Course Name: Object Oriented Programming
Course Code: DS-401
(Semester- IV)
Course Broad Category: Program Core (PC)

……

1. **Course Prerequisite:**
   Concept of Python Programming, C programming

2. **Course Learning Objectives:**

- To introduce the principles of Object-Oriented Programming (OOP) and their implementation in Java and python
- To enable students to design and develop robust, reusable, and maintainable software solutions.
- To provide hands-on experience in solving real-world problems using Java and familiarize students with standard libraries and tools available in Java and python.

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology –Lectures and Presentations, Learning Management System(LMS),Interactive Discussions and Case Studies, Guest Lectures.

4. **Evaluation System –**
   A. Mid-Term Exam (40 Marks)- Formative Assessment (CIA-1) [Continuous Assessment 2 (CIA-2)]
   B. Internal Assessment (40 Marks)- Formative Assessment [Continuous Assessment 2 (CIA-2)]
   C. End-Semester Exam (60 Marks)- Summative Assessment.

5. **Course Content:**
   Course Name: Object Oriented Programming
   Course Code: DS-401
   Hours per Week: 3L:0T:0P
   Credits: 3

Module	**Topic**	**45L**
1	**Introduction to Java and OOP Concepts:** History and evolution of Java, Overview of Object-Oriented Programming: Principles of OOP (Encapsulation, Abstraction, Inheritance, Polymorphism), Features of Java: JVM, JDK, JRE, platform	8L

	independence, Basics of Java: Data types, operators, control structures, and arrays.	
2	**Classes, Objects, and Methods** Defining classes and creating objects, Constructors, Access modifiers and static members, Method overloading and overriding, Object class and its methods	6L
3	**Inheritance and Interfaces and Polymorphism** Single and multilevel inheritance, Method overriding and the use of super keyword, Abstract classes and interfaces, Multiple inheritance using interfaces, Role of final and abstract, polymorphism	7L
4	**Multithreading and Concurrency and Exception Handling** Thread lifecycle and Thread class, Synchronization and inter-thread communication, Multithreading concepts in Java, Executors and thread pools, Types of exceptions and their hierarchy, Try, catch, throw, throws, and finally blocks, Custom exceptions, Importance of exception handling in robust software development	10L
5	**Java Collections Framework and File Handling** Introduction to Collections: Lists, Sets, and Maps, Iterators and enhanced for loops, Generics in Java, Collections utility class, File I/O using streams (byte and character streams), Serialization and deserialization, Reading and writing files	4L
6	**GUI Development with JavaFX/Swing** Introduction to JavaFX/Swing, Event-driven programming, Layout managers, Creating simple GUI applications.	5L
7	**OOPs Concepts in Python** Class, Objects, Polymorphism, Encapsulation, Inheritance, Data Abstraction all oops concept implemented by Python	5L
	Total	**45L**

5. References:

Text Book:

- The Complete Reference, Java 2 (Fourth Edition), Herbert Schild, TMH.
- Object Oriented Programming with C++ by Balagurusamy
- Python Programming: A Modern Approach, Vamsi Kurama, Pearson

Reference Books:

1 Introduction to Java Programming (Comprehensive Version), Daniel Liang,

Seventh Edition, Pearson.
2 Programming in Java, Sachin Malhotra & Saurabh Chaudhary, Oxford University Press.
3 Murach's Beginning Java 2, Doug Lowe, Joel Murach and Andrea Steelman, SPD.
4 Core Python Programming, W.Chun, Pearson
5 Core Java Volume-I Fundamentals, Eight Edition, Horstmann & Cornell, Pearson Education.
6 Java Programming, D. S. Malik, Cengage Learning.

6. Course Outcomes (CO):

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
DS-401.CO1	To differentiate object-oriented programming and procedural programming	Recognize	Level 1-Remember
DS-401.CO2	To construct classes, functions and objects	Illustrate	Level 1-Remember
DS-401.CO3	To implement OOP principles like constructors and inheritance, encapsulation and polymorphism.	Implement	Level 3-Apply
DS-401.CO4	Show competence in the use of the Java programming language in the development of small to medium-sized application programs that demonstrate professionally acceptable coding and performance standard	Determine	Level 4-Evaluate
DS-401.CO5	Demonstrate an introductory understanding of graphical user interfaces, multi- threaded programming, and event-driven programming	Construct	Level 6-Create

7. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5
1	3	-	-	-	-
2	-	3	-	-	-
3	-	-	3	-	3
4	-	3	2	-	-
5	-	-	-	3	-
6	-	-	2	-	2

8. Mapping of the Course outcomes to Program Outcomes (PO)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	1	2	2	1	1	-	-	-	-	-	-	1
CO2	1	2	3	1	1	-	-	-	-	-	-	1
CO3	1	2	1	1	1	-	-	-	-	-	-	1
CO4	1	2	2	1	1	-	-	-	-	-	-	1
CO5	1	2	2	2	3	-	-	-	-	-	-	1

9. Mapping to Program Specific Outcome (PSO)

	PSO1	PSO2	PSO3	PSO4
CO1	1	2	3	2
CO2	2	1	2	2
CO3	2	2	2	3
CO4	2	3	2	2
CO5	2	2	3	2

**** End of Syllabus****

REFERENCES:

NIT Durgapur/CSE/Object Oriented Programming using C++ Syllabus

IIEST Shibpur/CSE/ Object Oriented System Design Syllabus

NIT Rourkela/CSE/ Object Oriented Programming using JAVA Syllabus

DR. B. C. ROY ENGINEERING COLLEGE
DURGAPUR

Course Name: Operating System

Course Code: DS-402
(Semester- IV)
Course Broad Category: Ability Enhancement Courses

……

1. **Course Prerequisite:** Computer Organization and Architecture, Introduction to Computing, Data Structure and Algorithm**.**
2. **Course Learning Objectives:**
   This course introduces the evolution and core principles and techniques required in the design and implementation of operating systems. This course focus on the various concepts of Operating System, including process, thread and resource management and how process communicate to each other through IPC. It also covers various process scheduling [FCFS, RR, SJF etc.] and process synchronization algorithms as well as various deadlock handling algorithms, Students will learn various memory management techniques [paging, segmentation] and various file types and how to manage the file structures. Students will be able to understand various Input output resources and how operating system manages the I/O subsystem.
3. **Teaching methodology and evaluation system for the course:**

Teaching methodology –Lectures and Presentations, Learning Management System (LMS), Interactive Discussions and Case Studies, Industry Lectures.

Evaluation System –

A. Mid-Term Exam (40 Marks)- Formative Assessment [Continuous Assessment 2(CIA-1)]
B. Internal Assessment (20 Marks)- Formative Assessment [Continuous Assessment 2 (CIA-2)]
C. End-Semester Exam (60 Marks)- Summative Assessment.

