# R22-A

## CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

**AICTE Model Curriculum R 22-A with effect from AY 2025-26**

B.E. CSE (Artificial Intelligence and Machine Learning)

**SEMESTER – III**

| S. No | Course Code | Category | Title of the Course | Scheme of Instruction | | | Scheme of Examination | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Hours per Week | | | Duration of SEE in Hours | Maximum Marks | | |
| | | | | L | T | P/D | | CIE | SEE | |
| **THEORY** | | | | | | | | | | |
| 1 | 22CAC01N | PCC | Agile Software Development | 3 | - | - | 3 | 40 | 60 | 3 |
| 2 | 22CSC06 | PCC | Discrete Structures | 4 | - | - | 3 | 40 | 60 | 4 |
| 3 | 22CSC14N | PCC | Design and Analysis of Algorithms | 3 | - | - | 3 | 40 | 60 | 3 |
| 4 | 22ITC02N | PCC | Java Programming | 3 | - | - | 3 | 40 | 60 | 3 |
| 5 | 22ECC38 | BES | Analog and Digital Electronics | 3 | - | - | 3 | 40 | 60 | 3 |
| **PRACTICAL** | | | | | | | | | | |
| 6 | 22CAC02N | PCC | Agile Software Development Lab | - | - | 3 | 3 | 50 | 50 | 1.5 |
| 7 | 22CSC58 | PCC | Design and Analysis of Algorithms Lab | - | - | 3 | 3 | 50 | 50 | 1.5 |
| 8 | 22ITC03N | PCC | Java Programming Lab | - | - | 3 | 3 | 50 | 50 | 1.5 |
| 9 | 22CAI01 | | MOOCs/ Training/ Internship | 3-4 weeks / 90 Hours | | | - | - | - | 2 |
| **TOTAL** | | | | **16** | **0** | **09** | **-** | **350** | **450** | **22.5** |

**L: Lecture** **T: Tutorial** **P: Practical**

**CIE - Continuous Internal Evaluation** **SEE - Semester End Examination**

**22CAC01N**

## AGILE SOFTWARE DEVELOPMENT

| | |
|---|---|
| Instruction | 3L Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 60 Marks |
| CIE | 40 Marks |
| Credits | 3 |

**Pre-requisites**: Problem Solving and Programming using C, Python Programming

**COURSE OBJECTIVES**: This course aims to:

1. Demonstrate the ability to participate effectively in Agile Process for Software Development.
2. Explain the Purpose behind common Agile practices.
3. Apply Agile Principles and Values to a given real time problem.

**COURSE OUTCOMES**: After the completion of this course, the student will be able to

1. Interpret the concept of agile software engineering and its advantages in software development.
2. Analyse the core practices behind several specific agile methodologies.
3. Identify the roles and responsibilities in agile projects and their difference from projects following traditional methodologies.
4. Access implications of functional testing, unit testing, and continuous integration.
5. Determine the role of design principles in agile software design.

**CO-PO & PSO ARTICULATION MATRIX**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 3 | 2 | 2 | 1 | - | - | 1 | - | 1 | - | - | 3 | 2 | 2 |
| CO 2 | 3 | 3 | 2 | 2 | 2 | 1 | 2 | - | 2 | 1 | - | 3 | 2 | 3 |
| CO 3 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 2 | 2 | 1 | - | 2 | 2 | 2 |
| CO 4 | 3 | 3 | 3 | 3 | 3 | 2 | 3 | 2 | 3 | 2 | 1 | 3 | 3 | 3 |
| CO 5 | 3 | 3 | 2 | 2 | 2 | 3 | 3 | 2 | 3 | 3 | 2 | 3 | 2 | 3 |

**UNIT I**

**Traditional Software Engineering Practices:** A Generic Process Model, Process assessment and Improvement, Prescriptive Process Models: Waterfall Model, Incremental Process Models, RAD Model, Evolutionary Process Models - Prototyping, The Spiral Model, Specialized Process Models.

**UNIT II**

**Introduction to Agile Development:** Need of Agile software development, agile context– Manifesto, Principles, Methods, Values, Roles, Artifacts, Stakeholders, and challenges. Business benefits of software agility.

**UNIT III**

**Agile Methodologies:** Need of scrum, Scrum practices –Working of scrum, Project velocity, Burn down chart, Sprint backlog, Sprint planning and retrospective, Daily scrum, Scrum roles– Product Owner, Scrum Master, Scrum Team. Extreme Programming- Core principles, values and practices. Kanban, Feature-driven development, Lean software development.

**UNIT IV**

**Project Planning:** Recognizing the structure of an agile team– Programmers, Managers, Customers. User stories–Definition, Characteristics and content. Estimation– Planning poker, Prioritizing, and selecting user stories with the customer, projecting team velocity for releases and iterations.

**Project Design:** Fundamentals, Design principles–Single responsibility, Open-closed, Liskov substitution, Dependency-inversion, Interface-segregation.

**UNIT V**

**Testing:** The Agile lifecycle and its impact on testing, Test driven development– Acceptance tests and verifying stories, writing a user acceptance test, Developing effective test suites, Continuous integration, Code refactoring. Risk based testing, Regression tests, Test automation.

**TEXT BOOKS:**


1. Ken Schawber, Mike Beedle, “Agile Software Development with Scrum”, International Edition, Pearson.
2. Robert C. Martin, “Agile Software Development, Principles, Patterns and Practices”, First International Edition, Prentice Hall.
3. Roger S. Pressman “Software Engineering: A practitioner's approach”, McGraw Hill, 9th Edition, 2010.


**SUGGESTED READING:**


1. Lisa Crispin, Janet Gregory, “Agile Testing: A Practical Guide for Testers and Agile Teams”, International edition, Addison Wesley.
2. Pedro M. Santos, Marco Consolaro, and Alessandro Di Gioia, “Agile Technical Practices Distilled: A learning journey in technical practices and principles of software design”, First edition, Packt Publisher.
3. Alistair Cockburn, “Agile Software Development: The Cooperative Game”, 2nd Edition, Addison-Wesley
4. “The Complete Guide to Agile Software Development” https://clearbridgemobile.com/complete-guideagile-software-development/
5. “Agile Fundamentals Ebook: A Complete Guide for Beginners”, https://agileken.com/agile- fundamentals ebook/


**Online Resources:**


**1.** “Agile Software Development”, https://www.edx.org/course/agile-software-development Accessed on August 27, 2021.
2. “Agile Software Development”, https://www.coursera.org/learn/agile-software-development Accessed on August 27, 2021.

**22CSC06**

## DISCRETE STRUCTURES

| | |
|---|---|
| Instruction | 4L Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 60 Marks |
| CIE | 40 Marks |
| Credits | 4 |

**Course Objectives:**

This course aims to:

1. Introduce Propositional and Predicate Logic.
2. Introduce various proof techniques for validation of arguments.
3. Develop an understanding of counting, functions and relations.
4. Familiarize with fundamental notions and applicability of graph theory and algebraic systems.

**Course Outcomes:**

Upon completion of this course, students will be able to:

1. Describe rules of inference for Propositional and Predicate logic.
2. Demonstrate use of Set Theory, Venn Diagrams, relations, and functions in Real-world scenarios.
3. Model solutions using Generating Functions and Recurrence Relations.
4. Determine the properties of graphs and trees to solve problems arising in computer science applications.
5. Distinguish between groups, semi groups and monoids in algebraic systems

**CO-PO & PSO ARTICULATION MATRIX**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 2 | 2 | 1 | 1 | - | - | - | - | 1 | - | - | 3 | - | 2 |
| CO 2 | 2 | 1 | - | 1 | - | - | - | - | - | - | 1 | 3 | 2 | 2 |
| CO 3 | 2 | 2 | - | 1 | - | - | - | - | - | - | - | 3 | 2 | 2 |
| CO 4 | 2 | 1 | - | 1 | - | - | - | - | 1 | - | - | 3 | 2 | 2 |
| CO 5 | 2 | 1 | - | 1 | - | - | - | - | - | - | - | 3 | - | 2 |

**UNIT – I**

Introduction to Propositional Calculus: Basic Connectives and Truth tables, Logical Equivalence: Laws of Logic, Logical Implication; Rules of Inference. Predicates: The Use of Quantifiers, Quantifiers, Definitions and the Proofs of Theorems

**UNIT – II**

Sets: Sets and Subsets, Operations on sets and the Laws of Set Theory, Counting and Venn Diagrams. Relations and Functions: Cartesian Products and Relations. Partial ordering relations, POSET, Hasse diagrams, Lattices as Partially Ordered Sets, Equivalence relations. Pigeon hole principle. Functions: Types of Functions, Composition of functions and Inverse of functions

**UNIT – III**

Fundamental Principles of counting: The Rules of Sum and Product, Permutations, Combinations, Binomial Theorem; Generating Functions: Generating Functions, Calculating Coefficient of generating functions; Recurrence Relations: The First Order Linear Recurrence Relation, Second Order Linear. Homogeneous Recurrence relations with constant coefficients, Non Homogeneous Recurrence relations.

