School of Information Technology
Rajiv Gandhi ProudyogikiVishwavidyalaya, Bhopal
New Scheme of Examination as per AICTE Flexible Curricula
IV Semester Bachelor of Technology (B.Tech.)
Computer Science and Engineering (Data Science)
Syllabus

CD 401 (THEORY OF COMPUTATION)

UNIT I: Fundamentals: Strings, Alphabet, Language, Operations, Finite state machine, definitions, finite automaton model, acceptance of strings, and languages, deterministic finite automaton and nondeterministic finite automaton, transition diagrams and Language recognizers.
Finite Automata: NFA with Î transitions - Significance, acceptance of languages. Conversions & Equivalence: Equivalence between NFA with and without Î transitions, NFA to DFA conversion, minimization of FSM, equivalence between two FSM's, Finite Automata with output- Moore and Melay machines.

UNIT II: Regular Languages: Regular sets, regular expressions, identity rules, Constructing finite Automata for a given regular expressions, Conversion of Finite Automata to Regular expressions. Pumping lemma of regular sets, closure properties of regular sets.
Grammar Formalism: Regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and FA, inter conversion, Context free grammar, derivation trees, and sentential forms. Right most and left most derivation of strings.

UNIT III: Context Free Grammars: Ambiguity in context free grammars. Minimization of Context Free Grammars. Chomsky normal form, Greiback normal form, Pumping Lemma for Context Free Languages. Enumeration of properties of CFL.
Push Down Automata: Push down automata, definition, model, acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence. Equivalence of CFL and PDA, interconversion. (Proofs not required). Introduction to DCFL and DPDA.

UNIT IV: Turing Machine: Turing Machine, definition, model, design of TM, Computable functions, recursively enumerable languages. Church's hypothesis, counter machine, types of Turing machines (proofs not required). , linear bounded automata and context sensitive language

UNIT V: Computability Theory: Chomsky hierarchy of languages, decidability of, problems, Universal Turing Machine, undecidability of posts. Correspondence problem, Turing reducibility, Definition of P and NP problems, NP complete and NP hard problems.

Reference Books-

1. "Introduction to Automata Theory Languages and Computation". Hopcroft H.E. and Ullman J. D. Pearson Education.
2. Introduction to Theory of Computation –Sipser 2nd edition Thomson.
3. Introduction to Formal Languages, Automata Theory and Computation Kamala Krithivasan, Rama
4. Introduction to Computer Theory, Daniel I.A. Cohen, John Wiley.
5. Theory of Computation : A Problem – Solving Approach- Kavi Mahesh, Wiley India Pvt. Ltd.
6. "Elements of Theory of Computation", Lewis H.P. &Papadimition C.H. Pearson /PHI.
7. Theory of Computer Science – Automata languages and computation -Mishra and Chandrashekaran, 2nd edition, PHI.
8. Introduction to languages and the Theory of Computation, John C Martin, TMH.

Course Outcomes-

On successful completion of the course, the students will be able to:

CO1: Explain basic concepts in formal language theory, grammars, automata theory, computability theory, and complexity theory.

CO2: Demonstrate abstract models of computing, including deterministic (DFA), non-deterministic (NFA), Push Down Automata(PDA) and Turing (TM) machine models and their power to recognize the languages.

CO3: Explain the application of machine models and descriptors to compiler theory and parsing. Students will be able to relate practical problems to languages, automata, computability, and complexity.

CO4: Demonstrate an increased level of mathematical sophistication.

CO5: Apply mathematical and formal techniques for solving problems in computer science. Students will be able to explain the relationship among language classes and grammars with the help of Chomsky Hierarchy

CD 402 (OPERATING SYSTEMS)

UNIT I- Introduction: Concept of Operating Systems (OS), Generations of OS, Types of OS, OS Services, Interrupt handling and System Calls, Basic architectural concepts of an OS, Concept of Virtual Machine, Resource Manager view, process view and hierarchical view of an OS.

UNIT II- Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.
Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time. Scheduling algorithms: Pre-emptive and non-pre-emptive, FCFS, SJF, RR; Multiprocessor scheduling; Real Time scheduling: RM and EDF.

UNIT III - I/O Hardware: I/O devices, Device controllers, Direct Memory Access, Principles of I/O.
File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed),Free-space management (bit vector, linked list, grouping), directory implementation(linear list, hash table), efficiency and performance. Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.