4. **Course Content:**
5. **Course Name:** Operating System
   Course Code: DS-402
   Hours per Week: 3L:0T:0P
   Credits: 3

Module	Topics	45L
1.	**Introduction to Operating System:** Evolution of different types of operating system. Batch, iterative, time sharing, multiprocessor, distributed, cluster and real time system. Unix system introduction.	2L
2.	**Operating System Structure:** Computer system structure Introduction to the fundamental concept of operating system architecture. Network, Storage, I/O structure. Dual mode [User and Kernel] operations. System components. Operating system services, System calls, System program. System structure. Virtual machines. System design and implementation and System generation.	6L
3.	**Process and Threads:** Process concept. Process control block [PCB]. Process Scheduling. Operation on Processes. Cooperating Processes, Inter process communication. Communication in client server system. Multi-threaded models. Threading issues. PThread basic concept.	5L
4.	**CPU Scheduling:** Scheduling Criteria, Scheduling Algorithms, Multi-Processor Scheduling, Real Time Scheduling, Algorithm Evaluation, Process Scheduling Models.	6L
5.	**Process Synchronization:** Synchronization Background. The critical section problem. Synchronization Hardware, Semaphores. Classic problems of synchronization. Critical regions. Monitors and OS Synchronization.	6L
6.	**Deadlocks:** System Model, Deadlock characterization. Methods of handling deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock detection and recovery from deadlock.	6L
7.	**Memory Management:** Memory Management Background, Swapping, Contiguous Memory Allocation, Paging, Segmentation with Paging, Virtual Memory, Demand Paging, Process Creation, Page Replacement, Allocation of Frames, Thrashing, Operating-System Examples, Other Considerations.	6L
8.	**File Systems:** File Concept, Access Methods, Directory Structure, Filesystem Mounting, File Sharing, Protection File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management, Efficiency and Performance, Recovery, Log Structured File System, NFS.	5L
9.	**I/O Systems:** Hardware, Application I/O Interface, Kernel I/O Subsystem, Transforming I/O to Hardware Operations, STREAMS, Performance, Disk Structure, Disk	3L

	Scheduling, Disk Management, Swap-Space Management, RAID Structure, Disk Attachment, Stable-Storage Implementation, Tertiary-Storage Structure.	

Text Book:

- Operating System Principles-Abraham Silberschatz, Peter B. Galvin, Greg Gagne 10th Edition, John Wiley.
- Operating Systems -Internal and Design Principles, Stallings, 9th Edition-2018, Pearson education/PHI.

Reference Books:

- Modern Operating Systems, Andrew S Tanenbaum 5th edition Pearson/PHI

6. Course Outcomes (CO):

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
DS-402.CO1	Explain the functional architecture of operating system.	Explain	Level 1- Remember
DS-402.CO2	Design the process control algorithms, solution to deadlock and multithreaded application.	Design	Level 3- Apply
DS-402.CO3	Applying various synchronization tools to control and data access in shared memory IPC	Apply	Level 3-Apply
DS-402.CO4	Understand both the virtual and physical memory organization and management.	Understand	Level 2- Understand
DS-402.CO5	Implantation of standard FAT & Unix file system.	Implement	Level 3-Apply
DS-402.CO6	Identify the working principles of various I/O subsystem in the operating system	Identify	Level 2- Understand

7. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5	CO6
1	1	-	-	-	-	-
2	3	2	-	1-	-	-
3	-	3	2	-	1	-
4	-	2	-	-	-	-
5	-	-	3	2	-	-
6	-	3	1	1	-	-
7	-	-	-	3	-	-
8	-	-	-	-	3	-
9	-	-	-	-	-	3

8. Mapping of the Course outcomes to Program Outcomes (PO)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	3	-	-	3	-	2	2	1	-	1	2	1
CO2	3	3	1	3	3	1	-	-	-	1	-	1
CO3	2	2	2	1	1	-	1	1	-	-	-	1
CO4	1	3	2	3	3	1	-	-	-	1	3	1
CO5	1	2	2	3	1	-	-	-	-	-	-	1
CO6	-	-	3	-	3	3	2	-	1	2	2	1

9. Mapping to Program Specific Outcome (PSO)

	PSO1	PSO2	PSO3	PSO4
CO1	**1**	**2**	**3**	**2**
CO2	**2**	**2**	**2**	**3**
CO3	**2**	**2**	**2**	**3**
CO4	**2**	**3**	**2**	**2**
CO5	**2**	**2**	**3**	**2**
CO6	**1**	**2**	**2**	**3**

***** End of Syllabus*****

References

NIT Warangal /CSE/Operating Systems/Syllabus

NIT Durgapur/CSE/Operating Systems/Syllabus

Course Name: Artificial Intelligence
Course Code: DS-403
(Semester– IV)
Course Broad Category: Program Core (PC)

……

1. Course Prerequisite:

- Basic knowledge of computers and general mathematical operations.
- Data Structures and Algorithms

2. Course Learning Objectives:

i. To introduce the fundamental concepts of Artificial Intelligence (AI) and its real-world applications.
ii. To develop and understand the problem-solving methods using AI techniques, including search strategies and knowledge representation.
iii. To explore machine learning techniques, including supervised and unsupervised learning, and their applications.
iv. To analyze logical reasoning, inference mechanisms, and probabilistic models in AI.
v. To implement AI algorithms for problem-solving in business, engineering, and decision-making domains.
vi. To discuss ethical considerations, limitations, and challenges in AI applications.

3. Teaching methodology and evaluation system for the course:

Teaching methodology –Lectures and Presentations, Learning Management System (LMS), Interactive Discussions and Case Studies, Industry Lecture.

Evaluation System –

A. Mid-Term Exam (40 Marks) - Summative Assessment (CIA-1)
B. Internal Assessment (40 Marks)- Formative Continuous Assessment [Continuous Assessment 1 (CIA-2)]
C. End-Semester Exam (60 Marks)- Summative Assessment.

4. Course Content:

Course Name: Artificial Intelligence
Course Code: DS-403
Hours per Week: 3L: 0T: 0P
Credits: 3

Module	Topics	Lectures (45L)
1	**Introduction to Artificial Intelligence:** Definition, history, and scope of AI, Applications of AI in various domains, Intelligent agents and their properties.	6L
2	**Problem-Solving and Search Techniques:** State-space representation and problem-solving methods, Uninformed search techniques: BFS, DFS, Iterative Deepening Search, Heuristic search techniques: Best-First Search, A* Algorithm, Hill Climbing, Adversarial search: Minimax Algorithm, Alpha-Beta Pruning, Constraint Satisfaction Problems (CSPs)	8L
3	**Knowledge Representation and Reasoning:** Logical representation: Propositional and Predicate Logic, Rule-based systems and inference mechanisms, Semantic networks, frames, and ontologies, Probabilistic reasoning: Bayesian networks, Markov models, Fuzzy logic for uncertainty handling	7L
4	**Machine Learning for AI**: Overview of machine learning: Supervised, Unsupervised, Reinforcement Learning, Supervised Learning: Regression, Classification (Decision Trees, Support vector machine (SVM), Neural Networks), Unsupervised Learning: Clustering (K-Means, DBSCAN), Dimensionality Reduction (PCA), Reinforcement Learning: Q-Learning, Policy Gradient Methods, Basics of Neural Networks and Deep Learning	10L
5	**AI Applications in Real-World Scenarios:** AI in healthcare, finance, robotics, and business intelligence, AI in cybersecurity and fraud detection, Case studies in AI for industrial applications, Introduction to Natural Language Processing (NLP), AI ethics and bias issues	8L
6	**Advanced Topics and Future Trends in AI:** Explainable AI and AI interpretability, AI in IoT and Edge Computing, AI Governance, Regulations, and Societal Impact, Quantum Computing and AI, Future research directions in AI.	6L

5. References:

Textbooks:

1. Deepak Khemani, *A First Course in Artificial Intelligence*, McGraw Hill.
2. Lavika Goel, *Artificial Intelligence: Concepts and Applications*, Wiley.

