



## Dr. B. C. Roy Engineering College, Durgapur

### Department of CSE(AIML)

Field	Details
Course Name	Introduction to Artificial Intelligence
Course Code	AM-301
Semester	3
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:0P

### 1. Prerequisites

- Basic Programming Skills (preferably Python)
- Discrete Mathematics (including logic and probability)
- Linear Algebra (basic concepts of vectors and matrices)

### 2. Course Learning Objectives

- To provide students with a foundational understanding of the core concepts and techniques in Artificial Intelligence, encompassing search algorithms, knowledge representation, probabilistic reasoning, and machine learning.
- To equip students with the ability to formulate and solve AI problems using appropriate algorithms and techniques, including both uninformed and informed search, constraint satisfaction, and machine learning models.
- To foster critical thinking and problem-solving skills through practical application of AI concepts to real-world scenarios, emphasizing the design, implementation, and evaluation of AI systems.
- To cultivate an awareness of the ethical implications and societal impact of AI technologies, promoting responsible development and deployment of AI systems.

- To develop students' ability to critically evaluate and interpret the results of AI algorithms, understanding their limitations and potential biases.

### 3. Teaching Methodology

- Lectures and Presentations
- Interactive Discussions and Case Studies
- Lab Sessions
- Guest Lectures

### 4. Evaluation System

Activities	Class Test Full Marks	Assignment Full Marks	Attendance Full Marks	Total Marks
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

### 5. Course Modules

Module	Topics	Hours
1	<b>Introduction to Artificial Intelligence</b> <ul style="list-style-type: none"> <li>- What is AI? Definitions and subfields</li> <li>- History of AI: Key milestones and paradigms</li> <li>- Intelligent Agents: PEAS model, types of agents (reflex, model-based, goal-based, utility-based)</li> <li>- Environment types: fully observable, deterministic, episodic, static, discrete, single-agent vs. multi-agent</li> <li>- Problem formulation: State space representation,</li> </ul>	10

	<p>goal definition, actions, path cost</p> <ul style="list-style-type: none"> <li>- Introduction to Search: State space search, uninformed search (BFS, DFS) - concepts only, no detailed complexity analysis at this stage</li> <li>- Applications of AI: Real-world examples across various domains (e.g., healthcare, finance, robotics)</li> <li>- Ethical considerations in AI: Bias, fairness, accountability, transparency</li> </ul>	
2	<p><b>Search Algorithms and Problem Solving</b></p> <ul style="list-style-type: none"> <li>- Uninformed Search (BFS, DFS): Focus on practical application and algorithm design. Brief mention of completeness and optimality.</li> <li>- Informed Search (A*): Heuristic functions, admissibility, consistency. Practical examples and algorithm implementation.</li> <li>- Local Search: Hill climbing, simulated annealing (conceptual overview). Focus on applications and limitations.</li> <li>- Constraint Satisfaction Problems (CSPs): Backtracking search, constraint propagation. Practical examples (e.g., map coloring).</li> <li>- Game Playing: Minimax algorithm (conceptual overview), Alpha-Beta pruning (conceptual overview).</li> </ul>	8
3	<p><b>Knowledge Representation and Reasoning</b></p> <ul style="list-style-type: none"> <li>- Propositional Logic: Syntax, semantics, inference (truth tables, resolution). Focus on practical application and limitations.</li> <li>- First-Order Logic (FOL): Syntax and semantics (brief introduction). Focus on representing knowledge using FOL. Skip detailed proof procedures (resolution, deduction).</li> <li>- Knowledge Representation: Semantic networks, frames, ontologies (brief introduction). Emphasis on practical applications in AI systems.</li> <li>- Rule-Based Systems: Forward and backward chaining. Practical examples and limitations.</li> <li>- Reasoning under Uncertainty: Introduction to probabilistic reasoning (Bayes' Theorem - intuitive understanding).</li> </ul>	7
4	<p><b>Probabilistic Reasoning and Bayesian Networks</b></p> <ul style="list-style-type: none"> <li>- Probability basics: Joint probability, conditional probability, independence.</li> <li>- Bayes' Theorem: Detailed explanation and applications in various scenarios.</li> <li>- Bayesian Networks: Representation, inference (qualitative understanding of exact and</li> </ul>	6

