

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Computer Science and Design, IV Semester

BT401 Mathematics III

OBJECTIVES: The objective of this course is to fulfill the needs of engineers to understand applications of Numerical Analysis, Transform Calculus and Statistical techniques in order to acquire mathematical knowledge and to solving wide range of practical problems appearing in different sections of science and engineering. More precisely, the objectives are:

- To introduce effective mathematical tools for the Numerical Solutions algebraic and transcendental equations.
- To enable young technocrats to acquire mathematical knowledge to understand Laplace transformation, Inverse Laplace transformation and Fourier Transform which are used in various branches of engineering.
- To acquaint the student with mathematical tools available in Statistics needed in various field of science and engineering.

Course Outcome's: After completion of this course, the students are able to:

CO1: Use effective mathematical tools for the Numerical Solutions algebraic and transcendental equations.

CO2: Acquire mathematical knowledge to understand Laplace transformation, Inverse Laplace transformation and Fourier Transform and use them in various branches of engineering.

CO3: Implement mathematical tools available in Statistics needed in various field of science and engineering.

Unit 1 : Numerical Methods – 1: Solution of polynomial and transcendental equations – Bisection method, Newton-Raphson method and Regula-Falsi method. Finite differences, Relation between operators, Interpolation using Newton's forward and backward difference formulae. Interpolation with unequal intervals: Newton's divided difference and Lagrange's formulae.

Unit II : Numerical Methods – 2: Numerical Differentiation, Numerical integration: Trapezoidal rule and Simpson's 1/3rd and 3/8 rules. Solution of Simultaneous Linear Algebraic Equations by Gauss's Elimination, Gauss's Jordan, Crout's methods, Jacobi's, Gauss-Seidal, and Relaxation method.

Unit III : Numerical Methods – 3: Ordinary differential equations: Taylor's series, Euler and modified Euler's methods. RungeKutta method of fourth order for solving first and second order equations. Milne's and Adam's predictor-corrector methods. Partial differential equations: Finite difference solution two dimensional Laplace equation and Poission equation, Implicit and explicit methods for one dimensional heat equation (BenderSchmidt and Crank-Nicholson methods), Finite difference explicit method for wave equation.

Unit IV : Transform Calculus: Laplace Transform, Properties of Laplace Transform, Laplace transform of periodic functions. Finding inverse Laplace transform by different methods, convolution theorem. Evaluation of integrals by Laplace transform, solving ODEs by Laplace Transform method, Fourier transforms.

Unit V : Concept of Probability: Probability Mass function, Probability Density Function, Discrete Distribution: Binomial, Poisson's, Continuous Distribution: Normal Distribution, Exponential Distribution.

TEXT/ REFERENCE BOOKS:

1. P. Kandasamy, K. Thilagavathy, K. Gunavathi, Numerical Methods, S. Chand & Company, 2nd Edition, Reprint 2012.
2. S.S. Sastry, Introductory methods of numerical analysis, PHI, 4th Edition, 2005.
3. Erwin kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
4. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2010.
5. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
6. Veerarajan T., Engineering Mathematics, Tata McGraw-Hill, New Delhi, 2008.
7. P. G. Hoel, S. C. Port and C. J. Stone, Introduction to Probability Theory, Universal Book Stall, 2003 (Reprint).
8. S. Ross, A First Course in Probability, 6th Ed., Pearson Education India, 2002.
9. W. Feller, An Introduction to Probability Theory and its Applications, Vol. 1, 3rd Ed., Wiley, 1968. Statistics

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Computer Science and Design, IV Semester

SD402 Analysis and Design of Algorithms

Course Outcome's: After completion of this course, the students are able to:

CO1: Design an algorithm for a given problem using divide-and-conquer technique, analyze running times of algorithms based on asymptotic notation.

CO2: Understand the greedy strategy and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms.

CO3: Apply the concept of dynamic programming its approach to solve problems such as 0/1 Knapsack problem, multistage graph, reliability design and Floyd-Warshall algorithm.

CO4: Use backtracking, branch & bound method for solving various problems and parallel algorithm and lower bound theory for solving algebraic problem.

CO5: Describe tree and its type, traversal technique for Graphs, various problems like P, NP, NP hard, NP completeness.