Reference Books:

1. Elaine Rich, Kevin Knight and Shavashankar B. Nair, *Artificial Intelligence*, McGraw Hill.
2. Tom Mitchell, *Machine Learning*, McGraw Hill.
3. Melanie Mitchell, Artificial Intelligence,: A Guide for Thinking, Pelican.

4. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Pearson.

6. Course Outcomes (CO):

After going through this course the Students will be able to:

Course Outcomes	Details	Action Verb	Knowledge Level
DS-403.CO1	Understanding AI fundamentals and applications	Define, Interpret, Summarise	Level 2-Understand, Level 1-Remember
DS-403.CO2	Apply search algorithms for problem-solving	Develop, Construct, Demonstrate	Level 2-Understand, Level 3-Apply
DS-403.CO3	Implement knowledge representation and reasoning techniques	Analyze	Level 4-Analyze
DS-403.CO4	Apply AI models using machine learning.	Develop	Level 3-Apply
DS-403.CO5	Apply AI solutions in business and industry	Build	Level 3-Apply
DS-403.CO6	Discuss ethical implications in AI	Interpret, Define	Level 2-Understand, Level 1-Remember

7. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5	CO6
1		3	2	2	2	1
2		2	3	-	-	-
3		-	-	3	-	2
4		-	-	-	3	-
5		-	1	-	-	3
6		-	-	-	-	-

8. Mapping of the Course outcomes to Program Outcomes (PO)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	3	2	1	-	-	-	-	-	-	-	-	1
CO2	2	2	2	2	-	-	-	-	-	-	-	1
CO3	2	1	3	2	2	-	-	-	-	-	-	2
CO4	-	1	2	1	2	-	-	-	-	-	-	2
CO5	1	-	1	1	3	2	-	-	2	-	-	2
CO6	3	2	1	1	3	2	-	-	-	-	-	3

9. Mapping to Program Specific Outcome (PSO)

	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	1
CO2	2	2	2	2
CO3	2	1	2	2
CO4	1	2	2	2
CO5	2	2	2	2
CO6	2	2	2	1

*** End of the Syllabus ***

Course Name: Big Data Technology
Course Code: DS-404
(Semester- IV)
Course Broad Category: Program Core(PC)

……

1. Course Prerequisite: Knowledge of Database Management System, Programming languages, SQL and Data Warehouse.

2. Course Learning Objectives:

i. To understand the Big Data Platform and its Use cases
ii. To provide an overview of Apache Hadoop
iii. To provide HDFS Concepts and Interfacing with HDFS
iv. To understand NoSQL database.

3.Teaching methodology and evaluation system for the course:

Teaching methodology – Lectures and Presentations, Learning Management System (LMS), Interactive Discussions and Case Studies, Industry Lectures.

Evaluation System –

A. Mid-Term Exam (40 Marks)- Formative Assessment [Continuous Assessment 1 (CIA-1)]
B. Internal Assessment (40 Marks)- Formative Assessment [Continuous Assessment 2 (CIA-2)]
C. End-Semester Exam (60 Marks)- Summative Assessment.

4. Course Content:

Course Name: Big Data Technology
Course Code: DS-404
Hours per Week: 3L: 0T: 0P
Credits: 3

Module	**Topics**	**45L**
1.	**Introduction to Big Data:** Big Data - Characteristics of Big Data - Big data management architecture - Examining Big Data Types - Big Data Technology Components - Big data analytics - Big data analytics examples - Web Data Overview - Web Data in Action.	10L
2.	**Hadoop Introduction:** History of Hadoop - Hadoop Ecosystem - Analyzing data with Hadoop - Hadoop Distributed File System - Design - HDFS concepts - Hadoop file system - Data flow - Hadoop	10L

Module	Topics	45L
	I / O - Data integrity - Serialization - Setting up a Hadoop cluster - Cluster specification - cluster setup and installation - YARN.	
3.	**MapReduce Introduction**: Understanding MapReduce functions - Scaling out - Anatomy of a MapReduce Job Run - Failures - Shuffle and sort - MapReduce types and formats - features - counters - sorting - MapReduce Applications –Configuring and setting the environment - Unit test with MR unit - local test	8L
4.	**Modelling Streaming Data:** Data stream and data model versus data format, Use cases of stream processing, Data streaming systems - Data harvesting, Data processing, Data analytics; Importance and implications of streaming data, streaming data solutions, Exploring streaming sensor data, Analyzing the streaming data. .	7L
5.	**NoSQL Databases**: Introduction to NoSQL - MongoDB: Introduction - Data types - Creating - Updating and deleting documents - Querying - Introduction to indexing - Capped collections - Hbase: Concepts - Hbase Vs RDBMS - Creating records - Accessing data - Updating and deleting data - Modifying data - exporting and importing data. Transaction Management – Isolation Levels and Isolation Strategies, BASE Theorem, CAP Theorem.	10L

5. Text Books:

1. EMC Education Services, "Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data", Wiley Publishers, 2015.
2. Simon Walkowiak, "Big Data Analytics with R", PackT Publishers, 2016.
3. David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, No SQL, and Graph", Morgan Kaufmann/Elsevier Publishers, 2013.
4. Bart Baesens, "Analytics in a Big Data World: The Essential Guide to Data Science and its Applications", Wiley Publishers, 2015.
5. Big Data Principles and Paradigms, Rajkumar Buyya; Rodrigo N Calheiros; Amir Vahid Dastjerdi, Elsevier/Morgan Kaufmann, Cambridge, MA.
6. Hands-On Big Data Modelling, James Lee, Tao Wei, Suresh Kumar Mukhiya, Packt Publishing. ISBN: 9781788620901.

6.COURSE OUTCOMES (COs)

On completion of the course students will be able to

Course Outcomes	Details	Action Verb	Knowledge Level
DS-404.CO1	1.Understand the necessity of Big Data Infrastructure Plan in Information System Design	Understand	Level 2-Understand
DS-404.CO2	2. Recognize different types of data elements – structural issues, characterization issues, modelling issues.	Apply	Level 3-Apply
DS-404.CO3	3. Design of Algorithms to solve Data Intensive Problems using Map Reduce Paradigm, YARN job scheduling and shuffle sort task execution.	Design	Level 6-Construct
DS-404.CO4	3.Apply techniques to handle streaming data	Apply	Level 3-Apply
DS-404.CO5	4. Analyze the non-relational database framework like, NOSQL to efficiently store and process Big Data to generate analytics.	Analyze	Level 4-Analyze

7. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5
1	3	-	-	-	2
2	2		-	-	-
3	-	-	3	-	-
4	-	2	3	-	2
5	-	-	-	3	-

8. Mapping of COs with POs and PSOs (Course Articulation Matrix):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	3	3	3	3	2	-	-	-	-	-	-	-
CO2	3	3	3	3	3	-	-	-	-	-	1	-
CO3	3	3	3	1	3	-	-	-	-	-	2	-
CO4	3	2	3	3	2	-	-	-	-	-	1	-
CO5	2	3	2	3	2	-	-	-	-	-	1	2

9.Mapping to Program Specific Outcome (PSO)

	PSO1	PSO2	PSO3	PSO4
CO1	1	2	3	2
CO2	2	2	2	3
CO3	2	2	2	3
CO4	2	3	2	2
CO5	2	2	3	2

***** End of Syllabus*****

REFERENCES:

NIT Tiruchirappalli/CSE/ Big Data Analytics Syllabus

NIT Durgapur/CSE/Big Data Modelling and Management Syllabus

Course Name: Automata and Compiler Design
Course Code: DS-405
(Semester- IV)
Course Broad Category: Program Core (PC)

……

1. **Course Prerequisite:**
   Students are expected to have basic knowledge of Set theory and some basic programming concepts.