	<p>approximate inference). Focus on practical applications and interpretation of results.</p> <ul style="list-style-type: none"> <li>- Naive Bayes Classifier: Implementation and application to a classification problem.</li> <li>- Hidden Markov Models (HMMs): Conceptual overview and applications (e.g., speech recognition, part-of-speech tagging).</li> </ul>	
5	<p><b>Machine Learning Fundamentals</b></p> <ul style="list-style-type: none"> <li>- Supervised Learning: Regression and classification problems. Focus on practical algorithms and applications.</li> <li>- Decision Trees: ID3 algorithm, pruning, overfitting. Practical implementation and application.</li> <li>- Linear Regression: Simple linear regression, multiple linear regression. Focus on practical application and interpretation of results.</li> <li>- Logistic Regression: Application to binary classification problems. Focus on practical application and interpretation of results.</li> <li>- Model Evaluation: Metrics (accuracy, precision, recall, F1-score). Cross-validation techniques.</li> </ul>	6
6	<p><b>Introduction to Deep Learning and AI Ethics</b></p> <ul style="list-style-type: none"> <li>- Introduction to Neural Networks: Perceptron, multi-layer perceptron (MLP) – high-level overview. Focus on the architecture and functionality, not detailed mathematical derivations.</li> <li>- Deep Learning: Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) – high-level overview and applications.</li> <li>- Reinforcement Learning: Markov Decision Processes (MDPs) – conceptual overview. Focus on the basic concepts and applications.</li> <li>- AI Ethics: Bias in algorithms, fairness, accountability, transparency, societal impact of AI. Case studies and discussions.</li> </ul>	4

## 6. References

### Textbooks:

1. S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach, Pearson Education, 4th Edition, 2022

2. Elaine Rich and Kevin Knigh, Introduction to Artificial Intelligence, McGraw Hill, Third Edition, 2017.

**Reference Books:**

1. Michael Negnevitsley, Artificial Intelligence: A guide to Intelligent Systems, Addison Wesley, Third Edition, 2017.

2. G.F. Luger, and W.A. Stubblefield, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Addison-Wesley Publishing Company, 2011

**7. Course Outcomes**

ID	Statement	Action Verb	Knowledge Level
AM 304 .1	Students will be able to explain fundamental concepts of Artificial Intelligence, including its definition, history, key subfields, and ethical considerations.	Explain	Understand
AM 304 .2	Students will be able to apply uninformed and informed search algorithms (BFS, DFS, A*) to solve various problem-solving scenarios, and evaluate their performance.	Apply	Apply
AM 304 .3	Students will be able to design and implement a knowledge representation system using propositional logic or first-order logic to model a given problem domain, and perform basic reasoning tasks.	Design	Apply
AM 304 .4	Students will be able to analyze and interpret the results of probabilistic reasoning using Bayes' Theorem and Bayesian Networks in solving real-world problems, including building and applying a Naive Bayes classifier.	Analyze	Analyze
AM 304 .5	Students will be able to build and evaluate supervised machine learning models (e.g., decision trees, linear regression, logistic regression) for classification and regression tasks, using appropriate evaluation metrics and cross-validation techniques.	Build	Create
AM 304 .6	Students will be able to critically evaluate the ethical implications of AI systems, including bias, fairness, accountability, and transparency, and	Evaluate	Evaluate

	propose strategies.	potential	mitigation		
--	------------------------	-----------	------------	--	--

## 8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	1	1	1	-	1	-	1
CO2	3	3	2	2	2	1	1	1	1	1	1	1
CO3	3	3	3	2	2	1	1	1	1	1	1	1
CO4	3	3	2	3	2	1	1	1	1	1	1	1
CO5	3	2	3	3	3	1	1	1	1	1	1	1
CO6	1	1	1	1	1	3	2	3	1	2	1	1

## 9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	1	1
CO2	3	1	-
CO3	3	1	-
CO4	3	1	-
CO5	3	2	1
CO6	1	-	3



## Dr. B. C. Roy Engineering College, Durgapur

### Department of CSE(AIML)

Field	Details
Course Name	Data Structure and Algorithms
Course Code	AM-302
Semester	3
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

### 1. Prerequisites

- Introductory Programming with a focus on a procedural language (e.g., C, Python, Java)
- Discrete Mathematics (including basic set theory, logic, and graph theory)
- Basic Probability and Statistics

### 2. Course Learning Objectives

- To provide students with a solid foundation in fundamental data structures and algorithms essential for understanding and developing AI systems.
- To equip students with the ability to design, analyze, and implement efficient algorithms for solving common problems encountered in AI, including search, optimization, and machine learning.
- To enable students to apply graph theory and tree-based data structures to solve complex AI problems related to knowledge representation, pathfinding, and decision-making.
- To introduce students to advanced algorithmic techniques such as dynamic programming and heuristic search, and their applications in AI domains like natural language processing and game playing.

- To foster an understanding of core machine learning algorithms and their underlying principles, including model selection, bias-variance tradeoff, and regularization.