**UNIT – IV**

Introduction to Graphs: Graphs and their basic properties- degree, path, cycle, Sub graphs, Complements and Graph Isomorphism, Euler trails and circuits, Hamiltonian paths and cycles, Planar graphs, Euler formula, Graph Coloring and Chromatic polynomial, Matching, Applications. Trees: Definitions, Properties, Rooted Trees, Spanning Trees, Minimum Spanning trees: Algorithms of Kruskal and Prim.

**UNIT - V**

Algebraic Structures: Algebraic Systems, Examples and General Properties, Semi groups and Monoids. Groups: Definitions and Examples, Subgroups, Homomorphisms and cyclic groups.

**Text Books**:

1.Ralph P. Grimaldi, “Discrete and Combinatorial Mathematics”, An Applied Introduction, 5th edition, Pearson Education, 2016.

2.Rosen, K. H., “Discrete Mathematics and Its Applications”, 8th Edition, ISBN10: 125967651X ISBN13: 9781259676512, 2019

**Suggested Reading:**

1. Singh, S.B., “Discrete Mathematics”, Khanna Book Publishing Company, New Delhi., 3rd Edition, 2019

2. R. K. Bisht, H. S. Dhami, “Discrete Mathematics”, Oxford University Press, 2015.

3. David D. Railey, Kenny A. Hunt, “Computational Thinking for the Modern Problem Solving”, CRC Press, 2014

4. J. P. Tremblay, R. Manohar, “Discrete Mathematical Structures with Applications to Computer Science”, TATA Mc Graw-Hill Edition, 2017.

5. Joe L. Mott, Abraham Kandel, Theodore P. Baker, “Discrete Mathematics for Computer Scientists & Mathematicians”, 2nd Edition, PHI, 2015.

**Online Resources:**

1. .https://nptel.ac.in/courses/111107058/
2. https://nptel-discrete-mathematics-5217

**22CSC14N**

## DESIGN AND ANALYSIS OF ALGORITHMS

| | |
|---|---|
| Instruction | 3L Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 60 Marks |
| CIE | 40 Marks |
| Credits | 3 |

Prerequisite: Basics of Data structures and algorithms.

**Course Objectives:**

This course aims to:

1. Provide an introduction to formalisms to understand, analyze and denote time complexities of algorithms.
2. Introduce the different algorithmic approaches for problem solving through numerous example problems.
3. Provide some theoretical grounding in terms of finding the lower bounds of algorithms and the NP-completeness.

**Course Outcomes:**

Upon completion of this course, students will be able to:

1. Identify and apply asymptotic notations and recurrence-solving techniques to analyze the performance of recursive algorithms
2. Apply greedy and dynamic programming strategies to solve optimization problems and identify the most suitable design approach based on problem characteristics.
3. Implement backtracking and branch-and-bound techniques to solve combinatorial and decision problems and evaluate their efficiency.
4. Solve and evaluate the performance of graph traversal and shortest path algorithms.
5. Demonstrate NP-completeness through problem reductions and complexity classes.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 3 | 3 | - | - | 2 | - | - | - | - | - | - | 3 | - | 2 |
| CO 2 | 3 | 3 | 2 | - | 2 | - | - | - | - | - | - | 3 | 3 | 2 |
| CO 3 | 3 | 3 | 2 | - | 2 | - | - | - | - | - | - | 3 | - | 2 |
| CO 4 | 3 | 3 | - | - | 2 | - | - | - | - | - | - | 3 | 3 | 2 |
| CO 5 | 2 | 3 | - | - | 2 | - | - | - | - | - | - | 3 | - | 2 |

**UNIT - I**

Introduction: Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior. Performance measurements of Algorithm, Time and space trade-offs. Divide and Conquer: The general method, Minimum and Maximum Problem Analysis of recursive algorithms through recurrence relations: Iterative/Expansion method, Recursion tree method and Masters' theorem.

**UNIT - II**

Greedy Algorithms: The general method, Knapsack Problem, Huffman Codes, Job scheduling with deadlines. Dynamic Programming: The general method, 0/1 Knapsack, Travelling Salesman Problem, Matrix chain multiplication, Longest Common subsequence.

**UNIT - III**

Backtracking: The general Method, 8-Queens Problem, Graph Coloring, Hamiltonian Cycle. Branch-and-Bound: The general method, FIFO branch and bound, LC branch and bound, 0/1 Knapsack Problem using FIFO branch and bound, Travelling Salesperson problem using LC branch and bound.

**UNIT - IV**

Graph Algorithms: Applications of DFS: Bi-Connected components, strongly connected components, topological sorting. Shortest Path Algorithms: Dijkstra's, Bellman-Ford, Floyd-Warshall

**UNIT - V**

Theory of NP-Completeness: Polynomial time, Polynomial time verification, P, NP, NP-hard and NP-Complete classes, NP-Completeness and Reducibility. Standard NP-Complete Problems and Reduction Techniques: The Clique Problem, Vertex-Cover Problem.

**Text Books:**

1. Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, “Introduction to Algorithms”, MIT Press/McGraw-Hill, 4th Edition, 2022.
2. E. Horowitz, sartaj sahni and sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Universities Press, 2008.

**Suggested Reading:**

1. Michael T Goodrich and Roberto Tamassia, “Algorithm Design: Foundations, Analysis”, and Internet Examples, Wiley Second Edition.

**Online Resources:**

1. https://nptel.ac.in/courses/106101060/

**22ITC02N**

# JAVA PROGRAMMING

(Common to CSE, IT, AI&DS , CET and allied branches)

| | |
|---|---|
| Instruction | 3L Hours per week |
| Duration of SEE | 3 Hours |
| SEE | 60Marks |
| CIE | 40Marks |
| Credits | 3 |

**Course Objectives**

The course aims to:

1. Introduce the fundamental concepts of Object-Oriented Programming (OOP).
2. Guide students through the process of creating and managing classes and objects.
3. Explain and demonstrate the use of inheritance and polymorphism.
4. Teach effective handling of runtime exceptions and the basics of multithreading.
5. Provide hands-on experience with Java's IO package for application development.

**Course Outcomes**

By the end of this course, students will be able to:

1. Apply OOP concepts to develop structured Java applications.
2. Utilize inheritance and interfaces to enhance code reusability and flexibility.
3. Implement exception handling and multithreading to manage complex program flows.
4. Build applications using the Java Collection Framework.
5. Develop programs that handle input and output operations using the IO package.

**CO-PO & PSO ARTICULATION MATRIX**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 2 | 2 | 3 | 3 | 2 | 2 | - | 1 | 2 | 1 | 2 | 2 | 2 | 2 |
| CO 2 | 2 | 2 | 3 | 2 | 2 | 1 | - | 1 | 2 | 1 | 2 | 2 | 2 | 2 |
| CO 3 | 2 | 2 | 3 | 2 | 2 | 1 | - | 1 | 2 | 1 | 2 | 2 | 2 | 2 |
| CO 4 | 2 | 2 | 3 | 2 | 2 | 1 | - | 1 | 2 | 1 | 2 | 2 | 2 | 2 |
| CO 5 | 2 | 2 | 3 | 2 | 3 | 1 | - | 1 | 2 | 1 | 2 | 2 | 3 | 3 |

**UNIT-I**

**Introduction to Java:** Procedural and object-oriented programming paradigms, Principles, Features, Basic structure a java program, Java Primitive Data Types, Basic Operators, Flow-control statements. Defining Classes, Adding Instance Fields and Methods, Object Creation, Constructors, Access Modifiers, Method Overloading and Constructor Overloading, Use of static and final keyword, Arrays, Strings and String Tokenizer, Scanner.