2. **Course Learning Objectives:**
   By the end of this course, students will be able to:
   - **Understand Formal Languages and Automata Theory**
     Describe the basic concepts of formal languages, grammars, and automata. Distinguish between different types of automata: Finite Automata (DFA/NFA), Pushdown Automata, and Turing Machines.
   - **Design and Analyze Finite Automata**
     Construct deterministic and non-deterministic finite automata for regular languages. Convert regular expressions to finite automata and vice versa. Minimize deterministic finite automata.
   - **Work with Context-Free Grammars**
     Construct and analyze context-free grammars (CFGs). Convert CFGs to Chomsky Normal Form and Greibach Normal Form. Apply parsing techniques (top-down and bottom-up parsers) for syntax analysis.
   - **Understand Turing Machines and Computability**
     Explain the working of Turing Machines and their role in defining computability. Analyze the decidability and recognizability of languages.
   - **Apply Compiler Design Principles**
     Understand and implement phases of a compiler: lexical analysis, syntax analysis, semantic analysis, intermediate code generation, optimization, and code generation. Design lexical analyzers using finite automata and syntax analyzers using parsing techniques.
   - **Develop Simple Compilers or Interpreters**
     Apply theoretical concepts to design and implement basic components of a compiler. Use tools like Lex and Yacc (or similar) to generate lexical analyzers and parsers.
   - **Analyze and Optimize Code**
     Apply intermediate code generation strategies. Understand basic code optimization techniques for improving performance.

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology –Lectures and Presentations, Learning Management

System(LMS), Interactive Discussions and Case Studies, Industry Lectures.

Evaluation System –

A. Mid-Term Exam (20 Marks)- Formative Assessment [Continuous Assessment 2 (CIA-1)]
B. Internal Assessment (20 Marks)- Formative Continuous Assessment [Continuous Assessment 2 (CIA-2)]
C. End-Semester Exam (60 Marks)- Summative Assessment.

4. Course Content:

Course Name: Automata and Compiler Design
Course Code: DS-405
Hours per Week: 3L:0T:0P
Credits: 3

Module	Topics	45L
1.	**Language and Grammar:** definition, Alphabet, languages and grammars, productions and derivation, Chomsky hierarchy of languages. **Finite automata:** Definition, Characteristics, Transitional system, deterministic finite automata (DFA), Nondeterministic finite automata (NFA) , equivalence of DFA and NFA, Dead state, Finite Automata with output, Mealy machine and Moore machine, Conversio, Minimization of finite automata. Myhill-Nerode theorem, Two way finite automata, Application and limitation.	7L
2.	**Introduction to Compiler:** High level and low level language, Compiler, types, challenges in compiler design, phases and passes, language processing system, Lexical analyzer, role, issues, Token , pattern, lexeme, lexical errors, input buffering, Regular expression, regular definitions, recognition of tokens, transitional system Lex compiler. **Regular Expression:** Regular sets and regular expressions, Basic operations on regular expressions, Identities, Arden's Theorem, RE to NFA, ε-closure, NFA with ε move to DFA, Regular grammar from RE, pumping lemma for Regular expression, closure properties of regular expression, Decision problems of Regular expression, Application of RE.	8L
3.	**Context-free languages and pushdown automata:** Left and right linear grammars. Context-free grammars (CFG) and languages (CFL), parse trees, ambiguity in CFG, inherent ambiguity, Chomsky and Greibach normal forms, closure properties of CFL, pumping lemma for CFL, Application of context free grammar. Pushdown Automata(PDA), language recognized by PDA, deterministic and non deterministic PDA, equivalence of PDA and CFL, Multi stack PDA	7L
4.	**Parsing:** Top down parsing, Recursive decent parsing, Predictive parsing, Recursive predictive parsing, Non recursive predictive parsing, LL(1) parsing,	8L

	First, Follow, LL(1) table constructing, not LL(1), Error recovery in predictive parsing-panic mode and phrase level Bottom up parsing, handle, shift reduce parsing, problems/ conflict in shift reduce parsing, operator precedence parsing, LR parsing, SLR, SLR table constructing, canonical and LALR, YACC.	
5.	**Syntax directed transition:** Syntax directed definition (SDD), Attribute grammar, SDD for type checking, Abstract syntax tree, synthesized and inherited attribute, dependency graph, S and L attribute, Semantic error, Type checking, Static and dynamic check, type system, type expression, error recovery, specification of a simple type checker. **Intermediate code generation:** Advantages, forms of intermediate representation (Syntax tree, DAG, Three address code), Three address code(3AC)- types, quadruples, triple and indirect triple,3AC for relation and logical statement, Boolean expression, condition statement and loop, Backpatching. **Code generation:** Factor affecting code generation, register allocation, basic block and flow graph, transformation on basic blocks-structure preserving, common subexpression elimination, dead code elimination, renaming temporary variable, algebraic transformation, DAG and basic block, peephole optimization	8L
6.	**Context-sensitive languages:** Context-sensitive grammars (CSG) and languages, linear bounded automata and equivalence with CSG. **Turing machines:** The basic model for Turing machines (TM), Mechanical diagram Non deterministic TM. Turing-recognizable (recursively enumerable) and Turing-decidable (recursive) languages and their closure properties, concept of undecidability, reducibility, halting problem, Variation of Turing machine, Turing machine as an integer function. Recursive function, Godel number, Ackermann function, mu-recursive, lamda calculus, diagonalization, Concept of P and NP. **Run time environment:** Memory management, Storage organization, static vs dynamic allocation, activation tree, control stack, storage organization, heap allocation, activation records, garbage collection mark and sweep , reference counting, generation garbage collection, partial garbage collector.	7L

5. References:

Text Book:

- Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). Automata theory, languages, and computation. International Edition, 24, 19, TMH.
- A. V. Aho, R. Sethi and J. Ullman, Compilers: Principles, Techniques and Tools, Addison-Wesley.
- Martin, J.C (2011).Introduction to Languages and The Theory of Computation, 4th Edition, TMH.

- Mishra, Chandrasekharan, Theory of Computer Science: Automata, Languages and Computation, 3rd Edition , PHI.

