a login form. On submitting the form, the user should get navigated to a profile page.
7. Write an HTML code to illustrate the usage of the following: • Ordered List • Unordered List
8. Write code to create a frameset having header, navigation and content sections.
9. Write code to create your Institute website, Department Website.
10. Write code to demonstrate the usage of inline CSS.
11. Write code to demonstrate the usage of internal CSS.
12. Write code to demonstrate the usage of external CSS.
13. Make a form for keeping student record and validate it.
14. Write program to design an entry form of student details.

SEMESTER V							
S.No	Course Code	Course Name	L	T	P	Credits	Types
1	IT301	Theory of Automata	3	0	0	3	CC11
2	IT303	Computer Networks	3	0	0	3	CC12
3	IT305	Compiler Design	3	1	0	4	CC13
4	IT307	Soft Computing Techniques	3	0	0	3	CC14 / SEC
5	IT 315	Elective 1	3	0	0	3	E1 / DSE
6	IT 319	Elective 2	3	0	0	3	E2 / DSE
7	IT381	Computer Networks Lab	0	0	3	2	CC-L9
8	IT383	Compiler Design Lab	0	0	3	2	CC-L10
9	IT385	Soft Computing Techniques Lab	0	0	3	2	CC-L11 / SEC

10	GP	General Proficiency	Non Credit			
Total Hours and Credits			18	1	9	25

Theory of Automata			
Course Code	IT301	Course Credit	03
Course Category	CC	Course(U/P)	U
No of Lectures + Tutorials(Hrs./Week)	03+00	Mid Semester Exam Hours:	01
Course Year (U / P):	3U	Course Semester	5U
Total no of Lectures(L+T)	45+00	End Term Exam Hours:	03
COURSE OBJECTIVES			
1. The objective of this course is to introduce students to the foundation of computability theory.			
2. Application of mathematical techniques and logical reasoning to important problem.			
3. Develop a strong background in reasoning about finite state automata and formal language.			
4. This course is to explore the theoretical foundations of computer science from the perspective of formal language and classify machines by their power to recognize languages.			
5. the basic theory of computer science and formal methods of computation like automation theory, formal language, grammars, Turing machine			
Course Outcomes			
At the end of the course the student should be able to understand the :			
1. Under the basic property of regular grammar and design automata			

2. Language accepted by an automata i.e. DFA(Deterministic Finite Automata)/NFA(Non deterministic finite automata).
3. Understand the regular expression(RE) ,Kleen closure ,positive closure, RE to FA and FA to RE
4. Closure property of different language and Decidability /Undesirability property of different languages.
5. Define the various categories of language grammars in the Chomsky hierarchy and variants of Turing machine

UNIT I AUTOMATA

Introduction; alphabets, strings and languages; automata and grammars, deterministic finite automata (DFA)-formal definition, simplified notation: state transition graph, transition table, language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, language of NFA, equivalence of NFA and DFA, minimization of finite automata, distinguishing one string from other, Myhill-Nerode Theorem

UNIT II REGULAR EXPRESSIONS AND LANGUAGES

Regular expression (RE), definition, operators of regular expression and their precedence, algebraic laws for regular expressions, Kleen's theorem, regular expression to FA, DFA to regular expression, arden theorem, non-regular languages, pumping lemma for regular languages. Application of pumping lemma, closure properties of regular languages, decision properties of regular languages, FA with output: moore and mealy machine, equivalence of moore and mealy machine, applications and limitation of FA.

UNIT III CONTEXT-FREE GRAMMAR AND LANGUAGES

Context Free Grammar (CFG) and Context Free Languages (CFL): definition, examples, derivation, derivation trees, ambiguity in grammar, inherent ambiguity, ambiguous to unambiguous CFG, useless symbols, simplification of CFGs, normal forms for CFGs: CNF and GNF, closure properties of CFLs, decision properties of CFLs: emptiness, finiteness and membership, pumping lemma for CFLs.

UNIT IV PUSH DOWN AUTOMATA

Push Down Automata (PDA): description and definition, instantaneous description, language of PDA, acceptance by final state, acceptance by empty stack, deterministic PDA, equivalence of PDA and CFG, CFG to PDA and PDA to CFG, two stack PDA

UNIT V TURING MACHINES (TM)

Basic model, definition and representation, instantaneous description, language acceptance by TM, variants of turing machine, TM as computer of integer functions, universal TM, church's thesis recursive and recursively enumerable languages, halting problem, introduction to undecidability, undecidable problems about TMs. Post Correspondence Problem (PCP), modified PCP, introduction to recursive function theory.