**UNIT-II**

**Inheritances and Packages:** Types of Inheritance, super keyword, preventing inheritance, the Object class, method overriding and dynamic method dispatch, abstract classes and methods. Interfaces, Interfaces vs. Abstract classes, Inner classes and types, Packages, Creating and Accessing a Package, Understanding CLASSPATH, importing packages.

**UNIT-III**

**Exception Handling and Threading:** What are exceptions, Error vs. Exception, usage of try, catch, throw, throws and finally clauses, Multithreading in Java, Life cycle of Thread, how to create threads, Thread class in java, Thread priorities, Thread Synchronization. Introduction to Generics, Advantages of Generics, Generic class, Type Parameters, Generic Methods.

**UNIT-IV**

**Collections:** Overview of Java Collection Framework, Collection Interfaces – Collection, Set, List, Map, Collection classes – Array List, Linked List, Hash Set, Tree Set, Hash Map, Tree Map, Iteration over Collections – Iterator and List Iterator, Comparable and Comparator interface, Introduction to Java 8 Features, Lambda Expressions, Functional Interfaces.

**UNIT-V**

**Java I/O and NIO:** Input Stream, Output Stream, Reader, Writer, File Reader, File Writer, Bufferedreader, Bufferedwriter, Object Serialization and Deserialization, Java NIO: Non-blocking I/O, Path, Files, Selectors, Channels, Buffers, Asynchronous I/O, NIO vs. IO

**Text Books:**


1. Herbert Schildt, "Java: The Complete Reference", 12th Edition, Tata McGraw Hill Publications, 2020.
2. K Somasundaram "Advanced Programming in Java2" Jaico Publishing House, 2008.


**Suggested Reading:**


1. E Balaguruswamy "Programming with Java", TataMcGraw-Hill, 6th Edition, 2019.
2. Paul Deitel and Harvey Deitel "Java How to Program, Early Objects ", 11th Edition., 2018.


**Web Resources:**

1. https://www.cse.iitb.ac.in/~nlp-ai/javalect_august2004.html
2. https://nptel.ac.in/courses/106106147/2

**22ECC38**

## ANALOG AND DIGITAL ELECTRONICS

| | |
|---|---|
| Instruction | 3L Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 60 Marks |
| CIE | 40 Marks |
| Credits | 3 |

Prerequisite: Knowledge of Electronic device concepts.

**Course Objectives:**
This course aims to:

1. Learn basic concepts and working principles of analog devices.
2. Learn various techniques for logic minimization.
3. Comprehend the concepts of various combinational circuits and sequential circuits.

**Course Outcomes:**
Upon completion of this course, students will be able to:

1. Understand the basic concepts related to analog devices and digital circuits.
2. Design the combinational and sequential circuits.
3. Examine the behavior of logic gates.
4. Analyze the behavior of the digital system design.
5. Evaluate the performance of real time combinational circuits and sequential circuits.

**CO-PO & PSO ARTICULATION MATRIX**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 3 | 2 | 1 | 1 | 3 | 1 | 1 | 2 | 1 | 2 | 2 | 1 | 1 | 1 |
| CO 2 | 2 | 3 | 3 | 3 | 3 | - | 1 | 3 | 1 | 1 | 2 | 1 | 1 | 1 |
| CO 3 | 1 | 2 | 1 | 1 | 1 | - | 1 | - | - | 2 | 2 | 1 | 1 | 1 |
| CO 4 | 1 | 3 | 1 | 2 | 1 | - | 1 | 1 | 1 | 2 | 2 | 1 | 1 | 1 |
| CO 5 | 2 | 1 | 2 | 1 | 2 | - | 1 | 1 | - | 2 | 2 | 1 | 1 | 1 |

**UNIT -I**
**Devices:** Concepts of semiconductors, V-I Characteristics of P-N Junction diode, current equation. Characteristics of Zener Diodes, Special diodes: LED, Photo Diode.
**Applications:** Zener Diode as a voltage regulator, Half Wave Rectifier and Full Wave Rectifier.

**UNIT-II**
**Bipolar Junction Transistors:** Classification, Operation of Bipolar Junction Transistor, Configurations: CB, CE Characteristics, Applications.
**Field Effect Transistor:** Junction Field Effect Transistor: Principle of Operation, Characteristics of JFET, parameters and Operation of MOSFET

**UNIT-III**
**Boolean Algebra and Logic Simplification:** Number system representation and conversion, Binary Arithmetic, Complements, Review of Boolean Algebra and De Morgan's Theorem, SOP & POS forms, Canonical forms, Introduction to all Logic Gates, Minimization of Switching Functions: Karnaugh map method, Logic function realization: AND-OR, OR-AND and NAND/NOR realizations.

**UNIT-IV**
**Introduction to Combinational Design:** Binary Adders, Subtractors, Code converters Binary to Gray, Gray to Binary, BCD to excess3, BCD to Seven Segment display, Decoders, Encoders, Priority Encoders, Multiplexers, De-multiplexers.

**UNIT-V**
**Sequential Logic Design:** Latches, Flipflops, Difference between latch and flipflop, types of flipflops like S-R, D, T JK and Master-Slave JK FF, flipflop conversions, Ripple and Synchronous counters, Shift registers.

**Text Books:**


1. Robert L.Boylestad, Louis Nashelsky, “Electronic Devices and Circuits Theory”, Pearson Education, 9th Edition, LPE, Reprinted, 2006.
2. Morris Mano M. and Michael D.Ciletti, “Digital Design, With an Introduction to Verilog HDL”, 5th Edition, Pearson 2013.


**Suggested Reading:**


1. R.P. Jain, “Modern digital Electronics”, Tata McGraw Hill, 4th Edition, 2009.
2. Thomas L. Floyd, “Digital Fundamentals”, Pearson, 11th Edition, 2015.

**22CAC02N**

## AGILE SOFTWARE DEVELOPMENT LAB

| | |
|---|---|
| Instruction | 3P Hours per week |
| Duration of SEE | 3 Hours |
| SEE | 50 Marks |
| CIE | 50 Marks |
| Credits | 1.5 |

**Pre-requisites:** Data Structures, Programming Languages

**Course Objectives:**

This course aims to:

1. To understand project requirements by writing user stories, creating diagrams, and estimating effort.
2. To apply software design techniques and derive test cases from project blueprints.
3. To implement agile practices like functional testing, TDD, and CI for better code quality and delivery.

**Course Outcomes:**

Upon completion of this course, students will be able to:

1. Write clear and concise user stories and design essential UML artifacts such as use case and class diagrams.
2. Estimate project effort and plan development tasks effectively.
3. Derive and execute test cases for both web and mobile applications.
4. Apply agile practices like TDD and CI for improved software quality and delivery.
5. Refactor legacy code to enhance maintainability without altering existing functionality.

**CO-PO & PSO ARTICULATION MATRIX**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 3 | 2 | 3 | 1 | 2 | - | - | - | 2 | 3 | 1 | 3 | - | 2 |
| CO 2 | 2 | 2 | 2 | 1 | 1 | - | - | - | 2 | 2 | 3 | 3 | 2 | 2 |
| CO 3 | 3 | 3 | 3 | 3 | 3 | - | - | - | 1 | 2 | 2 | 3 | - | 2 |
| CO 4 | 2 | 3 | 3 | 2 | 3 | - | - | - | 2 | 3 | 2 | 3 | 2 | 2 |
| CO 5 | 3 | 2 | 2 | 2 | 3 | - | - | 2 | 1 | 2 | 2 | 3 | - | 2 |

**Lab Experiments:**

1. Write user stories for a given project.
2. Design a use case diagram for a given project.
3. Design a class diagram for a given project.
4. Design CRC cards for given classes of project.
5. Estimate the effort required for completing a given project.
6. Derive test cases for a given project through its blueprint.
7. Perform functional testing on a web application.
8. Derive test cases and test a simple mobile application.
9. Practice test-driven development (TDD) to write tests before implementing features.
10. Set up continuous integration (CI) and automate testing in an agile environment.
11. Refactor legacy code to improve readability and maintainability without altering its functionality.