Reference Books:

- Sipser, Michael. "Introduction to the Theory of Computation." ACM Sigact News,1996
- Holub – "Compiler Design in C" – PHI
- Kozen, Dexter C. Automata and computability. Springer Science & Business Media, 2012.
- David Galles, Modern Compiler Design, Pearson Education

6. Course Outcomes (CO):

On completion of the course students will be able to

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
DS-405.CO1	Explain the fundamental concepts of formal languages, automata theory, and grammar types.	Understanding	Level 2-Understand
DS-405.CO2	Design and analyze finite automata, regular expressions, and context-free grammars for language recognition.	Applying, Analyzing	Level 3-Apply, Level 4-Analyze
DS-405.CO3	Apply parsing techniques such as LL and LR parsing for syntax analysis.	Applying	Level 3-Apply
DS-405.CO4	Construct components of a compiler such as lexical analyzers, syntax analyzers, and semantic analyzers.	Applying, Creating	Level 3-Apply, Level 6-Create
DS-405.CO5	Implement code generation and optimization techniques to improve the performance of compiled code.	Applying, Evaluating	Level 3-Apply, Level 5 –Evaluate
DS-405.CO6	Demonstrate the role of automata in designing compiler components and understand their limitations.	Evaluating, Understanding	Level 2-Understand, Level 5-Evaluate

7. Mapping of course outcomes to module / course content

Module	CO1	CO2	CO3	CO4	CO5	CO6
1	3	3	-	-	-	-
2	-	3	-	-	-	-
3	-	-	-	3	-	-
4	-	-	3	2	-	-
5	-	-	-	-	3	-
6	-	-	-	-	3	3

8. Mapping of the Course outcomes to Program Outcomes (PO)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	3	3	3	3	2	1	1	1	1	1	2	2
CO2	3	2	3	3	-	1	1	1	1	1	2	2
CO3	3	3	3	2	-	-	-	-	1	1	-	2
CO4	3	2	3	3	-	-	-	-	1	1	-	2
CO5	3	3	2	3	-	-	-	-	1	1	-	2
CO6	3	3	3	3	-	-	-	-	1	1	-	2

9. Mapping to Program Specific Outcome (PSO)

	PSO1	PSO2	PSO3	PSO4
CO1	1	2	3	2
CO2	2	2	2	3
CO3	2	2	2	3
CO4	2	3	2	2
CO5	2	2	3	2
CO6	1	2	3	2

***** End of Syllabus*****

REFERENCE

IIEST Shibpur IT Department/

NIT Durgapur

Course Name: Advance Linear Algebra
Course Code: DS-406
(Semester IV)
Course Broad Category: Basic Science (BS)

……

1. **Course Prerequisite:**
   Concept of Mathematics in B. Tech 1st and 2nd semester.

2. **Course Learning Objectives:**

   This course is designed to provide students with an understanding of Mathematical concept on Linear algebra that includes basic as well as advanced level. Attempt is taken to cover both theoretical as well as computation perspectives. There are seven components: (i) Linear System of equations, (ii) Vector spaces, (iii) Linear transformations, (iv) Canonical forms and Jordan forms, (v) Inner product spaces and different operators in it, (vi) Bilinear and Quadratic forms, Orthogonal projection and Spectral theory and (vii) Singular value decomposition.

3. **Teaching methodology and evaluation system for the course:**
   Teaching methodology – Lectures and Presentations, Interactive Discussions and Case Studies, Guest Lectures.

   Evaluation System –

A. Mid-Term Exam (40 Marks)- Formative Assessment [Continuous Assessment 1(CIA-1)]
B. Internal Assessment (40 Marks)- Formative Continuous Assessment [Continuous Assessment 1 (CIA-2)]
C. End-Semester Exam (60 Marks)- Summative Assessment.

4. **Course Content:**
   Course Name: Advanced Linear Algebra
   Course Code: DS-406
   Hours per Week: 3L: 0T: 0P
   Credits: 3

Module	Topics	45L
1.	Vector space over a field - Definition and examples, Definition of subspace and intersection, union and sum of two subspaces, Definition of linear span, linearly dependent set, basis and dimensions of a vector space and examples related to it, Extension theorem, ordered basis and coordinate matrix with respect to the ordered basis, Change of Ordered basis in finite dimensional vector space (to find the coordinate matrix by changing the ordered basis), Row space of a matrix, computations concerning subspaces.	9L
2.	Linear Transformation- Definition, examples, properties, range space and null space of a linear transformation, rank nullity theorem, concept of row rank and column rank, Algebra of linear transformations, theorem related to the dimension of L (V, W), definition of linear operator and linear algebra, concept of inverse linear transformation and isomorphism, matrix representation of linear transformations – some examples and related theorems.	9L
3.	Definition of linear functional and dual space of a vector space, some examples of linear functional, double dual space of a vector space and the transpose of linear transformation and related theorem, characteristic value and characteristic polynomial of a linear operator, minimal and characteristic polynomial.	9L
4.	Caley-Hamilton theorem and its application, Invariant subspaces and some related examples, Diagonalizability of an operator. Inner products, norms, orthonormal bases, spectral theorem, unitary and orthogonal transformations.	9L
5.	Direct sum decompositions, Invariant direct sums, Primary Decomposition Theorem, Cyclic subspaces and annihilators, Cyclic decomposition theorem and rational form, Jordon form, concept of singular value decomposition of a rectangular matrix.	9L

5. References:

Text Book:

- Raisinghania, M.D.: Linear Algebra, S. Chand Publishing.
- Balakrishnan, V.K.: Linear Algebra, Schaum's Outline Series.
- Strang, G.: Introduction to Linear Algebra (Fifth Edition), Wellesley-Cambridge Press.
- Hoffman, K., and Kunze, R.: Linear Algebra, Prentice Hall.

Reference Books:

- Friedberg, Stephen H., Insel, Arnold J., and Spence, Lawrence E.: Linear Algebra, Pearson.
- Axler, Sheldon: Linear Algebra Done Right (2nd Edition), Springer.
- Dummit, D.S., and Foote, R.M.: Abstract Algebra, Wiley.
- Lay, David C., Lay, Steven R., and McDonald, Judi J.: Linear Algebra and Its Applications, Pearson.
- Dass, H.K., and Verma, Rama: Linear Algebra, S. Chand Publishing.
- Prasad, K.C., and Datta, K.B.: Linear Algebra, Narosa Publishing House.

6. Course Outcomes (CO):

Course Outcomes	**Details/Statement**	**Action Verb**	**Knowledge Level**
DS-406.CO1	Ability to understand the basic concepts of vector and matrix algebra, including linear dependence / independence, basis and dimension of a subspace, for analysis of matrices and systems of linear equations	Identify	Level 1-Remember
DS-406.CO2	Ability to find the dimension of spaces such as those associated with matrices and linear transformations.	Explain	Level 2-Understand
DS-406.CO3	Ability to understand Dual Space, subspaces, sub space of a linear transformation Minimal and Characteristic Polynomial.	Implement	Level 3-Apply
DS-406.CO4	To understand the construction of matrices for a linear transformation in the triangular/ Jordan form.	Organize	Level 4-Analyze
DS-406.CO5	Apply the decomposition theorem in context of mathematical applications to subspaces.	Assess	Level 5-Evaluate
DS-406.CO6	Build up logical and analytical skills to create a new idea appreciated by academics, research & emerging trends in industry.	Construct	Level 6-Create

7. Mapping of course outcomes to module / course content

Module	**CO1**	**CO2**	**CO3**	**CO4**	**CO5**	**CO6**
1	3	2	-	-	-	1

2	3	3	-	-	-	1
3	3	1	3	1	-	1
4	3	1	2	3	-	1
5	2	-	1	1	3	1

8.Mapping of the Course outcomes to Program Outcomes

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	2	2	1	1	2	-	-	-	-	-	-	1
CO2	1	2	3	1	-	-	-	-	-	-	-	1
CO3	1	2	2	1	1	-	-	-	-	-	-	1
CO4	1	2	1	1	2	-	-	-	-	-	-	2
CO5	2	2	2	2	3	-	-	-	-	-	-	1
CO6	1	1	1	1	-	-	-	-	-	-	-	1