Text Books:

1. Hopcroft, Ullman, "Introduction to Automata Theory, Languages and Computation", Pearson Education
2. K.L.P. Mishra and N.Chandrasekaran, "Theory of Computer Science : Automata, Languages and Computation", PHI

References Books:

3. Martin J. C., "Introduction to Languages and Theory of Computations", TMH
4. Papadimitrou, C. and Lewis, C.L., "Elements of the Theory of Computation", PHI

COMPUTER NETWORKS			
Course Code:	IT 303	Course Credits:	3
Course Category:	CC	Course (U / P)	U
Course Year (U / P):	3U	Course Semester	5P
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1. Understanding of computer networks and its components			
2. A general understanding of switching and OSI layers			
3. Understanding of concept of congestion in the network			
4. Understanding of protocols used in computer networks			
5. Understanding of addressing in the computer network			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Understand network scenario			
2. Understand OSI and TCP/IP layers			
3. Understand the concept of congestion in the network			
4. Understand various protocols used in each layer			
5. Able to create network with the use of IP address			

UNIT I INTRODUCTION AND PHYSICAL LAYER

Key concepts of computer network, transmission media, network devices, network topology, topology design issues, types of networks: LAN, MAN, WAN, PAN, ISDN systems and ATM network, OSI-reference model, open system standards, characteristics of network, TCP/IP model, protocols and standards, encoding technique.

UNIT II SWITCHING AND DATA LINK LAYER

Circuit switching, packet switching, message switching, hybrid switching, and ATM switching, multiplexing techniques: TDMA, FDMA, WDMA, CDMA, data link layer: LLC & MAC level protocols and design issues,

issues IEEE 802 LAN Standards, framing, CRC, error control, flow control, HDLC, ALOHA and performance issues. Frames relay networks and performance parameters.

UNIT III NETWORK LAYER

Network layer design issues, overview of IPv4 and IPv6, addressing: class full and classless, static and dynamic, subnet and super net, auto configuration through DHCP, routing protocols: RIP, DVR,LSR, OSFP, BGP, congestion control algorithm, subnet concept, virtual LAN, ICMP, multicasting, mobile IP.

UNIT IV TRANSPORT LAYER

Port addressing schemes, connectionless and connection oriented services: TCP and UDP, wireless TCP, Congestion control, queue management, NAT, PAT, socket format at transport level, socket interface and programming.

UNIT V APPLICATION LAYER

Client server architecture, domain name services, application services: HTTP, TELNET, RLOGIN, FTP, CBR, NFS, SMTP, POP, IMAP, MIME, voice and video over IP, social issues- privacy, freedom of speech, copy right.

Text Books:

1. S. Tanenbaum, Computer Networks, 4th edition, Prentice Hall, 2008
2. Forouzan, B.A., Data Communication and Networking, Tata McGraw-Hill.

References Books:

1. W. Stallings, Data and Computer Communications, 8th edition, Prentice Hall, 2007
2. Douglas E. Comer TCP/IP Principles, Protocols and Architecture, Pearson Education
3. F. Haball ,Data Communication, Computer network & open systems - Computer Networks : An Engineering approach - S. Keshav
4. Kurose, J.F. & Ross, K.W., Computer Networking: A Top-Down Approach Featuring the Internet, Addison Wesley.

COMPILER DESIGN			
Course Code:	IT305	Course Credits:	4
Course Category:	CC	Course (U / P)	U
Course Year (U / P):	3U	Course Semester (U / P):	5U
No. of Lectures + Tutorials (Hrs/Week):	03 + 01	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):	45 + 15	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1. Understand the basic principles of compiler design, its various constituent parts, algorithms and data structures required to be used in the compiler.			
2. Understand relations between computer architecture and how its understanding is useful in design of a compiler.			
3. How to construct efficient algorithms for compilers.			
4. Provide an understanding of the fundamental principles in compiler design.			
5. Learn the process of translating a modern high-level-language to executable code required for compiler construction.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Acquire knowledge of different phases and passes of the compiler and also able to use the compiler tools like LEX, YACC, etc. Students will also be able to design different types of compiler tools to meet the requirements of the realistic constraints of compilers.			
2. Understand the parser and its types i.e. Top-Down and Bottom-up parsers and construction of LL, SLR, CLR, and LALR parsing table.			
3. Implement the compiler using syntax-directed translation method and get knowledge about the synthesized and inherited attributes.			
4. Acquire knowledge about run time data structure like symbol table organization and different techniques used in that.			
5. Understand the target machine's run time environment, its instruction set for code generation and techniques used for code optimization.			

UNIT I

Introduction to Compiler: Phases and passes, Bootstrapping, Finite state machines and regular expressions and their applications to lexical analysis, Optimization of DFA-Based Pattern Matchers implementation of lexical analyzers, lexical-analyzer generator, LEX compiler, Formal grammars and their application to syntax analysis, BNF notation, ambiguity, YACC. The syntactic specification of programming languages: Context free

grammars, derivation and parse trees, Capabilities of CFG.

UNIT II

Basic Parsing Techniques: Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.