### 3. Teaching Methodology

- Lectures and Presentations
- Interactive Discussions and Case Studies
- Lab Sessions
- Guest Lectures

### 4. Evaluation System

Activities	Class Test Full Marks	Assignment Full Marks	Attendance Full Marks	Total Marks
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

### 5. Course Modules

Module	Topics	Hours
1	<p><b>Introduction to Data Structures and Algorithms for AI</b></p> <ul style="list-style-type: none"> <li>- Introduction to Algorithms and Data Structures in AI</li> <li>- Basic Terminology: Time and Space Complexity</li> <li>- Elementary Data Organization: Arrays, Lists</li> <li>- Data Structure Operations: Search, Insertion, Deletion</li> <li>- Complexity of Algorithms: Big O Notation</li> <li>- Arrays (Linear Arrays, Multidimensional Arrays)</li> <li>- Linked Lists (Singly, Doubly Linked Lists, Applications in AI)</li> <li>- Stacks (Applications in AI: Backtracking, Depth-First Search)</li> </ul>	10

	- Queues (Applications in AI: Breadth-First Search, Priority Queues)	
2	<b>Fundamental Algorithm Design Techniques for AI</b>  - Searching Algorithms (Linear Search, Binary Search, Applications in AI) - Sorting Algorithms (Merge Sort, QuickSort, Applications in AI: Data preprocessing) - Divide and Conquer (General Method, Binary Search, Merge Sort) - Greedy Algorithms (General Method, Applications in AI: Huffman coding, Clustering) - Graph Traversal Algorithms (Breadth-First Search, Depth-First Search, Applications in AI: Pathfinding, Social Network Analysis)	6
3	<b>Trees and Graphs for AI</b>  - Trees (Binary Trees, Binary Search Trees, Tree Traversal, Applications in AI: Decision Trees, Game Trees) - Graphs (Graph Representation, Adjacency Matrix, Adjacency List, Applications in AI: Knowledge Representation, Recommender Systems) - Minimum Spanning Trees (Prim's Algorithm, Kruskal's Algorithm, Applications in AI: Network Optimization) - Shortest Path Algorithms (Dijkstra's Algorithm, Applications in AI: Pathfinding, Navigation)	7
4	<b>Dynamic Programming and Heuristics in AI</b>  - Dynamic Programming (Introduction, Applications in AI: Sequence Alignment, NLP) - Heuristic Search Algorithms (A*, Best-First Search, Applications in AI: Pathfinding, Game Playing) - Introduction to Constraint Satisfaction Problems (CSPs) and backtracking search	5
5	<b>Introduction to Machine Learning Algorithms</b>  - Linear Regression - Logistic Regression - k-Nearest Neighbors (k-NN) - Decision Trees (ID3, CART) - Naive Bayes - Bias-Variance Tradeoff	7
6	<b>Hashing and Advanced Topics in AI</b>  - Hashing (Hash Tables, Collision Handling,	7

	Applications in AI: Data Indexing, Feature Extraction) - Introduction to Neural Networks (Perceptron, Multilayer Perceptron, Backpropagation) - Common Activation Functions (Sigmoid, ReLU) - Regularization Techniques (L1, L2)	
--	---	--

## 6. References

### Textbooks:

1. Data Structures Using C, Reema Thareja, Oxford University Press
2. Data Structures – Seymour Lipschutz –Tata McGraw-Hill -2006

### Reference Books:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Sartaj Sahni, Galgotia Publications Pvt. Ltd, New Delhi
2. Data Structures and Algorithms- Alfred V.Aho, John E.Hopcraft and Jeffrey D.Ullman - Pearson Education - Fourteenth Impression -2013

## 7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
PCC- AIML 302.1	Students will be able to define and explain fundamental data structures (arrays, linked lists, stacks, queues, trees, graphs) and their applications in AI algorithms.	Explain	Understand
PCC- AIML 302.2	Students will be able to apply basic algorithm design techniques (divide and conquer, greedy, dynamic programming) to solve problems relevant to AI, such as pathfinding and sequence alignment.	Apply	Apply
PCC- AIML 302.3	Students will be able to implement and analyze the time and space complexity of common searching and sorting algorithms (linear search, binary search, merge sort, quicksort) using Big O notation.	Implement	Apply

PCC- AIML 302.4	Students will be able to analyze and compare the performance of different graph traversal algorithms (BFS, DFS) and shortest path algorithms (Dijkstra's) in solving AI-related problems, such as social network analysis and pathfinding.	Analyze	Analyze
PCC- AIML 302.5	Students will be able to evaluate and apply basic machine learning algorithms (linear regression, logistic regression, k-NN, decision trees, Naive Bayes) to solve classification and regression problems, and explain the bias-variance tradeoff.	Evaluate	Evaluate
PCC- AIML 302.6	Students will be able to design and implement a solution using hashing techniques and explain the fundamental concepts of neural networks, including perceptrons, MLPs, backpropagation, and common activation and regularization functions.	Design	Create