9.Mapping to Program Specific Outcome (PSO)

	PSO1	**PSO2**	**PSO3**	**PSO4**
CO1	1	3	2	2
CO2	2	2	2	3
CO3	1	3	2	2
CO4	2	2	3	2
CO5	2	3	2	2
CO6	2	2	3	2

*** End of the Syllabus ***

DR. B. C. ROY ENGINEERING COLLEGE
DURGAPUR

Object Oriented Programming using Java and Python Lab
Course Code: DS-491
(Semester –IV)
Course Broad Category: Program Core (PC)

……

1. Course Prerequisite:

- Students must know about the basic programming in any language (preferably C or Python).
- Installation of **Java:** JDK 8 or above, Python**:** Version 3.x, **IDEs:** Eclipse (Java); PyCharm, VS Code (Python)

2. Course Learning Objectives:

- Develop foundational OOP skills in both Java and Python.
- Apply concepts such as encapsulation, inheritance, polymorphism, and abstraction.
- Gain hands-on experience with exception handling, multithreading, collections, and file I/O.
- Implement real-world applications using OOP methodologies.

3. Teaching methodology and evaluation system for the course:

Teaching methodology – Lectures and Presentations, Interactive Discussions and Case Studies, LMS, Industry Talk.

Evaluation System –
A. Internal Assessment (60 Marks)- Formative Continuous Assessment
B. End-Semester Exam (40 Marks)- Summative Assessment.

4. Course Content:

Course Name: Object Oriented Programming Using java and python Lab
Course Code: DS-491
Hours per Week: 0L: 0T: 4P
Credits: 2

Unit/Lab	Content

1	**Introduction to Java and Python** • **Java:** Write a program to print "Hello, World!" and demonstrate basic syntax. • **Python:** Write a program to print "Hello, World!" and demonstrate basic syntax.
2	**Data Types, Variables, Control Structures** • **Java:** Demonstrate primitive data types and variable declarations, Implement if-else, switch-case, and loops. • **Python:** Demonstrate dynamic typing and variable assignments, implement if-else and loops using Pythonic constructs.
3	**Functions and Methods** • **Java:** Define methods with various access modifiers and return types. • **Python:** Define functions with default arguments and variable-length arguments**.**
4	**Classes and Objects** • **Java:** Create a class with constructors, methods, and member variables. • **Python:** Create a class with constructors and methods.
5	**Inheritance** • **Java:** Implement single and multilevel inheritance. • **Python:** Implement inheritance and method overriding.
6	**Polymorphism** • **Java:** Demonstrate method overloading and overriding. • **Python:** Demonstrate dynamic method binding and polymorphism.
7	**Abstraction** • **Java:** Use abstract classes and interfaces. • **Python:** Use abstract base classes and interfaces.
8	**Encapsulation** • **Java:** Implement getters and setters for data hiding. • **Python:** Use property decorators for encapsulation.
9	**Exception Handling** • **Java:** Handle exceptions using try-catch blocks. • **Python:** Handle exceptions using try-except blocks.
10	**Multithreading** • **Java:** Create threads using Thread class and Runnable interface. • **Python:** Create threads using threading module.

11	**GUI Programming** • **Java:** Develop a simple GUI application using Swing. • **Python:** Develop a simple GUI application using Tkinter.
12	**Mini Project** • Develop a mini project that incorporates concepts from both Java and Python, such as a student management system or a basic e-commerce application.

5. References:

Text & References Books:

- Mastering Object-Oriented Python: Build powerful applications with reusable code using OOP design patterns and Python 3.7, 2nd Edition, Steven F. Lott (June 14, 2019), **ISBN-10:** 1789531365
- Java Foundations: Introduction to Program Design & Data Structures, 4th edition
- By John Lewis, Peter DePasquale, and Joseph Chase
- Python Programming: A Modern Approach, Vamsi Kurama, Pearson

6. Course Outcomes (CO):

After going through this course the Students will be able to:

Course Outcomes	Details	Action Verb	Knowledge Level
DS-491.CO1	Understand the basics of object-oriented programming using JAVA and Python	Familiarize, Solve	Level 2- Understand
DS-491.CO2	Develop and implement Java programs for simple applications that make use of classes, packages and interfaces	Develop	Level 3- Apply
DS-491.CO3	Develop and implement Java programs using Inheritance and Interfaces.	Implement, Solve	Level 3- Apply
DS-491.CO4	Develop and implement Java programs using array list, exception handling and multithreading.	Implement, Solve	Level 3- Apply
DS-491.CO5	Design applications using file processing, generic programming and event handling	Implement, Develop	Level 3 - Apply Level 6- Create
DS-491.CO6	Develop GUI applications using Applet classes, Swing components and Event handling programs.	Implement, Design	Level 3 - Apply Level 6- Create

7. Mapping of course outcomes to module / course content

Lab Experiments	CO1	CO2	CO3	CO4	CO5	CO6
1	3	-	-	-	-	-

2	-	3	3	-	-	-
3, 4, 5	-	-	3	-	-	-
6, 7	-	-	-	3	-	-
8, 9, 10	-	-	-	-	3	-
11, 12	-	-	-	-	-	3

8. Mapping of the Course outcomes to Program Outcomes (PO)

	PO 1	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PO 12**
CO1	3	2	2	-	-	-	-	-	-	-	-	-
CO2	3	3	2	-	-	-	-	-	-	-	-	-
CO3	3	3	3	2	-	-	-	-	-	-	-	-
CO4	3	3	3	2	-	-	-	-	-	-	-	-
CO5	3	3	3	2	-	-	-	-	2	-	-	-
CO6	3	3	3	2	-	-	-	-	2	-	-	-

9. Mapping to Program Specific Outcomes (PSO)

	PSO1	PSO2	PSO3	PSO4
CO1	2	2	2	1
CO2	2	2	1	2
CO3	2	2	2	2
CO4	2	2	2	2
CO5	2	2	2	2
CO6	2	2	2	2

***** End of Syllabus*****

Course Name: Operating System Lab
Course Code: DS-492
(Semester- IV)
Course Broad Category: Program Core(PC)

……

1. **Course Prerequisite: C Programming and Data Structures, Computer Organization and Architecture**

2. Course Learning Objectives:

i. To learn the Linux command line interface.
ii. To learn shell scripting.
iii. Simulate various operating system algorithms scheduling, memory management etc.
iv. To build a process and threads and their programming interface.
v. Use synchronization tools for inter process communication.