UNIT III

Syntax-directed Translation: Syntax-directed Translation schemes, Implementation of Syntax- directed Translators, Intermediate code, postfix notation, Parse trees & syntax trees, three address code, quadruple & triples, translation of assignment statements, Boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser. More about translation: Array references in arithmetic expressions, procedures call, declarations and case statements.

UNIT IV

Symbol Tables: Data structure for symbols tables, representing scope information. Run-Time Administration: Implementation of simple stack allocation scheme, storage allocation in block structured language. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors semantic errors.

UNIT V

Code Generation: Design Issues, the Target Language. Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Code Generator. Code optimization: Machine-Independent Optimizations, Loop optimization, DAG representation of basic blocks, value numbers and algebraic laws, Global Data-Flow analysis.

Text books:

1. K. Muneeswaran, Compiler Design, First Edition, Oxford University Press.
2. J.P. Bennet, "Introduction to Compiler Techniques", Second Edition, Tata McGraw-Hill, 2003.
3. Henk Alblas and Albert Nymeyer, "Practice and Principles of Compiler Building with C", PHI, 2001.
4. Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools", Pearson Education
5. V Raghvan, " Principles of Compiler Design", TMH

SOFTWARE ENGINEERING			
Soft Computing Techniques			
Course Code:	IT 3	Course Credits:	3

	0 7		
Course Category:	C C	Course (U / P)	U
Course Year (U / P):	3 U	Course Semest er (U / P):	5U
No. of Lectures + Tutorials (Hrs/Week):	0 3 + 0 0	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):	4 5 + 0 0	End Sem. Exam Hours:	3

COURSE OBJECTIVES

- 1 Develop the skills to gain a basic understanding of neural network theory and fuzzy logic theory.
- 2 Introduce students to artificial neural networks and fuzzy theory from an engineering perspective.
- 3 Understanding of various learning paradigms.
- 4 Develop the skills to gain understanding applications of soft computing.
- 5 Understanding the concept of genetic algorithm.

COURSE OUTCOMES

At the end of the course the students should be able to:

- 1 Comprehend the fuzzy logic and the concept of fuzziness involved in various systems and fuzzy set theory.
- 2 Understand the concepts of fuzzy sets, knowledge representation using fuzzy rules and fuzzy inference system.
- 3 To understand the fundamental theory and concepts of neural networks and neural network architectures.
- 4 Understand appropriate learning rules for each of the architectures and working of genetic algorithm.
- 5 Reveal different applications of these models to solve engineering and other problems.

UNIT I FUZY LOGIC

Introduction to fuzzy logic, classical and fuzzy sets, overview of fuzzy sets, membership function, fuzzy rule generation, operations on fuzzy sets: compliment, intersection, union, combinations on operations, aggregation operation.

UNIT II FUZZY ARITHMETIC

Fuzzy numbers, linguistic variables, arithmetic operations on intervals & numbers, uncertainty based information, information and uncertainty, no specificity of fuzzy and crisp sets, fuzziness of fuzzy sets.

UNIT III NEURAL NETWORK

Overview of biological neurons, computational neuron, mathematical model of neurons, ANN architecture, single layer and multilayer architectures, activation function, threshold value, self learning and forced learning algorithms, feed forward and feedback architectures.

UNIT IV LEARNING FUNDAMENTALS

Learning paradigms, supervised and unsupervised learning, reinforced learning, ANN training, algorithms perceptions, training rules, delta, back propagation algorithm, multilayer perception model, Hopfield networks, associative memories, applications of artificial neural networks,

UNIT V GENETIC ALGORITHMS

History of genetic algorithm, terminology of genetic algorithm, biological background, creation of offspring, working principles of genetic algorithms, fitness function, reproduction: Roulette wheel selection, Boltzmann selection, cross over mutation, inversion, deletion, and duplication, generation cycle.

Reference Books:

1. Artificial Neural Networks: An introduction to ANN Theory and Practice, Peteus J. Braspenning, PHI publication, 2005.
2. Fuzzy Logic: A spectrum of Theoretical and Practical issues, Paul P. Wang, pearson publication 2004.
3. Fuzzy Sets, Fuzzy logic, and Fuzzy Systems: Selected Papers- Lotfi Asker Zadeh, George J. Kilm, Bo yuan, 2005.
4. Foundations of Fuzzy logic and Soft Computing: 12th International Fuzzy conference proceeding, 2005.
5. Neural Networks Theory, Patricia Melin, Oxford University press, 2003.