## 8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	-	-	-	-	1	-	1
CO2	3	3	2	2	1	-	-	-	1	1	1	1
CO3	3	2	2	2	2	-	-	-	1	1	1	1
CO4	3	3	2	3	1	-	-	-	1	1	1	1
CO5	3	2	2	2	2	-	1	-	1	1	1	1
CO6	3	2	3	2	3	-	1	-	1	1	1	1

## 9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	1	1
CO2	3	1	1
CO3	3	1	1
CO4	3	1	1
CO5	3	2	2
CO6	3	2	2





## Dr. B. C. Roy Engineering College, Durgapur

### Department of CSE(AIML)

Field	Details
Course Name	Computer System Organization
Course Code	AM-303
Semester	3
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:0P

### 1. Prerequisites

- Discrete Mathematics
- Introduction to Computer Architecture
- Linear Algebra

### 2. Course Learning Objectives

- To provide students with a comprehensive understanding of the fundamental principles of computer systems and their crucial role in enabling efficient AI computations.
- To equip students with the knowledge to analyze and evaluate the performance of different computer architectures and hardware components in the context of AI workloads.
- To enable students to critically assess the trade-offs between various architectural choices (e.g., RISC vs. specialized architectures, different memory hierarchies) for optimizing AI applications.
- To foster students' ability to design and implement efficient solutions for handling the unique computational challenges posed by AI algorithms, including data movement and parallel processing.

- To develop students' understanding of the software and hardware ecosystem supporting large-scale AI deployments, including cloud computing platforms and distributed computing frameworks.

### 3. Teaching Methodology

- Lectures and Presentations
- Interactive Discussions and Case Studies
- Lab Sessions
- Guest Lectures

### 4. Evaluation System

Activities	Class Test Full Marks	Assignment Full Marks	Attendance Full Marks	Total Marks
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

### 5. Course Modules

Module	Topics	Hours
1	<b>Introduction to Computer Systems for AI</b> - Basic Principles of Computer Abstraction and Technology - Hardware Components relevant to AI (CPUs, GPUs, TPUs, NPUs) - Computer Types (servers, embedded systems, cloud computing)	10

	<ul style="list-style-type: none"> <li>- Functional Units relevant to AI workloads (e.g., matrix multiplication units)</li> <li>- Basic Operational Concepts</li> <li>- Von Neumann Architecture and its limitations for AI</li> <li>- Performance Measurement (FLOPS, throughput, latency)</li> </ul>	
2	<p><b>Instruction Set Architectures and AI Acceleration</b></p> <ul style="list-style-type: none"> <li>- Instruction Set Architectures (ISA) focusing on SIMD instructions</li> <li>- Types of Instructions relevant to AI (vector operations, matrix operations)</li> <li>- Addressing Modes</li> <li>- RISC vs. Specialized Architectures for AI (e.g., Tensor Processing Units)</li> <li>- ARM Architecture and its role in mobile AI</li> <li>- Instruction Cycle (Fetch-Decode-Execute) and its optimization for AI</li> <li>- Memory Access Patterns in AI workloads</li> </ul>	8
3	<p><b>Computer Arithmetic and AI-Specific Hardware</b></p> <ul style="list-style-type: none"> <li>- Computer Arithmetic for AI (focus on floating-point arithmetic and quantization)</li> <li>- Signed and Unsigned Numbers</li> <li>- Addition and Subtraction</li> <li>- Logical Operations</li> <li>- ALU Construction and its relevance to AI operations</li> <li>- Floating-Point Representation and Arithmetic (IEEE 754)</li> <li>- Parallelism in AI computations</li> </ul>	6
4	<p><b>Memory Systems and I/O for AI</b></p> <ul style="list-style-type: none"> <li>- Memory Hierarchy and its impact on AI performance</li> <li>- Cache Memory (Mapping Techniques) and its optimization for AI workloads</li> <li>- Main Memory and its bandwidth limitations</li> <li>- Virtual Memory and its role in managing large AI models</li> <li>- Input/Output (I/O) for AI (high-speed data transfer)</li> <li>- Data transfer optimization techniques for AI (DMA, high-speed interfaces)</li> <li>- Memory-mapped I/O and its use in AI accelerators</li> </ul>	7

5	<b>Pipelining, Parallel Processing, and AI Architectures</b> <ul style="list-style-type: none"> <li>- Pipelining and its limitations in AI computations</li> <li>- Parallel Processing Concepts (SIMD, MIMD)</li> <li>- GPU Architecture (Many-core Architecture, SIMT Model, Memory Hierarchy) and its application in deep learning</li> <li>- TPU Architecture (Systolic Arrays, Matrix Multiplication Unit)</li> <li>- NPU Architectures (On-chip AI Accelerators)</li> </ul>	6
6	<b>Multiprocessors and Distributed Systems for AI</b> <ul style="list-style-type: none"> <li>- Multicore Processors and their role in AI</li> <li>- Distributed Computing for AI (model parallelism, data parallelism)</li> <li>- Interconnection Networks and their impact on AI performance</li> <li>- Cloud Computing Platforms for AI (AWS, Google Cloud, Azure)</li> <li>- Software frameworks for distributed AI (TensorFlow, PyTorch)</li> </ul>	5