Unit I : Algorithms, Designing algorithms, analyzing algorithms, asymptotic notations, heap and heap sort. Introduction to divide and conquer technique, analysis, design and comparison of various algorithms based on this technique, example binary search, merge sort, quick sort, strassen's matrix multiplication.

Unit II : Study of Greedy strategy, examples of greedy method like optimal merge patterns, Huffman coding, minimum spanning trees, knapsack problem, job sequencing with deadlines, single source shortest path algorithm

Unit III : Concept of dynamic programming, problems based on this approach such as 0/1 knapsack, multistage graph, reliability design, Floyd-Warshall algorithm

Unit IV : Backtracking concept and its examples like 8 queen's problem, Hamiltonian cycle, Graph coloring problem etc. Introduction to branch & bound method, examples of branch and bound method like traveling salesman problem etc. Meaning of lower bound theory and its use in solving algebraic problem, introduction to parallel algorithms.

Unit V : Binary search trees, height balanced trees, 2-3 trees, B-trees, basic search and traversal techniques for trees and graphs (In order, preorder, postorder, DFS, BFS), NP-completeness.

Reference Book:

1. Cormen Thomas, Leiserson CE, Rivest RL; Introduction to Algorithms; PHI.
2. Horowitz & Sahani; Analysis & Design of Algorithm
3. Dasgupta; algorithms; TMH
4. Ullmann; Analysis & Design of Algorithm;
5. Michael T Goodrich, Roberto Tamassia, Algorithm Design, Wiley India
6. Rajesh K Shukla: Analysis and Design of Algorithms: A Beginner's Approach; Wiley

List of Experiments(expandable):

1. Write a program for Iterative and Recursive Binary Search.
2. Write a program for Merge Sort.
3. Write a program for Quick Sort.
4. Write a program for Strassen's Matrix Multiplication.
5. Write a program for optimal merge patterns.
6. Write a program for Huffman coding.
7. Write a program for minimum spanning trees using Kruskal's algorithm.
8. Write a program for minimum spanning trees using Prim's algorithm.
9. Write a program for single sources shortest path algorithm.
10. Write a program for Floyd-Warshall algorithm.
11. Write a program for traveling salesman problem.
12. Write a program for Hamiltonian cycle problem.

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Computer Science and Design, IV Semester

SD403 Software Engineering

RATIONALE: The purpose of this subject is to cover the underlying concepts and techniques used in Software Engineering & Project Management. Some of these techniques can be used in software design & its implementation.

PREREQUISITE:- The students should have at least one year of experience in programming a high-level language and databases. In addition, a familiarity with software development life cycle will be useful in studying this subject.

Course Outcome's: After completion of this course, the students are able to:

CO1: Explain different software engineering lifecycle models.

CO2: Understand and create software requirements specification.

CO3: Describe the software design process and user interface.

CO4: Understand software analysis, various testing approaches and object-oriented analysis.

CO5: Explain software project management and software maintenance practices.

Unit I: The Software Product and Software Process

Software Product and Process Characteristics, Software Process Models: Linear Sequential Model, Prototyping Model, RAD Model, Evolutionary Process Models like Incremental Model, Spiral Model, Component Assembly Model, RUP and Agile processes. Software Process customization and improvement, CMM, Product and Process Metrics

Unit II: Requirement Elicitation, Analysis, and Specification

Functional and Non-functional requirements, Requirement Sources and Elicitation Techniques, Analysis Modeling for Function-oriented and Object-oriented software development, Use case Modeling, System and Software Requirement Specifications, Requirement Validation, Traceability

Unit III: Software Design

The Software Design Process, Design Concepts and Principles, Software Modeling and UML, Architectural Design, Architectural Views and Styles, User Interface Design, Function oriented Design, SA/SD Component Based Design, Design Metrics.

Unit IV: Software Analysis and Testing

Software Static and Dynamic analysis, Code inspections, Software Testing, Fundamentals, Software Test Process, Testing Levels, Test Criteria, Test Case Design, Test Oracles, Test Techniques, Black-Box Testing, White-Box Unit Testing and Unit, Testing Frameworks, Integration Testing, System Testing and other Specialized, Testing, Test Plan, Test Metrics, Testing Tools. , Introduction to Object-oriented analysis, design and comparison with structured Software Engg.