**Text Books:**

1. Robert C. Martin, Agile Software Development, Principles, Patterns, and Practices “,First Edition, Pearson Education Limited 2014 .
2. Roger S. Pressman,”Software Engineering: A Practitioner’s Approach”, Seventh Edition, Roger S. Pressman,

**Suggested Reading:**

1**.** Martin Fowler,” UML Distilled: A Brief Guide to the Standard Object Modeling Language", Third Edition , Addison-Wesley

**22CSC58**

## DESIGN AND ANALYSIS OF ALGORITHMS LAB

| | |
|---|---|
| Instruction | 3 P Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 50 Marks |
| CIE | 50 Marks |
| Credits | 1.5 |

**Pre-requisites:** Programming and Problem Solving, Basics of Data structures and algorithms lab and Object Oriented Programming.

**Course Objectives:**

This course aims to:

1. Design and construct simple programs by using the different design strategies for solving different problems.
2. To enhance programming skills while improving their practical knowledge in implementing the algorithms.
3. To strengthen the practical ability and to apply suitable algorithmic approaches for solving real time problems.

**Course Outcomes:**

Upon completion of this course, students will be able to:

1. Apply Divide and Conquer to solve problems like Minimum-Maximum.
2. Develop solutions for optimization problems like Fractional Knapsack, Job Scheduling using Greedy algorithms.
3. Develop and Analyse solutions of 0/1 Knapsack and LCS using Dynamic Programming.
4. Design and Implement solutions using Backtracking for N-Queens, Graph Colouring problems.
5. Implement Topological Sorting and Dijkstra's Algorithm for graph problems.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 2 | 3 | 2 | - | - | - | - | - | - | - | - | 3 | - | 2 |
| CO 2 | 2 | 3 | 2 | - | - | - | - | - | - | - | - | 3 | 1 | 2 |
| CO 3 | 2 | 3 | 2 | - | - | - | - | - | - | - | - | 3 | 1 | 2 |
| CO 4 | 2 | 3 | 2 | - | - | - | - | - | - | - | - | 3 | - | 2 |
| CO 5 | 2 | 3 | 2 | - | - | - | - | - | - | - | - | 3 | 1 | 2 |

**List of Experiments:**

1. Implement the Minimum-Maximum Problem using the Divide and Conquer approach.
2. Implement the Fractional Knapsack Problem using the Greedy approach.
3. Implement Job Scheduling with Deadlines using the Greedy approach.
4. Implement the 0/1 Knapsack Problem using Dynamic Programming.
5. Implement the Longest Common Subsequence (LCS) using Dynamic Programming.
6. Implement the N-Queens Problem using Backtracking.
7. Implement the Graph Coloring Problem using Backtracking.
8. Implement Topological Sorting on directed acyclic graphs (DAGs).
9. Implement Dijkstra's Algorithm for finding the shortest path in a graph.
10. Implement bi-connected components
11. Case Studies on dynamic programming, backtracking, branch and bound, breadth first search

**Text Books:**


1. Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, "Introduction to Algorithms", MIT Press/McGraw-Hill, 4th Edition, 2022.
2. Michael T Goodrich and Roberto Tamassia, "Algorithm Design: Foundations, Analysis", and Internet Examples, Wiley Second Edition.

**22ITC03N**

# JAVA PROGRAMMING LAB

(Common to CSE, IT, AI&DS , CET and allied branches)

| | |
|---|---|
| Instruction | 3P Hours per week |
| Duration of SEE | 3 Hours |
| SEE | 50 Marks |
| CIE | 50 Marks |
| Credits | 1.5 |

**Course Objectives:**

**This course aims to:**

1. Introduce the core principles of Object-Oriented Programming (OOP).
2. Explain the object-oriented approach to designing and implementing classes and objects.
3. Demonstrate the use of inheritance and polymorphism in Java.
4. Illustrate exception handling and multithreading techniques for managing runtime behaviour.
5. Explore Java's IO package for developing basic input/output functionalities in applications.

**Course Outcomes:**

Upon successful completion of this course, student will be able to:

1. Apply OOP principles to design and develop Java applications.
2. Implement inheritance and interfaces to build modular and reusable code.
3. Use exception handling and multithreading to manage multiple execution paths efficiently.
4. Develop robust applications utilizing the Java Collection Framework.
5. Integrate Java IO concepts for effective data input and output operations in applications.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 2 | 2 | 3 | 3 | 2 | 2 | - | 1 | 2 | 1 | 2 | 2 | 2 | 2 |
| CO 2 | 3 | 3 | 3 | 2 | 2 | 1 | - | 1 | 2 | 2 | 2 | 2 | 2 | 2 |
| CO 3 | 2 | 2 | 3 | 2 | 2 | 1 | - | 1 | 2 | 1 | 2 | 2 | 2 | 2 |
| CO 4 | 2 | 2 | 3 | 2 | 2 | 1 | - | 1 | 2 | 1 | 2 | 2 | 2 | 2 |
| CO 5 | 3 | 3 | 3 | 2 | 3 | 1 | - | 1 | 2 | 2 | 2 | 2 | 3 | 3 |

**LIST OF EXPERIMENTS**

1. Implement the program(s) to handle the various data types, operators, expressions, control-flow, and strings.
2. Develop a java program(s) for constructors.
3. Develop a java program to demonstrate the method overloading and riding.
4. Develop a java program(s) to deal with different types of inheritances and interfaces.
5. Implement the program(s) to demonstrate the packages.
6. Develop a java program(s) to handle user defined exceptions with multiple catch blocks.
7. Implement program(s) to demonstrate Multithreading and thread synchronization.
8. Implement program(s) to demonstrate generics.
9. Implement the collection framework classes with Iterator/List Iterator.
10. Develop a java program(s) to implement the features of JDK8.

**Text Books:**


1. Herbert Schildt, "Java: The Complete Reference", 12th Edition, Tata McGraw Hill Publications, 2020.
2. K Somasundaram "Advanced Programming in Java2" Jaico Publishing House, 2008.


Suggested Reading:


1. E Balaguruswamy "Programming with Java", TataMcGraw-Hill, 6th Edition, 2019.
2. Paul Deitel and Harvey Deitel "Java How to Program, Early Objects ", 11th Edition., 2018.


Web Resources:

1. https://www.cse.iitb.ac.in/~nlp-ai/javalect_august2004.html
2. https://nptel.ac.in/courses/106106147/2

**22CAI01**

## MOOCs / Training / Internship-1

| | |
|---|---|
| Instruction / Demonstration / Training | 3-4 Weeks / 90 Hours |
| Duration of Semester End Presentation | - |
| Semester End Evaluation | - |
| Continuous Internal Evaluation | 50 Marks |
| Credits | 2 |

**Prerequisite :** Knowledge of Basics Sciences and Engineering Science

**COURSE OBJECTIVES :**

This course aims to:

1. Expose students to industrial environments, contemporary technologies, and real-world work conditions.
2. Enhance students' technical and managerial skills through practical learning opportunities and interaction with industry professionals and society.
3. Develop awareness of current technological trends, engineering responsibilities, and professional ethics relevant to their domain.