UNIT IV- Inter-process Communication: Concurrent processes, precedence graphs, Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Semaphores, Strict Alternation, Peterson's Solution, The Producer / Consumer Problem, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem, Barber's shop problem.
Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery. Concurrent Programming: Critical region, conditional critical region, monitors, concurrent languages, communicating sequential process (CSP)

UNIT V- Memory Management: Basic concept, Logical and Physical address maps, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction. Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page allocation, Partitioning, Paging, Page fault, Working Set, Segmentation, Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).
Case study: UNIX/Linux OS file system, shell, filters, shell programming, programming with the standard I/O, UNIX/Linux system calls.

Reference Books-

1. Operating System Concepts Essentials. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne.
2. Operating Systems: Internals and Design Principles. William Stallings.
3. Operating System: A Design-oriented Approach. Charles Patrick Crowley.
4. Operating Systems: A Modern Perspective. Gary J. Nutt.
5. Design of the Unix Operating Systems. Maurice J. Bach.
6. Understanding the Linux Kernel, Daniel Pierre Bovet, .itaseC ocraM

Course Outcomes-

On successful completion of the course, the students will be able to:

CO1: Understand the various OS functionalities and acquire the knowledge of various types of OS

CO2: Design and Implement CPU scheduling algorithms to meet and validate the scheduling criteria

CO3: Implement directories and perform various operations on files/directories

CO4: Apply the acquired knowledge of deadlocks to design and implement deadlock free computer programs as well as understand the issues in inter process communication

CO5: Understand how memory is allocated to processes by OS and Implement algorithms related to main and Virtual memory techniques

Suggested List of Experiments-

1. Unix/Linux commands (files directory, data manipulation, network communication etc), shell programming and vi editor
2. C programs for implementation of the following:
    a. Scheduling Algorithms
    b. Shared memory
    c. Thread and Multi Thread
    d. Inter Process Communication
    e. Deadlock Avoidance and Deadlock Detection
    f. Semaphore
    g. Memory Management
    h. Indexing and Hashing
3. C Programs for implementing certain commands and a shell like Unix/Linux system shell, using the Unix/Linux System calls.

CD 403 (ANALYSIS & DESIGN OF ALGORITHMS)

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, etc.

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

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 (Inorder, preorder, postorder, DFS, BFS), NP-completeness

Reference Books-

1. Coremen 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, Robarto Tamassia, Algorithm Design, Wiely India

Course Outcomes-

After the completion of this course, the students will be able to:

1 Implement sorting and searching algorithm
2 Experiment with techniques for obtaining maximum output with minnium efforts
3 Make use of dynamic programming for finding
4 Solve 8 queen's problem and others of the kind for application in real world scenarios .
5 Distinguish between NP hard and NP complete problems and develop their solutions

Suggested List of Experiments-

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 Floye-Warshal algorithm.
11. Write a program for traveling salesman problem.
12. Write a program for Hamiltonian cycle problem.

CD 404 (SOFTWARE ENGINEERING)

UNIT I- Introduction: Programming in the small vs. programming in the large; software project failures and importance of software quality and timely availability; of software engineering towards successful execution of large software projects; emergence of software engineering as a discipline, Software Engineering Historical Development from Jackson Structured Programming to Agile Development.

UNIT II- Software Project Management: Basic concepts of life cycle models – different models and milestones; software project planning –identification of activities and resources; concepts of feasibility study; techniques for estimation of schedule and effort; software cost estimation models and concepts of software engineering economics; techniques of software project control and reporting; introduction to measurement of software size; introduction to the concepts of risk and its mitigation; configuration management.

UNIT III- Software Quality Management and Reliability: Software quality; Garvin's quality dimensions, McCall's quality factor, ISO 9126 quality factor; Software Quality Dilemma; Introduction to Capability Maturity Models (CMM and CMMI); Introduction to software reliability, reliability models and estimation.

UNIT IV- Software Requirements Analysis, Design and Construction: Introduction to Software Requirements Specifications (SRS) and requirement elicitation techniques; techniques for requirement modelling – decision tables, event tables, state transition tables, Petri nets; requirements documentation through use cases; introduction to UML, introduction to software metrics and metrics-based control methods; measures of code and design quality.
Object Oriented Analysis, Design and Construction: Concepts -- the principles of abstraction, modularity, specification, encapsulation and information hiding; concepts of abstract data type; Class Responsibility Collaborator (CRC) model; quality of design; design measurements; concepts of design patterns; Refactoring; object-oriented construction principles; object oriented metrics.