Unit V: Software Maintenance & Software Project Measurement

Need and Types of Maintenance, Software Configuration Management (SCM), Software Change Management, Version Control, Change control and Reporting, Program Comprehension Techniques, Re-engineering, Reverse Engineering, Tool Support. Project Management Concepts, Feasibility Analysis, Project and Process Planning, Resources Allocations, Software efforts, Schedule, and Cost estimations, Project Scheduling and Tracking, Risk Assessment and Mitigation, Software Quality Assurance(SQA). Project Plan, Project Metrics.

Practical and Lab work

Lab work should include a running case study problem for which different deliverable sat the end of each phase of a software development life cycle are to be developed. Thiswill include modeling the requirements, architecture and detailed design. Subsequentlythe design models will be coded and tested. For modeling, tools like Rational Roseproducts. For coding and testing, IDE like Eclipse, Net Beans, and Visual Studio can be used.

Reference Books:

1. Pankaj Jalote ,”An Integrated Approach to Software Engineering”, Narosa Pub, 2005
2. Rajib Mall, “Fundamentals of Software Engineering” Second Edition, PHI Learning
3. R S. Pressman ,”Software Engineering: A Practitioner's Approach”, Sixth edition2006, McGraw-Hill.
4. Sommerville,”Software Enginerring”,Pearson Education.
5. Richard H.Thayer,”Software Enginerring & Project Managements”, Wiley India
6. Waman S.Jawadekar,”Software Enginerring”, TMH
7. Bob Hughes, M.Cotterell, Rajib Mall “ Software Project Management”, McGrawHill

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Computer Science and Design, IV Semester

SD404 Computer Graphics and Animation

Course Outcome's: After completion of this course, the students are able to:

CO1: Understand application of computer graphics and graphics display techniques.

CO2: Implement line, circle and curves generation methods.

CO3: Perform 2D and 3D transformation, viewing transformation and projections.

CO4: Apply line and polygon clipping and surface detection algorithms.

CO5: Explain color models, animation, image manipulation and storage.

Unit-I

Introduction to Computer Graphics: Interactive Computer Graphics, Application of Computer Graphics, Random and Raster Scan Displays, Storage Tube Graphics Display, Calligraphic Refresh Graphics Display, Flat Panel Display, Refreshing, Flickering, Interlacing, Resolution, Bit Depth, Aspect Ratio etc.

Unit-II

Scan Conversion Technique: Image representation, Line drawing: DDA, Bresenham's Algorithm. Circle Drawing: Mid-Point Circle Generation Algorithm, Bresenham's Circle Generation Algorithm, Area Filling: Boundary Filling Algorithm, Flood Filling Algorithm, Curves: Parametric Function, Bezier Method, B-Spline Method.

Unit-III

2D & 3D Transformations: Translation, Rotation, Scaling, Reflection, Shearing, Inverse Transformation, Composite Transformation, World Coordinate System, Viewing Transformation, Representation of 3D object on Screen, Parallel and Perspective Projections.

Unit-IV

Clipping: Point clipping, Line Clipping, Line Clipping Algorithm: Cohen Sutherland Line Clipping Algorithm, Midpoint Subdivision method, Liang-Barsky method, Polygon Clipping, Convex and Concave Polygon, Sutherland Hodgeman Polygon Clipping Algorithm, Weiler Atherton Polygon Clipping Algorithm, Visible Surface Detection, Back Face Removal, Hidden Surface Elimination: Z- Buffer algorithm and Painter's Algorithm.

Unit-V

Color Models: RGB, YIQ, CMY, HSV. **Computer Animation:** Introduction, Uses, key frames and tweening, types of animation, 2D and 3D animation, Principles and techniques of Animation.

Image Manipulation and Storage: Introduction to Digital Image Processing (DIP), Fundamental Steps and Components of DIP, Digital image enhancement, contrast stretching, Histogram Equalization, smoothing and median Filtering.

Text Books

1. Computer Graphics, Donald Hearn and M.P. Becker, PHI Publication.
2. Computer Graphics principle and Practice, FoleyVandam, Feiner, Hughes.
3. Principles of Computers Graphics, Rogers, TMH.
4. Computer Graphics, Sinha and Udai, TMH.
5. Digital Image Processing, Gonzalez.
6. Principle of multimedia Ranjan Parekh,TMH.