**COURSE OUTCOMES:** After the completion of this course, the student will be able to

1. Learn new technologies and solve real time projects.
2. Expose to the industrial environment problems and technologies
3. Gain knowledge on contemporary technologies industrial requirements.
4. Identify , Design and Develop solutions for real world problems
5. Communicate their ideas and learning experiences through reports and presentation.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 3 | 3 | 2 | 3 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| CO 2 | 2 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 2 |
| CO 3 | 3 | 2 | 1 | 2 | 2 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 2 | 2 |
| CO 4 | 1 | 3 | 3 | 3 | 1 | 2 | 1 | 2 | 3 | 2 | 2 | 2 | 2 | 2 |
| CO 5 | 1 | 1 | 1 | 1 | 1 | 1 | - | 2 | 3 | 2 | 1 | 2 | 2 | 2 |

**PROCESS TO BE FOLLOWED FOR CARRYING OUT INSTRUCTIONS TO STUDENTS:**

1. Students may apply for internships through the AICTE Portal or through CDC of the institute by filling the application form IAP-101.
2. Industry shall scrutinize the students based on their criteria and communicate a provisional offer or confirmation letter to the student.
3. If students apply through CDC, then CDC shall nominate the students for various opportunities accordingly by issuing NOC (IAP-104).
4. The respective head of the department shall assign a faculty mentor.
5. Student shall undergo internship/industrial training at the concerned Industry/Organization by submitting the form, IAP-103.
6. During the internship, Faculty Mentor will evaluate the performance of students twice either by visiting the Industry/Organization or through obtaining periodic reports from students.
7. Student shall submit internship report to the industry/organization at the end of internship program.
8. On successful completion of the Internship, Industry/Organization shall issue Internship Certificate to the students
9. All the students should maintain discipline, professional ethics and follow the health and safety precautions during internship

Student shall maintain diary during the internship and submit the internship report at the end of the internship. The report will be evaluated by the supervisor on the basis of the following criteria:

- Originality
- Adequacy and purposeful write-up
- Organization, format, drawings, sketches, style, language etc.
- Variety and relevance of learning experience
- Practical applications, relationships with basic theory and concepts taught in the course

**Evaluation of Internship:** The internship of the students will be evaluated in three stages:

1. Evaluation by the Industry ( in the scale of 1 to 10 where 1-Unsatisfactory; 10-Excellent)
2. Evaluation by faculty Mentor on the basis of site visit(s) or periodic communication (15 marks)
3. Evaluation through seminar presentation/Viva-Voce at the Institute(This can be reflected through marks assigned by Faculty Mentor (25 marks)

# R22-A

## CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY (AUTONOMOUS)

**AICTE Model Curriculum R 22-A with effect from AY 2025-26**

B.E. CSE (Artificial Intelligence and Machine Learning)

**SEMESTER – IV**

| S. No | Course Code | Category | Title of the Course | Scheme of Instruction | | | Scheme of Examination | | | Credits |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Hours per Week | | | Duration of SEE in Hours | Maximum Marks | | |
| | | | | L | T | P/D | | CIE | SEE | |
| **THEORY** | | | | | | | | | | |
| 1 | 22CAC03N | PCC | Principles of Artificial Intelligence | 3 | - | - | 3 | 40 | 60 | 3 |
| 2 | 22CAC04N | PCC | Introduction to Machine Learning | 3 | 1 | - | 3 | 40 | 60 | 4 |
| 3 | 22CAC05N | PCC | Computer Architecture and Microprocessor | 3 | 1 | - | 3 | 40 | 60 | 4 |
| 4 | 22CSC11N | PCC | Database Management Systems | 3 | - | - | 3 | 40 | 60 | 3 |
| 5 | 22MTC13 | BS | Mathematical Foundation for Data Science and Security | 3 | - | - | 3 | 40 | 60 | 3 |
| **PRACTICAL** | | | | | | | | | | |
| 6 | 22CAC06N | PCC | Principles of Artificial Intelligence Lab | - | - | 3 | 3 | 50 | 50 | 1.5 |
| 7 | 22CAC07N | PCC | Introduction to Machine Learning Lab | - | - | 2 | 3 | 50 | 50 | 1 |
| 8 | 22CSC13N | PCC | Database Management Systems Lab | - | - | 3 | 3 | 50 | 50 | 1.5 |
| 9 | 22CAU01 | | Upskill Certification Course - I | - | | | - | 25 | - | 0.5 |
| **TOTAL** | | | | **15** | **2** | **8** | **-** | **375** | **450** | **21.5** |

**L: Lecture** **T: Tutorial** **P: Practical**

**CIE - Continuous Internal Evaluation** **SEE - Semester End Examination**

**22CAC03N**

**PRINCIPLES OF ARTIFICIAL INTELLIGENCE**

| | |
|---|---|
| Instruction | 3L Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 60 Marks |
| CIE | 40 Marks |
| Credits | 3 |

**Pre-requisites:** Data structures, Discrete Mathematics, Probability Theory.

**COURSE OBJECTIVES:** This course aims to

1. To list the significance of AI.
2. To discuss the various components that are involved in solving an AI problem.
3. To analyze the various knowledge representation schemes, reasoning and learning techniques of AI.

**COURSE OUTCOMES:** On Successful completion of the course, students will be able to

1. Explain the role of agents and interaction with the environment to establish goals.
2. Identify and formulate search strategies to solve problems by applying suitable search strategies.
3. Compare and contrast the various knowledge representation schemes of AI.
4. Appraise probabilistic reasoning and model building
5. Apply Markov decision Process to solve real world problems.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 3 | 2 | - | - | - | 1 | - | - | - | - | 2 | 2 | 1 | 2 |
| CO 2 | 3 | 3 | 2 | 2 | 2 | - | - | - | - | - | 2 | 3 | 2 | 3 |
| CO 3 | 3 | 2 | 2 | 2 | 2 | - | - | - | - | - | 3 | 2 | 2 | 3 |
| CO 4 | 3 | 2 | 2 | 3 | 2 | - | - | - | - | - | 3 | 2 | 3 | 3 |
| CO 5 | 3 | 3 | 3 | 2 | 2 | 1 | - | - | - | - | 3 | 3 | 3 | 3 |

**UNIT - I**

Introduction: **Foundations of AI, History, State of the Art, Risks and Benefits.**
**Intelligent agents:** Agents and Environment, The Concept of Rationality, Structure of an Agent.
**Solving problems by Search-** Problem-Solving Agents, State space representation, Search graph and Search tree Searching for Solutions.

**UNIT - II**

**Uninformed Search Strategies**: Uniform cost search, Iterative deepening Depth-first search, Bidirectional search**.**
**Informed (Heuristic) Search Strategies:** Heuristic Functions, Hill- climbing, Greedy best-first search, A* search, Simulated Annealing search.

**UNIT – III**

**Adversarial Search:** Game Theory, Alpha–Beta Pruning, Constraint Satisfaction Problems.
**Logic Concepts and Logic Programming:** Introduction, Propositional Calculus Propositional Logic, Natural Deduction System, Axiomatic System, Semantic Tableau, Predicate Logic, Resolution Refutation in Propositional Logic and Predicate Logic

**UNIT - IV**

**Knowledge Representation:** Introduction, approaches to knowledge Representation, Knowledge Representation using Semantic Network, Knowledge Representation using Frames.
**Probabilistic Reasoning:** Probability, inference using full joint distributions, Bayes rule, Bayesian networks-representation, construction, exact and approximate inference, temporal model, hidden Markov model.

**UNIT – V**

**Markov Decision process:** MDP formulation, utility theory, multi attribute utility functions, decision networks, sequential decision problems value iteration, policy iteration partially observable MDP.

**Text Books:**


1. Russell, Norvig, “Artificial Intelligence: A Modern Approach”, Pearson Education, 4th Edition, 2020.
2. Saroj Kaushik, “Artificial Intelligence”, Cengage Learning India, First Edition, 2011.


**Suggested Reading:**


1. Rich, Knight, Nair, “Artificial Intelligence”, Tata McGraw Hill, 3rd Edition 2009.
2. Trivedi. M.C., “A classical approach to Artificial Intelligence”, Khanna Publishing House, Delhi.


**Online Resources:**

1. http://nptel.ac.in/courses/106106126/
2. http://nptel.ac.in/courses/106105077/

**22CAC04N**

## INTRODUCTION TO MACHINE LEARNING

| | |
|---|---|
| Instruction | 3L+1P Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 60 Marks |
| CIE | 40 Marks |
| Credits | 4 |

**Pre-requisites:** Linear Algebra and Probability theory basics, Artificial Intelligence.

**Course Objectives:** The objectives of this course are to

1. Understand the need for Machine Learning
2. Study various machine learning techniques and its applications
3. Design solutions for real world problems using machine learning techniques.

**Course Outcomes:** On Successful completion of the course, students will be able to

1. Define the basic concepts related to Machine Learning.
2. Recognize the underlying mathematical relationships across ML algorithms and their paradigms.
3. Determine the various applications of Machine Learning.
4. Model, design and develop solutions to real world problems using Machine Learning Algorithms.
5. Evaluate and interpret the results of the various machine learning tools.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 3 | 2 | - | - | 2 | - | - | - | - | 2 | - | 3 | - | 2 |
| CO 2 | 3 | 3 | 2 | 2 | 2 | - | - | - | - | 1 | - | 3 | 3 | 2 |
| CO 3 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | - | - | 1 | - | 3 | 3 | 2 |
| CO 4 | 3 | 3 | 3 | 3 | 3 | - | - | - | 2 | 2 | 2 | 3 | 3 | 2 |
| CO 5 | 2 | 3 | 2 | 3 | 3 | - | - | - | 1 | 2 | 2 | 3 | 3 | 2 |

**UNIT-I**

**Introduction to Machine Learning:** Introduction to ML, Types of Machine Learning, Data formats, Applications of Machine Learning.