UNIT V- Software Testing: Introduction to faults and failures; basic testing concepts; concepts of verification and validation; black box and white box tests; white box test coverage – code coverage, condition coverage, branch coverage; basic concepts of black-box tests – equivalence classes, boundary value tests, usage of state tables; testing use cases; transaction based testing; testing for non-functional requirements – volume, performance and efficiency; concepts of inspection; Unit Testing, Integration Testing, System Testing and Acceptance Testing.

Agile Software Engineering: Concepts of Agile Methods, Extreme Programming; Agile Process Model - Scrum, Feature; Scenarios and Stories

Reference Books-

1. Software Engineering, Ian Sommerville
2. Software Engineering A Practitioner's Approach, Roggers S. Pressman and Bruce R. Maxim.
1. The Essentials of Modern Software Engineering: Free the Practices from the Method Prisons!, Ivar Jacobson, Harold "Bud" Lawson, Pan-Wei Ng, Paul E. McMahon and Michael Goedicke.
2. Fundamentals of Software Engineering, Carlo Ghezzi, Jazayeri Mehdi and Mandrioli Dino.
3. Software Requirements and Specification: A Lexicon of Practice, Principles and Prejudices, Michael Jackson.
4. The Unified Development Process, Ivar Jacobson, Grady Booch and James Rumbaugh.
5. Design Patterns: Elements of Object-Oriented Reusable Software, Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides.
6. Software Metrics: A Rigorous and Practical Approach, Norman E Fenton and Shari Lawrence Pfleeger.
7. Software Engineering: Theory and Practice, Shari Lawrence Pfleeger and Joanne M. Atlee.
8. Object-Oriented Software Construction, Bertrand Meyer.
9. Object Oriented Software Engineering: A Use Case Driven Approach --Ivar Jacobson.
10. Touch of Class: Learning to Program Well with Objects and Contracts --Bertrand Meyer.

11. UML Distilled: A Brief Guide to the Standard Object Modeling Language --Martin Fowler.
12. Introduction to Business Domains for Software Engineers, Manoj Kumar Lal
13. Knowledge Driven Development – Bridging Waterfall and Agile Methodologies –- Manoj Kumar Lal

Course Outcomes-

On successful completion of the course, the students will be able to:

CO1: Understand software engineering process and practices
CO2: Distinguish between various software process models and cost estimation models and choose appropriate model for project
CO3: Understand various software quality attributes and software reliability metrics
CO4: Develop the SRS document for project
CO5: Apply the knowledge of testing techniques and strategies to test developed software

Suggested List of Experiments-

Development of requirements specification, function-oriented design using SA/SD, object-oriented design using UML, test case design, implementation using C++ and testing. Use of appropriate CASE tools and other tools such as configuration management tools, program analysis tools in the software life cycle.

CD 405 (DATA MINING AND WAREHOUSING)

COURSE OBJECTIVES: Student should understand the value of Historical data and data mining in solving real-world problems. Student should become affluent with the basic Supervised and unsupervised learning algorithms commonly used in data mining . Student develops the skill in using data mining for solving real-world problems.

Unit I

Data Warehousing: Introduction, Delivery Process, Data warehouse Architecture, Data Preprocessing: Data cleaning, Data Integration and transformation, Data reduction. Data warehouse Design: Dataware house schema, Partitioning strategy Data warehouse Implementation, Data Marts, Meta Data, Example of a Multidimensional Data model. Introduction to Pattern Warehousing.

Unit II

OLAP Systems: Basic concepts, OLAP queries, Types of OLAP servers, OLAP operations etc. Data Warehouse Hardware and Operational Design: Security, Backup And Recovery,

Unit III

Introduction to Data& Data Mining :Data Types, Quality of data, Data Preprocessing, Similarity measures, Summary statistics, Data distributions, Basic data mining tasks, Data Mining V/s knowledge discovery in databases. Issues in Data mining. Introduction to Fuzzy sets and fuzzy logic.

Unit IV

Supervised Learning: Classification: Statistical-based algorithms, Distance-based algorithms, Decision tree-based algorithms, Neural network-based algorithms, Rule-based algorithms, Probabilistic Classifiers

Unit V

Clustering & Association Rule mining : Hierarchical algorithms, Partitional algorithms, Clustering large databases – BIRCH, DBSCAN, CURE algorithms.Association rules : Parallel and distributed algorithms such as Apriori and FP growth algorithms.