3. **Teaching methodology and evaluation system for the course:**

Teaching methodology – Practical

Evaluation System –

A. Internal Assessment (40 Marks)- Formative Continuous Assessment [Continuous Assessment; Notebook (30 Marks), Viva Voce (20 Marks), Attendance (10 Marks)]

B. End-Semester Exam (60 Marks)- Summative Assessment.

4. **Course Content:**

Course Name: Operating System Lab
Course Code: DS-492
Hours per Week: 0L: 1T: 2P
Credits: 2

Experiment No	Experiment Name	PO	CO
1	Basics of UNIX commands and	**1,12**	1
2	Implementation of Shell Programming	**1,2,12**	3
3	Implementation of CPU Scheduling. (i) FCFS, (ii) SJF, (iii) Shortest Remaining Time First and (iv) Priority (v),RR scheduling.	**2,3**	2

4	Producer-Consumer Problem using Semaphores and Reader Writer Problem	**2,3**	2
5	Implementation of semaphore in the case of IPC	**3,12**	5
6	Implementation of Pipe IPC version V	**3,12**	5
7	Simulate algorithm for deadlock prevention and detection	**3**	2
8	Simulate the Bankers Algorithm for deadlock prevention and detection.	**3**	2
9	Simulate the memory allocation methods 1) Best Fit 2) Worst Fit 3) First Fit	**3**	2
10	Simulate page replacement algorithms: FIFO, LRU and Optimal	**3**	2
11	Implementation of Process and thread (Life cycle of process): (i) Process creation and Termination, orphan and zombie process (ii) Thread creation and Termination, various Thread Functions	**2,3,12**	4
12	To know about the operating system signal interface for handling different signals.	**1,2**	5

5. **References**

Text Book:

1. Operating System Lab-Manual
2. Linux command and shell scripting bible, 3rd Edition by Christine Bresnahan, Richard BLUM
3. Design of the Unix Operating Systems, 8th Edition by Maurice Bach, Prentice-Hall of India.
4. Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati, O'Reilly and Associates

6. Course Outcomes (COs)

On completion of the course students will be able to

Course Outcomes	**Details**	**Action Verb**	**Knowledge Level**
DS-492.CO1	Use of different Linux Commands and their utility.	Use	Level 3-Apply
DS-492.CO2	Simulate & analyze the different operating system algorithms.	Analyze	Level 4-Analyze
DS-492.CO3	Implement the BASH scripting for various system operations	Implement	Level 3-Apply
DS-492.CO4	Apply of process and threads system calls e.g. fork and thread libraries	Apply	Level 3-Apply
DS-492.CO5	Use of signal and different synchronization tools for co-operating processes	Use	Level 3-Apply

7. **Mapping of course outcomes to module / course content**

Module	CO1	CO2	CO3	CO4	CO5
1	3	-	3	-	-
2	-	3	-	-	-
3	-	-	-	3	
4	-	-	-	-	3
5	-	-	-	-	3
6	-	-	-	-	3

8. Mapping of the Course outcomes to Program Outcomes (PO)

Mapping of COs with POs (Course Articulation Matrix):

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	1	1	1	2	1	-	-	-	1	2	2	1
CO2	2	3	3	3	-	-	-	-	2	1	2	
CO3	1	1	3	2	2	-	-	-	1	2	2	1
CO4	3	2	3	2	-	-	-	-	2	2	2	1
CO5	1	1	2	3	-	-	-	-	1	2	2	2

9. Mapping of COs with PSOs (Course Articulation Matrix):

	PSO1	PSO2	PSO3	PSO4
CO1	**1**	**2**	**3**	**2**
CO2	**2**	**2**	**2**	**3**
CO3	**2**	**2**	**2**	**3**
CO4	**2**	**3**	**2**	**2**
CO5	**2**	**2**	**3**	**2**

***** End of Syllabus****

Course Name: Artificial Intelligence Lab
Course Code: DS-493
(Semester– IV)
Course Broad Category: Program Core(PC)

……

1. Course Prerequisite:

- Basic knowledge of computers and general mathematical operations.
- Data Structures and Algorithms

2. Course Learning Objectives:

i. To develop an understanding of fundamental AI algorithms and techniques through hands-on experiments.
ii. To implement AI concepts such as search algorithms, knowledge representation, machine learning, and expert systems using Python.
iii. To apply AI-based problem-solving techniques in real-world applications.
iv. To evaluate AI models for accuracy, efficiency, and robustness.
v. To integrate AI models with data science and analytics tools for predictive insights.

3. Teaching methodology and evaluation system for the course:

Teaching methodology: Instruction: This method recognizes that students have different learning styles, abilities, and backgrounds, and aims to create a learning environment that accommodates these differences.

Evaluation System –

A. **Internal Assessment (60 Marks)-** Formative Continuous Assessment [Continuous Assessment; Note Book (30 Marks), Viva Voce (20 Marks), Attendance (10 Marks)]

End-Semester Exam (40 Marks)- Summative Assessment.

4. Course Content:

Course Name: Artificial Intelligence Lab
Course Code: DS-493
Hours per Week: 0L: 0T: 3P
Credits: 1.5

Experiment Number	Title
1	Implementation of Uninformed Search Algorithms for Breadth First Search (BFS) and Depth First Search (DFS)
2	Implementation of Informed Search Algorithms (A* and Greedy Best-First Search)
3	Solving Constraint Satisfaction Problems (CSPs) using AI techniques
4	Implementing Supervised Learning Models (Decision Trees, Naïve Bayes, SVMs).
5	Implementing Unsupervised Learning Models (K-Means, Hierarchical Clustering).
6	Applying Reinforcement Learning using OpenAI Gym.
7	Developing an AI model for Image Recognition using CNNs.
8	Creating an AI-based Predictive Model using Regression techniques
9	Designing a simple AI-based chatbot using NLP techniques
10	Developing an AI-powered Voice Assistant
11	Implementing a Rule-Based Expert System (Medical Diagnosis System) using AI Technique
12	Designing a simple AI-based chatbot using NLP techniques.

N.B.: The above experiments will be implemented using Python

5. References:

Textbooks:

1. Deepak Khemani, A First Course in Artificial Intelligence, McGraw Hill.
2. Lavika Goel, Artificial Intelligence: Concepts and Applications, Wiley.

Reference Books:

1. Patrick H. Winston, *Artificial Intelligence*, Pearson Education.
2. Tom Mitchell, *Machine Learning*, McGraw Hill.
3. Elaine Rich and Kevin Knight, *Artificial Intelligence*, McGraw-Hill.
4. Stuart Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education.

6. Course Outcomes (CO):

After going through this course the Students will be able to:

Course Outcomes	Details	Action Verb	Knowledge Level
DS-493.CO1	Understanding AI fundamentals and applications	Identify, Explain, Understand	Level 1-Remember, Level 2-Understand
DS-493.CO2	Develop AI-based models for problem-solving	Implement, Design, Construct	Level 3-Apply, Level 6-Create
DS-493.CO3	Evaluate the performance of different AI models.	Compare, Analyze, Evaluate	Level 4-Analyze, Level 5-Evaluate
DS-493.CO4	Integrate AI models with real-world applications.	Apply, Develop, Implement	Level 3-Apply, Level 6-Create
DS-493.CO5	Demonstrate AI techniques in supervised and unsupervised learning.	Utilize, Execute, Illustrate	Level 2-Understand Level 3-Apply ,

7. Mapping of course outcomes to Experiments

Experiment No.	Title	CO1	CO2	CO3	CO4	CO5
1	Implementation of Uninformed Search (BFS, DFS)	✓	✓			
2	Implementation of Informed Search (A*, Greedy BFS)	✓	✓			
3	Solving	✓	✓			