Course Code:	IT315	Course Credits:	3
Course Category:	E1	Course (U / P)	U
Course Year (U / P):	3U	Course Semester (U / P):	5U
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1
Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1 Present the basic of Computer Based Numerical & Statistical Techniques			
2 Helps in learning step by step approach.			
3 Understanding of Methods.			
4 Understanding of learning the process and application			
5 Understanding of Computer Based Numerical & Statistical Techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Will have basic knowledge and understanding of Computer Based Numerical & Statistical Techniques.			
2 Apply methods.			
3 learn various techniques.			
4 Practical Approach			

UNIT-I

Floating point Arithmetic: Representation of floating point numbers, Operations, Normalization, Pitfalls of floating point representation, Errors in numerical computation Iterative Methods: Zeros of a single transcendental equation and zeros of polynomial using Bisection Method, Iteration Method, Regula-Falsi method, Newton Raphson method, Secant method, Rate of convergence of iterative methods.

UNIT-II

Simultaneous Linear Equations: Solutions of system of Linear equations, Gauss Elimination direct method and pivoting, Ill Conditioned system of equations, Refinement of solution. Gauss Seidal iterative method, Rate of Convergence Interpolation and approximation: Finite Differences, Difference tables Polynomial Interpolation: Newton's forward and backward formula Central Difference Formulae: Gauss forward and backward formula, Stirling's, Bessel's, Everett's formula. Interpolation with unequal intervals: Langrange's Interpolation, Newton Divided difference formula, Hermite's Interpolation Approximation of function by Taylor's series and Chebyshev polynomial.

UNIT-III

Numerical Differentiation and Integration: Introduction, Numerical Differentiation, Numerical Integration, Trapezoidal rule, Simpson's rules, Boole's Rule, Weddle's Rule Euler- Maclaurin Formula Solution of differential

equations: Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta methods, Predictor-corrector method, Automatic error monitoring, stability of solution.

UNIT-IV

Curve fitting, Cubic Spline and Approximation: Method of least squares, fitting of straight lines, polynomials, exponential curves etc. Frequency Chart: Different frequency chart like Histogram, Frequency curve, Pi-chart. Regression analysis: Linear and Non-linear regression, Multiple regression.

UNIT-V

Time series and forecasting: Moving averages, smoothening of curves, forecasting models and methods. Statistical Quality Controls methods Testing of Hypothesis: Test of significance, Chi-square test, t-test, ANOVA, F-Test Application to medicine, agriculture etc.

References Books:

- [1] Rajaraman V., "Computer Oriented Numerical Methods", PHI
- [2] Gerald & Wheatley, "Applied Numerical Analyses", AW
- [3] Jain, Iyengar and Jain, "Numerical Methods for Scientific and Engineering Computations", New Age Int.
- [4] Grewal B. S., "Numerical methods in Engineering and Science", Khanna Publishers, Delhi
- [5] T. Veerarajan, T Ramachandran, "Theory and Problems in Numerical Methods", TMH
- [6] Pradip Niyogi, "Numerical Analysis and Algorithms", TMH
- [7] Francis Scheld, "Numerical Analysis", TMH
- [8] Gupta S. P., "Statistical Methods", Sultan and Sons

SOFTWARE PROJECT MANAGEMENT			
Course Code:	IT319	Course Credits:	3
Course Category:	E2	Course (U / P)	U
Course Year (U / P):	3U	Course Semester (U / P):	5U
No. of Lectures + Tutorials (Hrs/Week):	03 + 00	Mid Sem. Exam Hours:	1

Total No. of Lectures (L + T):	45 + 00	End Sem. Exam Hours:	3
COURSE OBJECTIVES			
1 Present the basic software project management approach			
2 Helps in learning step by step approach.			
3 Understanding of Methods.			
4 Understanding of learning the process and application			
5 Understanding of paradigms of software project management techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Will have basic knowledge and understanding of software project management.			
2 Apply methods.			
3 learn various techniques.			
4 Practical Approach			

UNIT-I: INTRODUCTION AND SOFTWARE PROJECT PLANNING

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

Project Elements, Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Project Lifecycle 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

Dimensions of Project Monitoring & Control, Earned Value Analysis, Earned Value Indicators: 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, Desk checks, Walkthroughs, Code Reviews, Pair Programming.

UNIT-IV: SOFTWARE QUALITY ASSURANCE AND TESTING

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, Formal SQA Approaches: Proof of correctness, Statistical quality assurance, Cleanroom process.

UNIT-V: PROJECT MANAGEMENT AND PROJECT MANAGEMENT TOOLS

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, Software Project Management Tools: CASE Tools, Planning and Scheduling Tools, MS-Project.

Textbooks:

[1]. M. Cotterell, Software Project Management, Tata McGraw-Hill Publication.

[2]. Royce, Software Project Management, Pearson Education

Reference Books:

[3]. Kieron Conway, Software Project Management, Dreamtech Press

[4]. S. A. Kelkar, Software Project Management, PHI Publication.

[5]. Harold R. Kerzner, Project Mangment "A Systems Approach to Planning, Scheduling, and Controlling" Wiley.