List of Experiments:

1. Write a program to draw a line using DDA line drawing algorithm.
2. Write a program to draw a line using Bresenham's line drawing algorithm.
3. Write a program to draw a circle using Midpoint circle generation algorithm.
4. Write a program to draw a circle using Bresenham's circle generation algorithm.
5. Write a program to perform boundary fill algorithm.
6. Write a program to perform flood fill algorithm.
7. Write a program to perform translation of an object.
8. Write a program to perform rotation of an object.
9. Write a program to perform scaling of an object.
10. Study of Color Models.

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Computer Science and Design, IV-Semester

SD405 Operating Systems

RATIONALE: The purpose of this subject is to cover the underlying concepts Operating System. This syllabus provides a comprehensive introduction of Operating System, Process Management, Memory Management, File Management and I/O management.

Course Outcome's: After completion of this course, the students are able to:

CO1: Explain the type, features, functions and services of operating system.

CO2: Evaluate and analyze the concept of file systems, directory structures and Disk Scheduling Algorithms.

CO3: Demonstrate CPU scheduling algorithms and memory management techniques.

CO4: Understand the concepts of Input-Output principles and implementation of Synchronization and deadlock Mechanism.

CO5: Analyzing different operating system through case studies.

UNIT 1. Introduction to Operating Systems: Function, Evolution, Different Types, Desirable Characteristics and features of an O/S, Operating Systems Services: Types of Services, Different ways of providing these Services – Utility Programs, System Calls.

UNIT II. File Systems: File Concept, User's and System Programmer's view of File System, Disk Organization, Tape Organization, Different Modules of a File System, Disk Space Allocation Methods – Contiguous, Linked, Indexed. Directory Structures, File Protection, System Calls for File Management, Disk Scheduling Algorithms.

UNIT III. CPU Scheduling : Process Concept, Scheduling Concepts, Types of Schedulers, Process State Diagram, Scheduling Algorithms, Algorithms Evaluation, System calls for Process Management; Multiple Processor Scheduling; Concept of Threads. Memory Management: Different Memory Management Techniques – Partitioning, Swapping, Segmentation, Paging, Paged Segmentation, Comparison of these techniques, Techniques for supporting the execution of large programs: Overlay, Dynamic Linking and Loading, Virtual Memory – Concept, Implementation by Demand Paging etc.

UNIT IV. Input / Output : Principles and Programming, Input/Output Problems, Asynchronous Operations, Speed gap Format conversion, I/O Interfaces, Programme Controlled I/O, Interrupt Driven I/O, Concurrent I/O. Concurrent Processes : Real and Virtual Concurrency, Mutual Exclusion, Synchronization, Inter- Process Communication, Critical Section Problem, Solution to Critical Section Problem : Semaphores – Binary and Counting Semaphores, WAIT & SIGNAL Operations and their implementation. Deadlocks: Deadlock Problems, Characterization, Prevention, Avoidance, Recovery.

UNIT V. Introduction to Network, Distributed and Multiprocessor Operating Systems. Case Studies: Unix/Linux, WINDOWS and other Contemporary Operating Systems.

Text Books

1. Silberschatz, Galvin, Gagne, "Operating System Concepts", Wiley, 9/E
2. William Stalling, "Operating Systems", Pearson Education

Reference Books

1. Andrew S. Tanenbaum, "Modern Operating Systems", 3/e, Prentice Hall
2. Maurice J. Bach, "The Design of Unix Operating System", Prentice Hall of India,
3. Bovet & Cesati, "Understanding the Linux Kernel", O'Reilly, 2/E.

List of Experiment

1. Write a program to implement FCFS CPU scheduling algorithm.
2. Write a program to implement SJF CPU scheduling algorithm.
3. Write a program to implement Priority CPU Scheduling algorithm.
4. Write a program to implement Round Robin CPU scheduling algorithm.
5. Write a program to compare various CPU Scheduling Algorithms over different Scheduling Criteria.
6. Write a program to implement classical inter process communication problem (producer consumer).
7. Write a program to implement classical inter process communication problem (Reader Writers).
8. Write a program to implement classical inter process communication problem (Dining Philosophers).
9. Write a program to implement & Compare various page replacement algorithms.
10. Write a program to implement & Compare various Disk & Drum scheduling Algorithms
11. Write a program to implement Banker's algorithms.
12. Write a program to implement Remote Procedure Call(RPC).
13. Write a Devices Drivers for any Device or pheriperal.