## 6. References

### Textbooks:

1. D. A. Patterson and J. L. Hennessy, Computer Organisation and Design: The Hardware/ Software Interface, Fourth Edition, Morgan Kaufman, 2009
2. Carl Hamacher, "Computer Organization", 5th Edition, McGraw Hill Publishers, 2002

### Reference Books:

1. William Stallings, "Computer Organization and Architecture Designing for Performance", 8th Edition, Pearson Education, 2010.
2. John P Hayes, "Computer Architecture and Organization", 3rd revised Ed., McGraw-Hill, 1998

## 7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
PCC-AIML 303.1	Students will be able to explain the fundamental components of computer systems relevant to AI,	Explain	Understand

	including CPUs, GPUs, TPUs, and NPUs, and describe their roles in AI workloads.		
PCC-AIML 303.2	Students will be able to apply their knowledge of instruction set architectures and addressing modes to analyze the efficiency of different instruction types for AI-specific operations, such as vector and matrix manipulations.	Apply	Apply
PCC-AIML 303.3	Students will be able to analyze the impact of computer arithmetic, including floating-point representation and quantization, on the accuracy and performance of AI algorithms.	Analyze	Analyze
PCC-AIML 303.4	Students will be able to evaluate the performance of different memory systems and I/O techniques in the context of AI workloads, proposing optimizations for data transfer and cache utilization.	Evaluate	Analyze
PCC-AIML 303.5	Students will be able to design and compare different parallel processing architectures (SIMD, MIMD, GPU, TPU, NPU) for specific AI tasks, considering factors such as performance, energy efficiency, and scalability.	Design	Create
PCC-AIML 303.6	Students will be able to synthesize their knowledge to create a plan for deploying a machine learning model on a chosen distributed computing platform (AWS, Google Cloud, Azure), considering factors such as model parallelism, data parallelism, and the selection of appropriate software frameworks (TensorFlow, PyTorch).	Synthesize	Create

## 8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	1	1	1	-	-	-	-	1	-	1
CO2	3	2	1	1	1	-	-	-	-	1	-	1
CO3	3	2	1	2	1	-	-	-	-	1	-	1

CO4	3	2	2	2	2	-	1	-	-	1	1	1
CO5	3	2	3	2	2	1	1	1	1	2	1	1
CO6	3	2	3	2	3	1	1	1	2	2	2	1

## 9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	1	1
CO2	3	1	1
CO3	3	1	1
CO4	3	1	1
CO5	3	2	1
CO6	3	3	2



## Dr. B. C. Roy Engineering College, Durgapur

### Department of CSE(AIML)

Field	Details
Course Name	Introduction to Object Oriented Programming
Course Code	AM-304
Semester	3
Course Category	Program Core Courses
Credits	3
Hours per Week	3L:0T:4P

### 1. Prerequisites

- High School Algebra
- Basic understanding of computer architecture and operating systems
- Familiarity with problem-solving and logical reasoning

### 2. Course Learning Objectives

- To provide students with a solid foundation in object-oriented programming principles and their practical application using Java, equipping them with the skills to design, implement, and debug robust software.
- To introduce students to fundamental data structures and algorithms, emphasizing their relevance to artificial intelligence applications and fostering an understanding of algorithm efficiency.
- To enable students to develop basic graphical user interfaces and handle user input, enhancing their ability to create interactive applications.
- To equip students with the knowledge and skills to manage files, handle multithreading, and interact with databases, crucial for developing data-intensive AI applications.

- To offer a foundational understanding of key AI concepts and introduce the C++ programming language, focusing on aspects relevant to performance and memory management in AI systems.

