Zeal Education Society's

ZEAL COLLEGE OF ENGINEERING & RESEARCH, PUNE – 41

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

NBA Accredited, NAAC Accredited with A+ Grade, ISO 21001:2018

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATA SCIENCE

Curriculum Structure and Syllabus of

S.Y. B. Tech. – Artificial Intelligence & Data Science

(With effect from - Academic Year 2025 - 26)
(2024 Pattern)

VISION OF THE INSTITUTE

To be a premier institute in technical education by imparting academic excellence, research, social and entrepreneurial attitude.

MISSION OF THE INSTITUTE

- To achieve academic excellence through innovative teaching and learning process.
- To imbibe the research culture for addressing industry and societal needs.
- To inculcate social attitude through community engagement initiatives.
- To provide conducive environment for building the entrepreneurial skills

MISSION:

M1: To empower students for developing intelligent systems and innovative products through academic excellence.
M2: To produce competent computer professionals to serve the needs of the society by preserving human core values.
M3: To provide research culture and conductive environment for entrepreneurship and to solve industry needs.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs):

PEO1: To provide graduates with the proficiency to utilize the fundamental knowledge of basic sciences, mathematics, Artificial Intelligence, data science and statistics to build systems that require management and analysis of large volume of data.
PEO2: To enrich graduates with necessary technical skills to pursue pioneering research in the field of AI and Data Science and create disruptive and sustainable solutions for the welfare of ecosystems.
PEO3: To enable graduates to think logically, pursue lifelong learning and collaborate with an ethical attitude in a multidisciplinary team.

PROGRAM OUTCOMES (POs):

PO1: **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2: **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3: **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4: **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSOs):

PSO1: Apply theoretical and practical knowledge of Information Technology to design, develop, and manage efficient information systems and interdisciplinary applications

PSO2: Analyze real-world problems, identify system requirements, and design appropriate IT infrastructure and solutions, particularly for large-scale and enterprise computing systems.

PSO3: Demonstrate understanding of professional, ethical, legal, and societal responsibilities in IT practices, along with business and security processes.

PSO4: Exhibit effective communication, teamwork, and decision-making skills using modern tools and technologies to perform responsibly in professional environments.

LIST OF ABBREVIATIONS

Abbreviation	Description
BSC	Basic Science Course
ESC	Engineering Science Course
PCC	Program Core Course
PEC	Program Elective Course
MDM	Multidisciplinary Minor
OE	Open Elective - Other than a particular program
VSEC	Vocational and Skill Enhancement Course
AEC	Ability Enhancement Course
ENTR	Entrepreneurship
EC	Economics
MC	Management Courses
IKS	Indian Knowledge System
VEC	Value Education Courses
RM	Research Methodology
CEP	Community Engagement Project
FP	Field Project
PROJ	Project
INT	Internship
OJT	On Job Training
CC	Co-curricular Courses
HSSM	Humanities Social Science and Management
ELC	Experiential Learning Course
B. Tech	Bachelor of Technology
L	Lecture
P	Practical
T	Tutorial
H	Hours
CR	Credits
CIE	Continuous Internal Evaluation
ETE	End Term Evaluation
TW	Term Work
OR	Oral
PR	Project

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Second Year B. Tech. – ARTIFICIAL INTELLIGENCE AND DATA SCIENCE - III

Course Code	Course Type	Course Name	Teaching Scheme (hrs./Week)							Evaluation Scheme					
			L	P	T	H	CR			CIE	ETE	TW	PR	OR	Total
							TH	PR/ Tut	Total						
ADPC302	PCC	Data structure and Algorithms	3	2	-	5	3	1	4	40	60	-	25		125
ADPC303	PCC	Object Oriented programming with C++	2	2	-	4	2	1	3	40	60	-	-	50	150
ADPC304	PCC	Computer Graphics	3	-	-	3	3	-	3	40	60	-	-		100
ADMD301	MDM	Statistics for AI	3	-	-	3	3	-	3	40	60	-	-	-	100
ALOE301	OE	Open Elective - I #	2	-	-	2	2	-	2	40	60	-	-	-	100
ADMC301	HSSM-MC	Digital Marketing and social media	1	-	1	2	1	1	2	-	-	25	-	-	25
ADVS303	VSEC	Advanced Python	-	4	-	4	-	2	2	-	-	25	25	-	50
ADCE301	CEP	Project Based Learning	-	2	-	2	-	1	1	-	-	25	-	-	25
ADIN302	ELC - INT	Internship – II	4 Week				-	2	2	-	-	25	-	-	25
Total			**14**	**10**	**1**	**25**	**14**	**8**	**22**	**200**	**300**	**100**	**50**	**50**	**700**

# - Select any one course from the given Open Elective Courses		
Course Code	**Course Type**	**Open Elective - I**
ALOE301A	OEC	Digital Literacy and Applications
ALOE301B		Environmental Studies
ALOE301C		Green Energy and Sustainability
ALOE301D		Basics of Consumer Electronics
ALOE301E		Renewable Energy Systems

BoS Chairman

ZES's Zeal College of Engineering & Research PU/PN/Engg./285/2007 Narhe, Pune-41

PRINCIPAL
ZES's Zeal College of Engineering & Research
Narhe, Pune - 411041.