	Constraint Satisfaction Problems (CSPs)					
4	Implementing Supervised Learning Models		✓	✓	✓	✓
5	Implementing Unsupervised Learning Models		✓	✓	✓	✓
6	Applying Reinforcement Learning (Use OpenAI Gym)		✓	✓	✓	✓
7	AI Model for Image Recognition (CNNs)		✓	✓	✓	✓
8	AI-based Predictive Model using Regression		✓	✓	✓	✓
9	AI-based Chatbot using NLP		✓	✓	✓	✓
10	AI-powered Voice Assistant		✓	✓	✓	✓
11	Rule-Based Expert System (Medical Diagnosis)	✓	✓	✓	✓	✓
12	AI-based Chatbot using NLP		✓	✓	✓	✓

8. Mapping of the Course outcomes to Program Outcomes (PO)

	PO1	**PO2**	**PO 3**	**PO4**	**PO5**	**PO 6**	**PO 7**	**PO 8**	**PO9**	**PO10**	**PO11**	**PO12**	**Explanation**
CO1	2	2	-	-	-	-	-	-	-	-	-	-	Understanding AI fundamentals aligns with knowledge application

													and problem analysis.
CO2	3	2	2	-	1	-	-	-	-	-	-	-	AI-based model development requires problem-solving and modern tool usage.
CO3	2	1	2	1	1	-	-	-	-	-	-	-	Evaluating AI model performance involves analysis, research methods, and modern tools.
CO4	2	1	1	1	1	1	1	-	1	1	1	1	AI integration requires problem-solving, ethical considerations, teamwork, and communication
CO5	2	1	1	2	2	-	-	-	2	-	-	-	Supervised and unsupervised learning application involves problem analysis and model evaluation.
AVG	2.2	1.4	1.2	0.8	1	0.2	0.2	-	0.6	0.2	0.2	0.2	

.													

9. Mapping to Program Specific Outcome (PSO)

	PSO1	PSO2	PSO3	PSO4	Explanation
CO1	2	-	-	-	AI fundamentals contribute to understanding data science principles.
CO2	2	1	-	-	AI model development aligns with solving real-world challenges using analytics.
CO3	1	2	1	-	AI performance evaluation enhances insights from data for business and engineering.
CO4	1	2	1	1	AI integration enables solving societal problems using theoretical and industrial tools.
CO5	1	2	1	1	Demonstrating AI techniques aligns with AI application and analytics-driven solutions.

*** End of the Syllabus ***

Course Name: **Big Data Technology and OLAP Lab**
Course Code: DS-494

(Semester- IV)
Course Broad Category: Program Core(PC)

……

1. **Course Prerequisite: Essential Knowledge of Database Management System, Programming languages, SQL and Data Warehouse.**

2. Course Learning Objectives:

1. To realize storage of big data using MongoDB
2. To experiment with various Linux and HDFS commands to store data.
3. To implement Map-Reduce programs for processing big data.
4. Apply Exploratory Data Analytics

3. Teaching methodology and evaluation system for the course:

Teaching methodology –Practical

Evaluation System

A. Internal Assessment (60 Marks)- Formative Continuous Assessment [Continuous Assessment; Note Book (30 Marks), Viva Voce (20 Marks), Attendance (10 Marks)]
B. End-Semester Exam (40 Marks)- Summative Assessment.

4. Course Content:

Course Name: Big Data Technology Lab
Course Code: DS-494
Hours per Week: 0L: 0T: 2P
Credits: 1

Exp. No	Title
1.	Introduction to No-SQL, Difference between RDBMS to NOSQL, JSON and BSON documents, Introduction to MongoDB and its features. Installation of MongoDB and its shell.

Exp. No	Title
2.	Implement NoSQL Database Operations: CRUD operations, Arrays using MongoDB.
3.	Implement Functions: Count – Sort – Limit – Skip using MongoDB. Indexing in MongoDB.
4.	Mongoimport in MongoDB(import Jason in MongoDB)
5.	MongoDB Aggregation, logical operators
6	Dealing with arrays in MongoDB
7	Install, configure and run Hadoop and HDFS. Run basic HDFS shell commands.
8	Implement the following file management tasks in Hadoop: · Adding files and directories · Retrieving files · Deleting files and directories.
9.	Implement word count / frequency programs using MapReduce.
10.	Multi-Dimensional Data modeling Using OLAP
11	Use data set from popular open source dataset sources to perform data cleaning.
12	Use data visualization tool to generate reports on sample datasets.

5. References:

Text Book:

1. Tom White, "Hadoop: The Definitive Guide" Fourth Edition, O'reilly Media, 2015.
2. Bradshaw, Shannon, Eoin Brazil, and Kristina Chodorow. MongoDB: the definitive guide: powerful and scalable data storage. " O'Reilly Media, Inc.", 2019.

Reference Books:

1. Glenn J. Myatt, Making Sense of Data, John Wiley & Sons, 2007 Pete Warden, Big Data Glossary, O'Reilly, 2011.
2. Michael Berthold, David J.Hand, Intelligent Data Analysis, Spingers, 2007.
3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGrawHill Publishing, 2012.

4. Anand Rajaraman and Jeffrey David UIIman, Mining of Massive Datasets Cambridge University Press, 2012.

6. Course Outcomes (CO):

Course Outcomes	Details/Statement	Action Verb	Knowledge Level
DS-494.CO1	Introduction to mongo shell, CRUD Operations, Database Operations, Read and Write Operations.	Understand Apply	Level 2-Understand Level 3-Apply
DS-494.CO2	Apply MongoDB logical operators Aggregation, Indexing	Apply	Level 3-Apply
DS-494.CO3	Configure Hadoop and perform File Management Tasks	Understand	Level 2-Understand
DS-494.CO4	Apply Map-Reduce programs to real time issues like word count.	Apply	Level 3-Apply
DS-494.CO5	Multi-Dimensional Data modeling Using OLAP	Apply	Level 3-Apply
DS-494.CO6	Apply Exploratory data Analytics	Apply	Level 3-Apply

7. Mapping of course outcomes to experiments

EXP No	CO1	CO2	CO3	CO4	CO5	CO6
EXP1	3	3	0	0	0	0
EXP2	0	2	2	0	0	0
EXP3	2	2	0	0	0	0
EXP4	0	0	2	2	0	0
EXP6	2	2	0	0	0	0
EXP7	2	2	0	0	0	0
EXP 8	0	2	2	2	0	0
EXP 9	2	2	0	0	2	0
EXP 10	2	0	2	0	0	0
EXP 11	0	0	0	0	3	0
EXP 12	0	0	0	0	0	3

8. Mapping of the Course outcomes to Program Outcomes (PO)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO7	PO 8	PO 9	PO 10	PO 11	PO 12
CO1	3	2	-	-	1	1	-	-	-	-	-	2
CO2	3	3	2	-	-	-	-	-	-	-	-	2
CO3	3	2	-	-	-	-	-	-	-	-	-	2
CO4	3	2	1	-	-	-	-	-	-	-	-	2
CO5	3	2	2	2	-	-	1	-	-	-	-	2
CO6	2	2	1	1	-	-	-	-	-	-	-	2

9. Mapping to Program Specific Outcome (PSO)

	PSO1	PSO2	PSO3	PSO4
CO1	1	2	3	2
CO2	3	3	2	3
CO3	3	3	3	3
CO4	2	2	2	3
CO5	2	3	3	3
CO6	2	3	3	2

***** End of the Syllabus *****