**Concept Learning:** Concept Learning Task and General to Specific Ordering of Hypothesis.

**Data Pre-processing Techniques:**

Loading the data, Missing values, Outlier, Managing categorical data, k- Fold Cross-validation and Data splitting with training and test sets.

**UNIT-II**

**Feature Engineering:**

Data scaling and normalization, whitening, Feature selection and filtering, Visualization of high-dimensional Datasets and PCA.

**Regression Algorithms:** Linear Regression, Logistic regression, Lasso and Ridge regression, Error measures with Regression.

**UNIT-III**

**Classification Algorithms:** Decision Trees**,** k-Nearest Neighbors (k-NN), Bayes theorem, Naïve Bayes classifiers, Discriminant analysis, Support Vector Machines, Grid search, Confusion Matrix with metrics.

**UNIT–IV**

**Ensemble Learning:** Introduction to Ensemble Learning with Random Forests, AdaBoost, XGBoost, Light GBM and Voting classifier.

**Clustering Algorithms:** Basics of Clustering, K-means, Hierarchical Clustering, DBSCAN, Evaluation methods of Clustering.

**UNIT–V**

**Machine Learning Architectures:** Scikit-learn tools for machine learning architectures with Pipeline.

**Neural Networks:**

Introduction to Neural Networks, Activation functions, multi-layer perceptron, Deep learning architectures with convolutional neural networks, recurrent neural networks.

**Text Books:**


1. Giuseppe Bonaccorso, “Machine Learning Algorithms”, 2nd Edition, Packt, 2018.
2. Tom Mitchel “Machine Learning”, Tata McGraW Hill, 2017.


**Suggested Reading:**


1. Laurence Moroney,” AI and Machine Learning for Coders”, O'Reilly Media,October 2020.
2. Jeeva Jose, Introduction to Machine Learning, Khanna Book Publishing Company, 2020.
3. M N Murty and Ananthanarayana V S, “Machine Learning: Theory and Practice”, Universities Press; First Edition,2024.
4. Kevin Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.


**Online Resources:**

1. Introduction to Machine Learning: https://www.youtube.com/watch?v=r4sgKrRL2Ys&list=PL1xHD4vteKYVpaIiy295pg6_SY5qznc77 and https://onlinecourses.nptel.ac.in/noc23_cs18/preview
2. Introduction to Machine Learning: https://www.youtube.com/watch?v=T3PsRW6wZSY&list=PLIg1dOXc_acbdJo-AE5RXpIM_rvwrerwR and https://onlinecourses.nptel.ac.in/noc22_cs97/preview

**22CAC05N**

## COMPUTER ARCHITECTURE AND MICROPROCESSOR

| | |
|---|---|
| Instruction | 3 L Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 60 Marks |
| CIE | 40 Marks |
| Credits | 3 |

Pre-requisites: Analog and digital circuits.

**Course Objectives:** The objectives of this course are

1. To understand the basic principles of Instruction-Level Architecture and Instruction Execution, Memory System Design.
2. To learn various I/O devices and their operations, with emphasis on Instruction Level Parallelism.
3. To gain knowledge on Micro Programming and Pipelining techniques.

**Course Outcomes:** On Successful completion of this course, student will be able to

1. Understand the functional block diagram of single bus architecture of a computer, and perform basic arithmetic operations.
2. Comprehend the architecture and functioning of the 8086 microprocessor.
3. Design assembly language programs using 8086 instruction set.
4. Analyze memory transfer operations and improve performance using pipelining techniques.
5. Interpret the working of memory systems and architectures of large computer systems.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 2 | 1 | - | 1 | - | - | - | - | - | - | 1 | - | - | 1 |
| CO2 | 2 | 1 | 1 | 1 | 2 | - | - | - | - | - | 1 | - | - | 2 |
| CO3 | 1 | 2 | 3 | - | 2 | - | - | - | 1 | 1 | - | - | - | 2 |
| CO4 | 1 | 2 | 2 | 1 | 2 | - | - | 2 | 1 | - | - | 1 | - | 2 |
| CO5 | 2 | 2 | 1 | - | 1 | 1 | - | - | - | - | 1 | - | - | 2 |

**UNIT - I**

Basic Structure of Computers: Computer Types, Functional Units, Basic Operational Concepts, Bus Structures, Software, Performance, Multiprocessors and Multi-computers.

Arithmetic: Addition and Subtraction of Signed numbers, Design of Carry Look-ahead Adder, Multiplication of positive numbers, Signed-Operand Multiplication, Integer Division.

**UNIT - II**

Basic Processing Unit: Fundamental concepts, Execution of a complete instruction, Multiple-Bus organization, Hardwired control, Micro programmed control.

8086 Architecture: CPU Architecture, Internal operation, Machine language instructions, 8086 Addressing modes.

**UNIT- III**

Assembly Language Programming: Instruction format, Instruction execution timing. Data transfer instructions, Arithmetic instructions.

Branch instructions, Loop instructions, NOP and HLT, Flag manipulation instructions, Logical instructions, Shift and Rotate instructions, Directives and Operators. Procedures, Interrupts and Interrupt routines, Macros .

**UNIT - IV**

Peripheral devices and their characteristics: Input-output subsystems, I/O device interface, I/O transfers– Program Controlled, Interrupt Driven and DMA, privileged and non-privileged instructions, Software Interrupts and Exceptions. Programs and processes – role of interrupts in process state transitions.

Pipelining: Basic concepts, Data hazards, Instruction hazards, Structural hazard. Influence on instruction sets, Data path and control considerations.

**UNIT – V**

The Memory System: Memory hierarchy, Semiconductor RAM Memories, Cache Memories, Performance considerations, Virtual Memories, Memory Management requirements, Secondary Storage.

Large Computer Systems: Forms of Parallel Processing, Array Processors, Structure of general-purpose

multiprocessors, Program Parallelism and Shared Variables.

**Text Books:**


1. Carl Hamacher, ZvonkoVranesic, SafwatZaky, “Computer Organization”,5thEdition, McGraw Hill Education Edition2011.
2. Yu-chengLiu, Glenn A.Gibson, “Microcomputer Systems: The 8086/ 8088 Family”, 2nd Edition, PHI Learning 2011.


**Suggested Reading:**


1. M. M. Mano, “Computer System Architecture”, 3rdedition, Prentice Hall,1994.
2. William Stallings, “Computer Organisation and Architecture, Design for Performance”, Pearson, 9th Edition, 2013.
3. Douglas Hall. “Microprocessor and Interfacing programming and Hardware”, Tata McGraw Hill, Revised 2nd Edition, 2007.
4. Brey B. Brey, “The Intel Microprocessor, 8086/8088, 80186/80188, 80286, 80386, 80486, Pentium and Pentium Pro-Processors-Architecture, Programming and Interfacing”, 4thEdition, Prentice Hall.

**22CSC11N**

## DATABASE MANAGEMENT SYSTEMS

| | |
|---|---|
| Instruction | 3 L Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 60 Marks |
| CIE | 40 Marks |
| Credits | 3 |

**Prerequisite:** Programming and Data Structures.

**Course Objectives:**

This course aims to:

1. Familiarize students with fundamental concepts of database management. These concepts include aspects of database design, database languages and database-system implementation
2. Understand about data storage techniques and indexing.
3. Impart knowledge in transaction management, concurrency control techniques and recovery procedures.

**Course Outcomes:**

Upon completion of this course, students will be able to:

1. Understand fundamental concepts of database and design database schema for an application.
2. Write relational algebra expressions and SQL queries for various tasks.
3. Apply the principles of functional dependency and normalization to ensure data integrity
4. Understand indexing and transaction processing
5. Analyze transaction processing, concurrency control and recovery mechanisms.