RAJIV GANDHI PROUDYOGIKI VISHWAVIDYALAYA, BHOPAL

New Scheme Based On AICTE Flexible Curricula

Computer Science and Design, IV-Semester

SD406 Programming Practices/ Java Lab

Objective: To introduce and understand students to programming concepts and techniques using the Java language and programming environment, class, objects, also learn about lifetime, scope and the initialization mechanism of variables and improve the ability general problem solving abilities in programming. Be able to use the Java SDK environment to create, debug and run simple Java program.

Course Outcome's: After completion of this course, the students are able to:

- CO1: Implement various object oriented feature using java.
- CO2: Develop java programs using collection framework.
- CO3: Apply synchronization in multithreading environment.
- CO4: Create web application using JSP and servlet.
- CO5: Understand the concept of J2ME,EJB and XML.

Basic Java Features - C++ Vs JAVA, JAVA virtual machine, Constant & Variables, Data Types, Class, Methods, Objects, Strings and Arrays, Type Casting, Operators, Precedence relations, Control Statements, Exception Handling, File and Streams, Visibility, Constructors, Operator and Methods Overloading, Static Members, Inheritance: Polymorphism, Abstract methods and Classes

Java Collective Frame Work - Data Structures: Introduction, Type-Wrapper Classes for Primitive Types, Dynamic Memory Allocation, Linked List, Stack, Queues, Trees, Generics: Introduction, Overloading Generic Methods, Generic Classes, Collections: Interface Collection and Class Collections, Lists, Array List and Iterator, Linked List, Vector. Collections Algorithms: Algorithm sorts, Algorithm shuffle, Algorithms reverse, fill, copy, max and min Algorithm binary Search, Algorithms add All, Stack Class of Package java. Util, Class Priority Queue and Interface Queue, Maps, Properties Class, Unmodifiable Collections.

Advance Java Features - Multithreading: Thread States, Priorities and Thread Scheduling, Life Cycle of a Thread, Thread Synchronization, Creating and Executing Threads, Multithreading with GUI, Monitors and Monitor Locks. Networking: Manipulating URLs, Reading a file on a Web Server, Socket programming, Security and the Network, RMI, Networking, Accessing Databases with JDBC: Relational Database, SQL, MySQL, Oracle

Advance Java Technologies - Servlets: Overview and Architecture, Setting Up the Apache Tomcat Server, Handling HTTP get Requests, Deploying a web Application, Multitier Applications, Using JDBC from a Servlet, Java Server Pages (JSP): Overview, First JSP Example, Implicit Objects, Scripting, Standard Actions, Directives, Multimedia: Applets and Application: Loading, Displaying and Scaling Images, Animating a Series of Images, Loading and playing Audio clips

Advance Web/Internet Programming (Overview): J2ME, J2EE, EJB, XML.

Reference Books:

1. E. Balaguruswamy, "Programming In Java"; TMH Publications
2. The Complete Reference: Herbert Schildt, TMH
3. Deitel & Deitel, "JAVA, How to Program"; PHI, Pearson.
4. Cay Horstmann, Big JAVA, Wiley India.
5. Merlin Hughes, et al; Java Network Programming, Manning Publications

List of Experiments:

1. Installation of J2SDK
2. Write a program to show Scope of Variables

3. Write a program to show Concept of CLASS in JAVA
4. Write a program to show Type Casting in JAVA
5. Write a program to show How Exception Handling is in JAVA
6. Write a Program to show Inheritance
7. Write a program to show Polymorphism
8. Write a program to show Access Specifiers (Public, Private, Protected) in JAVA
9. Write a program to show use and Advantages of CONTRUCTOR
10. Write a program to show Interfacing between two classes
11. Write a program to Add a Class to a Package
12. Write a program to show Life Cycle of a Thread
13. Write a program to demonstrate AWT.
14. Write a program to Hide a Class
15. Write a Program to show Data Base Connectivity Using JAVA
16. Write a Program to show "HELLO JAVA " in Explorer using Applet
17. Write a Program to show Connectivity using JDBC
18. Write a program to demonstrate multithreading using Java.
19. Write a program to demonstrate applet life cycle.
20. Write a program to demonstrate concept of servlet.