### 3. Teaching Methodology

- Lectures and Presentations
- Interactive Discussions and Case Studies
- Lab Sessions
- Guest Lectures

### 4. Evaluation System

Activities	Class Test Full Marks	Assignment Full Marks	Attendance Full Marks	Total Marks
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

### 5. Course Modules

Module	Topics	Hours
1	<b>Introduction to Programming and Java Fundamentals</b> - Object-Oriented Programming (OOP) paradigm: Need, core concepts (classes, objects, methods), Abstraction, Encapsulation - Introduction to Java: History, features relevant to AI (e.g., platform independence) - Data types, Variables (scope and lifetime), Operators,	10

	<p>Expressions, Control statements (if-else, loops), Type conversion</p> <ul style="list-style-type: none"> <li>- Simple Java program structure, Comments</li> <li>- Arrays and ArrayLists (introducing dynamic arrays)</li> <li>- Java programming environment (IDE setup, compilation, execution)</li> <li>- Basic input/output (console I/O)</li> </ul>	
2	<p><b>Object-Oriented Programming in Java: Core Concepts</b></p> <ul style="list-style-type: none"> <li>- Classes and Objects: Constructors, methods, access modifiers (public, private, protected)</li> <li>- `this` keyword, Method overloading, Inheritance (single, multilevel), Polymorphism (method overriding)</li> <li>- Exception Handling: `try-catch-finally`, common exceptions (e.g., `NullPointerException`, `ArrayIndexOutOfBoundsException`)</li> <li>- Introduction to Recursion (with examples relevant to AI algorithms like tree traversal)</li> <li>- String manipulation (essential for data processing in AI)</li> <li>- Introduction to Collections: ArrayList, understanding iterators</li> </ul>	10
3	<p><b>Advanced OOP and Data Structures</b></p> <ul style="list-style-type: none"> <li>- Abstract classes and interfaces (polymorphism, design patterns)</li> <li>- Packages and imports</li> <li>- Generics (introducing type safety and reusability)</li> <li>- Common Data Structures: Linked Lists (singly, doubly), Stacks, Queues (implementations and use cases relevant to AI, e.g., search algorithms)</li> <li>- Introduction to algorithm analysis (Big O notation – basic understanding)</li> </ul>	6
4	<p><b>GUI Programming and Event Handling (Introduction)</b></p> <ul style="list-style-type: none"> <li>- Introduction to GUI programming with JavaFX (simpler than AWT/Swing): basic window creation, adding controls (buttons, labels, text fields)</li> <li>- Event handling: Event listeners, handling user input (mouse clicks, keyboard events)</li> <li>- Simple layout management (using JavaFX layout panes)</li> </ul>	6
5	<p><b>Multithreading and File I/O</b></p> <ul style="list-style-type: none"> <li>- Multithreading: Creating and managing threads, thread synchronization (basic concepts, avoiding race conditions)</li> <li>- File I/O: Reading and writing data to files (essential for</li> </ul>	5

	data loading in AI applications) - Introduction to Databases and SQL (basic SELECT, INSERT, UPDATE, DELETE queries – crucial for AI data management)	
6	<b>Introduction to AI Concepts and C++ Fundamentals</b>  - Introduction to key AI concepts: Search algorithms (BFS, DFS – conceptual overview), knowledge representation (basic concepts), machine learning (very high-level introduction) - Introduction to C++: Basic syntax, data types, control structures, functions (focus on aspects relevant to performance in AI applications) - Pointers (brief introduction, emphasizing memory management)	5

## 6. References

### Textbooks:

1. Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011
2. Kathy Sierra and Bert Bates, Head First Java, O'Reilly Media

### Reference Books:

1. Nageswararao R., Core Java: An Integrated Approach, Dreamtech Press, 2008
2. Y. Daniel Liang, Introduction to Java Programming, 7/e, Pearson, 2013.

## 7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
PCC- AIML 301 .1	Students will be able to write Java programs that utilize fundamental programming constructs, including data types, operators, control structures, and arrays, to solve basic computational problems.	Write	Apply
PCC- AIML 301 .2	Students will be able to design and implement Java programs using object-oriented principles,	Design and Implement	Apply

	including classes, objects, inheritance, polymorphism, and exception handling, demonstrating understanding of encapsulation and abstraction.		
PCC- AIML 301 .3	Students will be able to analyze the efficiency of algorithms using Big O notation and implement common data structures such as linked lists, stacks, and queues in Java, explaining their use cases in AI algorithms like search.	Analyze and Implement	Analyze
PCC- AIML 301 .4	Students will be able to develop simple graphical user interfaces (GUIs) using JavaFX, handling user input events and employing basic layout management techniques.	Develop	Apply
PCC- AIML 301 .5	Students will be able to create and manage multithreaded Java applications, perform file I/O operations, and execute basic SQL queries to manage data, demonstrating an understanding of their relevance to AI data processing.	Create and Manage	Apply
PCC- AIML 301 .6	Students will be able to compare and contrast fundamental AI concepts such as search algorithms (BFS, DFS) and knowledge representation, and explain the basic syntax and data structures of C++, emphasizing memory management using pointers, in the context of AI application performance.	Compare, Contrast, and Explain	Understand