**CO-PO & PSO ARTICULATION MATRIX**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 2 | 2 | 2 | 2 | 3 | - | - | - | - | - | 1 | 2 | 3 | 2 |
| CO 2 | 2 | 3 | 2 | 2 | 3 | - | - | - | - | - | 1 | 2 | 3 | 2 |
| CO 3 | 2 | 1 | 2 | 1 | 3 | - | - | - | - | - | - | 2 | 3 | 2 |
| CO 4 | 2 | 1 | 1 | - | - | - | - | - | - | - | - | 3 | 2 | 2 |
| CO 5 | 2 | 1 | - | 1 | - | - | - | - | - | - | - | 3 | 2 | 3 |

**UNIT-I**

**Introduction:** Database System Applications, Purpose of Database Systems, View of Data, Database Languages, Database Users and Administrators, Database System Architecture, Data Models, **E-R Model**: Introduction, Constraints, E-R Diagrams, E-R Design Issues, Mapping from ER to relational model, Extended E-R Features.

**UNIT-II**

**Relational Algebra:** Introduction to relational algebra operations, Basic relational algebra operators, Natural join, Assignment operator. **SQL:** Data Types, Basic Structure of SQL Queries, Modification of the Database, Set Operations, Aggregate Functions, Data-Definition Language, Integrity Constraints, Null Values, Views, Join Expression. Simple Queries (select/project/join/ aggregate queries), Complex queries (With Clause, Nested Sub queries, Views)

**UNIT-III**

**Functional Dependency**: Trivial and Nontrivial Dependencies, Closure of Set of Functional Dependencies, Attribute closure, Irreducible Set of Functional Dependencies, lossless decomposition, **Normalization**–1NF, 2NF, 3NF and BCNF, Dependency preserved decomposition, Comparison of BCNF and 3NF.

**UNIT–IV**

**Indexing:** Basic Concepts, Primary Index, Dense and Sparse Indices, Secondary Indices, Tree-Structured Indexing, Indexed Sequential Access Method (ISAM), B+Tree Index Files, Hash indices, Bitmap indices. **Transaction**

**Processing:** Concept of transactions and schedules, ACID properties, Conflict-serializability

**UNIT-V**

**Concurrency control:** Lock-Based Protocols, Dead lock handling, Timestamp-Based Protocols, Validation-Based Protocols. **Recovery system:** Failure classification, Log based recovery, recovery algorithm, ARIES.

**Text Books:**


1. Silberschatz, Korth and Sudarshan, “Database System Concepts”, 7th Edition, McGraw-Hill, 2021.
2. C.J. Date, "An Introduction to Database Systems", 8th edition, Pearson, 2020,


**Suggested Reading:**


1. Raghu Ramakrishnan, JohnnesGehrke, “Database Management Systems”, 3rd Edition, McGraw Hill, 2014.
2. Elmasri and Navathe, “Fundamentals of Database Systems”, 7th Edition, Pearson Pubs, 2017.
3. Lemahieu, Broucke and Baesens, “Principles of Database Management”, Cambridge University Press, 2018.
4. Krishnan, “Database Management Systems”, McGraw Hill.


**Online Resources:**

1. http://www.nptelvideos.in/2012/11/database-managementsystem.html.

**22MTC13**

## MATHEMATICAL FOUNDATION FOR DATA SCIENCE & SECURITY

| | |
|---|---|
| Instruction | 3 L Hours per week |
| Duration of Semester End Examination | 3 Hours |
| SEE | 60 Marks |
| CIE | 40 Marks |
| Credits | 3 |

**Course Objectives:**

1. Able to learn and Analyzing data in Linear and Non-Linear form.
2. Able to fit the hypothetical data using probability distribution.
3. To know the characteristic of various continuous probability distributions.
4. To discuss the testing of hypothesis of sample data.
5. To know the security issues of Cryptography.

**Course outcomes:** On successful completion of this course the students shall be able to

1. Analyze the coefficient of skewness and fitting of the data by various methods.
2. Apply properties of Mathematical Expectations and analyze the various distributions.
3. Evaluate areas of curves by using various distributions.
4. Apply various tests for testing the significance of sample data.
5. Apply RSA –PKC for solving security issues.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 3 | 3 | 3 | 3 | - | - | - | - | - | - | 2 | 1 | 1 | - |
| CO2 | 3 | 3 | 2 | 3 | - | - | - | - | - | - | 2 | 1 | 1 | - |
| CO3 | 3 | 3 | 2 | 3 | - | - | - | - | - | - | 2 | 1 | 1 | - |
| CO4 | 3 | 3 | 3 | 3 | - | - | - | - | - | - | 2 | 1 | 1 | - |
| CO5 | 3 | 3 | 2 | 3 | - | - | - | - | - | - | 2 | 1 | 1 | - |

**UNIT-I: Basic Statistics**

Measures of Central Tendency, Measures of Dispersion, Moments (Moments about the mean and moments about a point). Skewness, Karl Pearson's coefficient of skewness and Bowley's coefficient of skewness for frequency distribution, Kurtosis. Correlation, coefficient of correlation, limits of correlation coefficient. Linear Regression, Regression coefficients, Properties of Regression Coefficients. Curve fitting by the Method of Least Squares, Fitting of Straight lines and Exponential curve.

**UNIT-II: Mathematical Expectation and Discrete Probability Distribution**

Conditional Probability, Baye's theorem. Random variable, discrete random variable, Probability Mass Function, continuous random variable, probability density function. Mathematical expectation, properties of Expectation, properties of variance. Poisson distribution, MGF and Cumulates of the Poisson distribution, Recurrence formula for the probabilities of Poisson distribution (Fitting of Poisson distribution)

**UNIT-III: Continuous Probability Distributions**

Normal distribution, Characteristics of normal distribution and Normal probability Curve, MGF and CGF of Normal distribution, Areas under normal curve. Uniform distribution, Moment generating function, Mean and Variance of uniform distribution. Exponential distribution, MGF, CGF, Mean and Variance of Exponential distribution.

**UNIT-IV: Testing of Hypotheses**

Test of significance, null and alternative hypotheses, Errors in sampling, level of significance. Large sample test: Test of significance for single proportion, difference of proportions, single mean and difference of means. Small Sample Tests: T-Test for single mean, differences of Means. F- test for equality of two population variances. Chi-Square test of Goodness of fit.

**UNIT-V: Number Theory & CRYPTOGRAPHY (RSA – PKC)**

Division Algorithm, Greatest Common Divisor, Euclidean Algorithm, Wilson's Theorem, Euler's Phi-Function, Euler's Theorem, Some Properties of the Phi-Function. The RSA public key cryptosystem, Implementation and

security issues, Pollard's p−1 factorization algorithm, Quadratic Residues and quadratic reciprocity.

**Text books:**

1. S.C.Gupta, V.K.Kapoor, "Fundamentals of Mathematical Statistics", Sultan Chand and Sons, 2014.
2. Burton, David M. (2007) Elementary Number Theory (7thedu.). Tata McGraw Hill Edition, Indian Reprint
3. Mathematical Cryptography by Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman Springer Science+ Business Media LLC.

**Suggested Reading:**

1 .W. Feller, "An Introduction to Probability Theory and its Applications", Vol. 1, 3rd Ed., Wiley, 1968.
2. Sheldon Ross, "A First Course in Probability", 9th Edition, Pearson publications, 2014.
3. Koshy, T.Elementary Number Theory with Applications, Elsevier Publications, New Delhi, 2002.
4. G.A.Jones & J.M.Jones "Elementary Number Theory", Springer UTM, 2007.

Online Resources:

| Course Code | Course Name | Resource Links |
|---|---|---|
| 22MTC13 | Mathematical Foundation for Data Science and Security | 1. https://archive.nptel.ac.in/courses/110/107/110107114/<br>2. https://archive.nptel.ac.in/courses/111/101/111101137/ (Week 1,2,4,5 &7) |

**22CAC06N**

## PRINCIPLES OF ARTIFICIAL INTELLIGENCE LAB

| | |
|---|---|
| Instruction | 3P Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 50 Marks |
| CIE | 50 Marks |
| Credits | 1.5 |

**Pre-requisites:** Programming Basics, Probability and Statistics.

**COURSE OBJECTIVES:** This course aims to

1. To design and analyze various computing algorithms and techniques using Python.
2. To apply different learning algorithms to solve real time problems.
3. To recognize the underlying mathematical models and logics behind various AI techniques.