Text Books:

1. Pang – ningTan , Steinbach & Kumar, “Introduction to Data Mining”, Pearson Edu, 2019.
2. Jaiwei Han, Micheline Kamber, “Data Mining : Concepts and Techniques”, Morgan Kaufmann Publishers.

Reference Books:

1. Margaret H. Dunham, “Data Mining : Introductory and Advanced topics”, Pearson Edu., 2009.
2. Anahory& Murray, “Data Warehousing in the Real World”, Pearson Edu., 2009.

Course Outcomes:

After completion of this course, the students would be able to:

CO1. Understand the need of designing Enterprise data warehouses andwill be enabled to approach business problems analytically by identifying opportunities to derive business.

CO2. Compare and contrast, various methods for storing & retrieving data from different data sources/repository.

CO3. Ascertain the application of data mining in various areas and Preprocess the given data and visualize it for a given application or data exploration/mining task

CO4. Apply supervised learning methods to given data sets such as classification and its various types.

CO5. Apply Unsupervised learning methods to given data sets such as clustering and its various types.

CO6 Apply Association rule Mining to various domains.

CD-406 (COMPUTER PROGRAMMING IV)
(JAVA)

Unit I-Overview of Java, Installation, First Simple Program, Compilation process, JavaKeywords,Identifiers, Literals, Comments, Data Types, Variables, Dynamic initialization, typeconversion and casting, Operators, Control Statements.

Unit II-Declaring Objects, Introducing Methods, Constructors, this Keyword, GarbageCollection, finalize Method, Overloading Methods, Overloading Constructors, Using Objects asParameters, Inheritance, Creating a Multilevel Hierarchy, Packages and Interfaces, ExceptionHandling, Multithreaded

Unit III-The Applet Class: Applet Basics, The Applet Class, Applet Architecture, AppletInitialization and Termination , Simple Applet Display Methods, Simple Banner Applet, Usingthe Status Window, The HTML APPLET Tag, Passing Parameters to Applets, Improving theBanner Applet.

Unit IV-Introducing the AWT: Working with Windows, Graphics, and Text, AWT Classes,Window Fundamentals, Component, Container, Panel, Frame, Working with Frame Windows,Handling Events in a Frame Window, AWT Controls, Layout Managers, and Menus, Addingand Removing Controls, Grid Layout, Border Layout, introduction to swing and servlet.

Unit V-Event Handling, Two Event Handling Mechanisms, The Delegation Event Model,Events, Event Sources, Event Listeners, Event Classes, The Mouse Event Class and others,JDBC: JDBCODBC bridge, the connectivity model, the driver manager, navigating the result setobject contents, the JDBC exceptional classes, connecting to remote database.

References:

1. E. Balagurusamy, "Programming with java A Primer", McGrawHill.
2. Sharanam Shah, " Core Java 8 for Beginners", Shroff Publisher.
3. Naughton&Schildt, "The Complete Reference Java 2", Tata McGraw Hill.
4. Horstmann& Cornell, "Core Java 2" (Vol I &II), Pearson.

Course Outcomes:
On the completion of this course students will be able to understand:
1. The concepts of Java programming
2. The basic terminology used in computer programming and write, compile and debugprograms in JAVA language.
3. The different data types, decision structures, loops, functions to design Java programs.
4. Develop program using the java collection API as well as the java standard class library.
5. Develop Java applets

List of Experiments:
1. Write a program that accepts two numbers from the user and print their sum.
2. Write a program to calculate addition of two number using prototyping of methods.
3. Program to demonstrate function overloading for calculation of average.
4. Program to demonstrating overloaded constructor for calculating box volume.
5. Program to show the detail of students using concept of inheritance.
6. Program to demonstrate package concept.
7. Program to demonstrate implementation of an interface which contains two methods declaration square and cube.
8. Program to demonstrate exception handling in case of division by zero error.
9. Program to demonstrate multithreading.
10. Program to demonstrate JDBC concept using create a GUI based application for student information.
11. Program to display "Hello World" in web browser using applet.
12. Program to add user controls to applets.

13. Write a program to create an application using concept of swing.
14. Program to demonstrate student registration functionality using servlets with session management.