## 8. CO-PO Mapping

<b>CO</b>	<b>PO1</b>	<b>PO2</b>	<b>PO3</b>	<b>PO4</b>	<b>PO5</b>	<b>PO6</b>	<b>PO7</b>	<b>PO8</b>	<b>PO9</b>	<b>PO10</b>	<b>PO11</b>	<b>PO12</b>
CO1	3	2	3	1	3	1	1	1	1	1	1	1
CO2	3	2	3	1	3	1	1	1	1	1	1	1
CO3	3	3	2	3	3	1	1	1	1	2	1	1
CO4	2	1	3	1	3	1	1	1	1	1	1	1
CO5	3	2	3	2	3	1	1	1	1	1	2	1
CO6	2	2	1	1	2	1	1	1	1	2	1	1

## 9. CO-PSO Mapping

<b>CO</b>	<b>PSO1</b>	<b>PSO2</b>	<b>PSO3</b>
CO1	2	1	-
CO2	2	1	-
CO3	3	1	-
CO4	1	1	-
CO5	2	1	-
CO6	3	1	1



## Dr. B. C. Roy Engineering College, Durgapur

### Department of CSE(AIML)

Field	Details
Course Name	Discrete Mathematics
Course Code	AM-305
Semester	3
Course Category	Basic Science Courses
Credits	3
Hours per Week	3L:0T:0P

### 1. Prerequisites

- Introductory Discrete Mathematics (sets, functions, basic proof techniques)
- Fundamental programming proficiency and algorithmic problem-solving skills (e.g., Python or Java)
- Basic familiarity with algebraic concepts such as modular arithmetic and elementary group properties

### 2. Course Learning Objectives

- Cultivate a deep conceptual understanding of discrete mathematical foundations--including logic, algebraic structures, combinatorics, and probability--and their relevance to core computer-science problems and AI techniques.
- Enable students to formulate, analyze, and prove statements using rigorous proof methods (direct, contrapositive, induction, contradiction) across a variety of discrete structures and algorithmic contexts.
- Develop the ability to model, represent, and manipulate discrete structures such as relations, groups, and graphs, and to apply appropriate algorithmic strategies (e.g., traversal, shortest-path, spanning-tree, matching) to solve computational problems.

- Equip learners with analytical tools for evaluating algorithmic performance, including recurrence solving, generating-function techniques, and asymptotic analysis, and to interpret these results in terms of efficiency and scalability.
- Foster interdisciplinary thinking by connecting discrete mathematics concepts to practical applications in computer science domains such as cryptography, hashing, machine learning, and AI knowledge representation.

### 3. Teaching Methodology

- Lectures and Presentations
- Interactive Discussions and Case Studies
- Lab Sessions
- Guest Lectures

### 4. Evaluation System

Activities	Class Test Full Marks	Assignment Full Marks	Attendance Full Marks	Total Marks
CIA-1	25	10	05	40
CIA-2	25	10	05	40
End Semester Examination (ESE)	-	-	-	60
Total				100 Marks

### 5. Course Modules

Module	Topics	Hours
1	<b>Foundations of Mathematical Logic and Proof Techniques</b> - Propositional logic: syntax and semantics	5

	<ul style="list-style-type: none"> <li>- Logical connectives and truth tables</li> <li>- Normal forms (CNF &amp; DNF)</li> <li>- Predicate logic: quantifiers, free and bound variables</li> <li>- Rules of inference and proof strategies (direct, contrapositive, induction, contradiction)</li> <li>- Equivalence, implication, and consistency</li> <li>- Applications of logic to AI knowledge representation</li> </ul>	
2	<b>Algebraic Structures and Binary Relations</b> <ul style="list-style-type: none"> <li>- Properties of binary relations (reflexive, symmetric, transitive, antisymmetric)</li> <li>- Equivalence relations and partitioning</li> <li>- Partial orders and posets</li> <li>- Introduction to modular arithmetic (useful for hashing)</li> <li>- Basic algebraic structures: semigroups, monoids, groups (finite examples)</li> <li>- Simple group properties relevant to cryptography and coding</li> </ul>	6
3	<b>Enumerative Combinatorics and Counting Methods</b> <ul style="list-style-type: none"> <li>- Counting principles: sum rule and product rule</li> <li>- Permutations and combinations</li> <li>- Binomial coefficients and the Binomial Theorem</li> <li>- Inclusion-exclusion principle</li> <li>- Pigeonhole principle and typical applications</li> <li>- Combinatorial arguments used in algorithm analysis</li> <li>- Basic counting for probability models</li> </ul>	7
4	<b>Recurrence Relations and Generating Functions</b> <ul style="list-style-type: none"> <li>- Linear recurrence relations and solving by substitution</li> <li>- Homogeneous recurrences with characteristic-root method</li> <li>- Master theorem for divide-and-conquer recurrences</li> <li>- Introduction to ordinary generating functions</li> <li>- Extracting coefficients from generating functions</li> <li>- Using generating functions to solve simple recurrences</li> <li>- Asymptotic analysis (Big-O, <math>\Theta</math>, <math>\Omega</math>) for algorithm runtime</li> </ul>	8
5	<b>Discrete Probability and Expectation</b> <ul style="list-style-type: none"> <li>- Axiomatic definition of probability</li> </ul>	8