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

1. Understand the basic components of library environment and installations.
2. Analyze the design heuristics and apply various techniques to solve real world problems.
3. Apply variety of algorithms to solve problems.
4. Identify how to use GitHub and submit back genuine contributions.
5. Implement problems using game search algorithms.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 3 | 2 | - | - | 2 | - | - | - | - | - | 2 | 2 | 1 | 2 |
| CO2 | 3 | 3 | 3 | 3 | 3 | - | - | - | - | - | 3 | 3 | 3 | 3 |
| CO3 | 3 | 3 | 3 | 2 | 3 | - | - | - | - | - | 3 | 3 | 3 | 3 |
| CO4 | 2 | 2 | 2 | - | 3 | - | 2 | 3 | 3 | 3 | 3 | 2 | 2 | 2 |
| CO5 | 3 | 2 | 3 | 2 | 3 | - | - | 2 | - | - | 3 | 3 | 2 | 3 |

**Load all the experiments in GitHub Repository**

**Lab Experiments:**

1. Design/construct the workflow of a general AI project using draw.io
2. Implement Water Jug Problem using A* search
3. Implement an 8-puzzle solver using Heuristic search technique.
4. Implement the Constraint Satisfaction problem using backtracking.
5. Implement a program for game search.
6. Implement a Bayesian network from a given data and infer the data from that Bayesian network.
7. Implement a Hidden Markov Model for a given data.
8. Implement a MDP to run value iteration in any environment.
9. Implement a MDP to run policy iteration in any environment.
10. Build a bot to build any game using easy AI libraries

**TEXT BOOKS:**


1. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, 3rd Edition, Prentice **Hall, 2010.**
2. Saroj Kaushik, “Artificial Intelligence”, Cengage Learning India, 2011.

**22CAC07N**

# INTRODUCTION TO MACHINE LEARNING LAB

| | |
|---|---|
| Instruction | 2 P Hours per week |
| Duration of SEE | 3 Hours |
| SEE | 50 Marks |
| CIE | 50 Marks |
| Credits | 1 |

**COURSE OBJECTIVES:** This course aims to

1. To make use of Data sets in implementing the machine learning algorithms.
2. To implement the machine learning concepts and algorithms.
3. To use real world data and implement machine learning models.

**COURSE OUTCOMES:** After completion of this course, students will be able to,

1. Identify the fundamental issues and challenges of machine learning: data, model selection, model complexity, etc.
2. Identify and utilize modern tools that are useful for data analysis.
3. Recognize and implement various ways of selecting suitable model parameters for different machine learning techniques.
4. Implement and evaluate various Machine Learning approaches on real world problems.
5. Apply Keras ,Tensorflow and PyTorch to implement ML techniques and Case studies.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO1 | 3 | 3 | 2 | 2 | 2 | - | - | - | - | 1 | 1 | 3 | 2 | 2 |
| CO2 | 2 | 2 | 2 | 2 | 3 | - | - | - | 1 | 1 | 2 | 3 | 2 | 2 |
| CO3 | 2 | 3 | 3 | 3 | 3 | - | - | - | 1 | 2 | 2 | 3 | 2 | 2 |
| CO4 | 3 | 3 | 3 | 3 | 3 | - | - | - | 2 | 2 | 2 | 3 | 2 | 2 |
| CO5 | 2 | 2 | 3 | 2 | 3 | - | - | - | 2 | 2 | 2 | 3 | 2 | 2 |

**LIST OF EXPERIMENTS:**

1. Identification and Installation of python environment towards the machine learning, installing python modules/Packages Import scikitlearn, keras and tensorflows etc.
2. Implement the Pre-processing with Handling missing values, Detecting and treating outliers.
3. Implement the Data splitting into training, validation, and test sets, k-Fold Cross-validation.
4. Implement the Feature Engineering with Feature selection (filter, wrapper, embedded methods) and also Feature Scaling.
5. Implement Dimensionality Reduction Using Principal Component Analysis (PCA) and Supervised algorithms.
6. To build the Regression Algorithms with Linear Regression, Logistic Regression and Lasso Regression including Error measures: MAE, MSE, RMSE, $R^2$ Score.
7. Implement the Classification algorithms with Decision Tree, k-Nearest Neighbors (k-NN) and Support Vector Machines (SVM) with Confusion Matrix and performance metrics.
8. Implement the Ensemble Learning with Random Forest, XGBoost and Voting Classifier (Hard and Soft Voting) with performance metrics.
9. Implement the Clustering with K-Means Clustering and Hierarchical Clustering (Agglomerative & Divisive) with evaluation metrics.
10. Implement the Multi-layer Perceptron (MLP) with supervised learning classification algorithms.

**TEXT BOOKS:**

1. Aurélien Géron ,"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems", Third Edition,2019.

**ONLINE RESOURCES:**

1. http://www.cs.cmu.edu/~tom/
2. http://www.holehouse.org/mlclass/

**22CSC13N**

## DATABASE MANAGEMENT SYSTEMS LAB

| | |
|---|---|
| Instruction | 3 P Hours per Week |
| Duration of SEE | 3 Hours |
| SEE | 50 Marks |
| CIE | 50 Marks |
| Credits | 1.5 |

**Course Objectives:** This course aims to:

1. Become familiar with the concepts of structured query language.
2. Understand about Programming Language / Structured Query Language (PL/SQL).
3. Learn database constraints, DCL, TCL and advanced SQL commands.
4. Familiarize with cursors, triggers, exceptions, procedures and functions in PL/SQL.

**Course Outcomes:** Upon completion of this course, students will be able to:

1. Outline the built-in functions of SQL and Create, Alter and Drop table.
2. Demonstrate Queries to retrieve and Change Data using Select, Insert, Delete and Update. Construct Queries using Group By, Order By and Having Clauses.
3. Demonstrate Commit, Rollback, save point commands and formulate the Queries for Creating Views and constraints.
4. Develop queries using Joins, Sub-Queries.
5. Develop PL/SQL code to create stored procedures, functions, cursors and exceptions.

**CO-PO & PSO Articulation Matrix**

| PO/PSO / CO | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PSO1 | PSO2 | PSO3 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CO 1 | 3 | 2 | 2 | - | - | - | - | - | - | - | 1 | 2 | 3 | 2 |
| CO 2 | 3 | 2 | 2 | 2 | 3 | - | - | - | - | - | 1 | 3 | 3 | 2 |
| CO 3 | 3 | 1 | 2 | 1 | 3 | - | - | - | - | - | - | 3 | 3 | 2 |
| CO 4 | 3 | - | | 2 | - | - | - | - | - | - | - | 3 | 3 | 2 |
| CO 5 | 3 | 1 | 2 | 1 | - | - | - | - | - | - | - | 3 | 3 | 3 |

List of Experiments:

1. Queries using Built-In functions, like aggregate functions, String Functions, Numeric Functions, Data Functions, Conversion Functions and other miscellaneous.
2. Queries using DDL and DML statements.
3. Queries using Group By, Order By, Having Clauses and set operations.
4. Queries on Controlling Data: Commit, Rollback and Save point.
5. Queries for Creating, Dropping and Altering Tables, Views and Constraints.
6. Queries using Joins, views and Sub-Queries.
7. Write PL/SQL code using Basic Variables, bind and substitution variables.
8. Write PL/SQL code using Control Structures.
9. Write PL/SQL code using Procedures, Functions.
10. Write PL/SQL code using Cursors, Triggers and Exceptions.

Text Books:


1. “Oracle: The complete Reference”, Oracle Press.
2. Nilesh Shah, “Database Systems Using Oracle”, PHI, 2007.


Suggested Reading:


1. Rick FVander Lans, “Introduction to SQL”, 4th Edition, Pearson Education, 2007.
2. "The Language of SQL (Learning)" by Larry Rockoff.
3. Steven Feuerstein, “Oracle PL/SQL Programming”, 6th Edition, O’reilly publications, 2014.


Online Resources:

- https://onlinecourses.nptel.ac.in/noc22_cs91/preview

**22CAU01**

**UPSKILL CERTIFICATION COURSE - I (Need to get from Central Committee)**