	<ul style="list-style-type: none"> <li>- Discrete random variables and common distributions (Bernoulli, Binomial, Geometric, Poisson)</li> <li>- Joint, marginal, and conditional distributions</li> <li>- Bayes' theorem and its role in AI inference</li> <li>- Expectation, linearity of expectation, and variance</li> <li>- Applications of expectation in algorithm analysis</li> <li>- Basic probabilistic reasoning for machine-learning models</li> </ul>	
6	<p><b>Graph Theory: Structures, Algorithms, and Applications</b></p> <ul style="list-style-type: none"> <li>- Fundamental graph concepts and terminology</li> <li>- Matrix representations (adjacency and incidence matrices)</li> <li>- Walks, paths, cycles, and connectivity</li> <li>- Eulerian and Hamiltonian graph criteria</li> <li>- Directed graphs, bipartite graphs, and basic properties</li> <li>- Graph traversal algorithms: BFS, DFS, and their use in AI search</li> <li>- Shortest-path algorithms: Dijkstra and A* (overview)</li> <li>- Minimum spanning trees (Kruskal's &amp; Prim's algorithms)</li> <li>- Vertex coloring, chromatic number, and greedy coloring</li> <li>- Matchings in bipartite graphs, Hall's theorem, König's theorem</li> <li>- Overview of greedy and approximation approaches for classic graph problems</li> </ul>	8

## 6. References

### Textbooks:

1. Rosen, Kenneth H.: Discrete Mathematics and its Applications with Combinatorics and Graph Theory (7th Edition), TMH (Tata McGraw-Hill).
2. Mott, Joe L., Kandel, Abraham, and Baker, Theodore P.: Discrete Mathematics for Computer Scientists and Mathematicians, Pearson Education.
3. Johnsonbaugh, Richard: Discrete Mathematics, Pearson Education.
4. Chandrasekaran, N., and Umavathi, M.: Discrete Mathematics, PHI Learning.

### Reference Books:

1. Tremblay, J.P., and Manohar, R.: Discrete Mathematical Structures with Applications to Computer Science, TMH (Tata McGraw-Hill).
2. Mott, Joe L., Kandel, Abraham, and Baker, Theodore P.: Discrete Mathematics for Computer Scientists and Mathematicians (2nd Edition), Pearson Education.
3. Johnsonbaugh, Richard: Discrete Mathematics (7th Edition), Pearson Education.

## 7. Course Outcomes

ID	Statement	Action Verb	Knowledge Level
AM-305.1	Recall and correctly state the syntax and semantics of propositional and predicate logic, including truth tables, normal forms (CNF/DNF), and the use of quantifiers.	Recall	Remember
AM-305.2	Apply rules of inference and proof strategies (direct, contrapositive, contradiction, induction) to construct valid proofs for logical propositions and simple algorithmic claims.	Apply	Apply
AM-305.3	Solve linear recurrence relations and use ordinary generating functions to obtain closed-form expressions, and employ these techniques to determine the time-complexity of divide-and-conquer algorithms.	Solve	Apply
AM-305.4	Analyze discrete probability distributions, compute expectations, variances, and conditional probabilities, and use Bayes' theorem to perform inference in basic machine-learning scenarios.	Analyze	Analyze
AM-305.5	Evaluate the correctness and efficiency of graph algorithms (BFS, DFS, Dijkstra, A*, Kruskal's and Prim's) on given problem instances, and justify the selection of the most suitable algorithm for AI search or optimization tasks.	Evaluate	Evaluate

AM-305.6	Design and implement an end-to-end AI/ML prototype (e.g., a knowledge-base reasoner, a probabilistic classifier, or a graph-based path planner) that integrates logical reasoning, combinatorial analysis, and algorithmic techniques covered in the course.	Design	Create
----------	--	--------	--------

## 8. CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	1	1	1	1	1	1	1	2
CO2	3	3	2	2	1	1	1	1	1	2	1	2
CO3	3	3	2	2	2	1	1	1	1	2	1	2
CO4	3	3	2	3	3	2	2	2	2	2	1	3
CO5	3	3	3	2	3	2	2	2	2	3	2	2
CO6	3	3	3	3	3	3	2	3	3	3	3	3

## 9. CO-PSO Mapping

CO	PSO1	PSO2	PSO3
CO1	3	2	1
CO2	3	2	1
CO3	3	2	1
CO4	3	3	1
CO5	3	2	1
CO6	3	3	2