Director

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Second Year B. Tech. – ARTIFICIAL INTELLIGENCE AND DATA SCIENCE – IV

Course Code	Course Type	Course Name	Teaching Scheme (hrs/Week)				Evaluation Scheme							
			L	P	T	H	CR			CIE	ETE	TW	PR	OR	Total
							TH	PR/Tut	Total						
ADPC405	PCC	Operating System	3	2	-	5	3	1	4	40	60	-	-	25	125
ADPC406	PCC	Database Management System	2	2	-	4	2	1	3	40	60	-	25	-	125
ADPC407	PCC	Data Science	3	-	-	3	3	-	3	40	60	-	-		100
ADMD402	MDM	Internet of things	3	-	-	3	3	-	3	40	60	-	-	-	100
ALOE402	OE	Open Elective - II #	2	-	-	2	2	-	2	40	60	-	-	-	100
ADMC402	HSSM-MC	E-Commerce	1	-	1	2	1	1	2	-	-	25	-	-	25
ADAE402	AEC	Quantitive Aptitude & Logical reasoning	-	2	-	2	-	1	1	-	-	25	-	-	25
ADVS404	VSEC	R Programming for Data Science	-	4	-	4	-	2	2	-	-	-	25	50	75
ADIN403	ELC - INT	Internship - III	4 Weeks				-	-	2	-	25	25			25
Total			**14**	**10**	**1**	**25**	**14**	**6**	**22**	**200**	**300**	**75**	**50**	**75**	**700**

# - Select any one course from the given Open Elective Courses		
Course Code	**Course Type**	**Open Elective - II**
ALOE402A	OEC	Cyber Security and Laws
ALOE402B		Sustainability and Climate Change
ALOE402C		Energy Audit and Electrical Safety
ALOE402D		Digital Marketing
ALOE402E		Entrepreneurship and Innovations

BOS Chairman

ZES's Zeal College of Engineering & Research PU/PN/Engg./285/2007 Narhe, Pune-41

PRINCIPAL
ZES's Zeal College of Engineering & Research
Narhe, Pune - 411041.

Director

INDEX

SYLLABUS
SEMESTER - III

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)								**Semester:** III	
Course: Data Structures and Algorithms								**Code:** ADPC302	
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	04	40	60	-	-	25	125

Prerequisites:

- Programming and Problem Solving.
- Object Oriented Programming Using C++.

Course Objectives:

- Understand the importance of data structures and algorithm analysis in efficient programming.
- Study and apply different linear data structures like arrays, stacks, and queues.
- Implement non-linear data structures such as linked lists, trees, and graphs.
- Analyze algorithmic strategies like divide and Conquer and Greedy methods through case studies.
- Apply various sorting, searching, and hashing algorithms.
- Develop real-world applications using appropriate data structures and algorithms.

Course Outcomes: Upon successful completion of the course, students will be able to:

CO1	Understand various types of data structures and algorithms.
CO2	Apply various sorting and searching algorithms for given problem.
CO3	Demonstrate operations on different types of linked lists using dynamic memory allocation.
CO4	Implement stack operations and use stacks in expression evaluation.
CO5	Implement various queue structures and apply them in real-time task scheduling scenarios.
CO6	Demonstrate basic operations on trees and graphs.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Unit I**- Introduction to Data Structures and Algorithms. **Introduction:** Introduction to Data Structures: Abstract Data Types (ADT), Linear and Non-linear, Static and Dynamic. **Algorithms:** Space complexity, Time complexity, Asymptotic notation- Big-O, Theta and Omega, finding complexity using step count method, Analysis of programming Constructs-Linear, Quadratic, Cubic, Logarithmic. Algorithmic Strategies: Introduction to algorithm design strategies- Divide and Conquer, and Greedy strategy. Case Study: E-commerce Product Sorting using Divide and Conquer strategy Google Calendar application using Greedy strategy.	7Hrs
2.	**Unit II**- Linear Data Structures, searching and sorting. **Overview of Array**, Array as an Abstract Data Type, Operations on Array,	7Hrs

	Multidimensional Arrays, Sparse matrix representation using 2D Searching: Sequential Search/Linear Search, Binary Search, Fibonacci Search, and Indexed Sequential Search. **Sorting:** Internal and External Sorting, Bubble sort, Insertion Sort, Selection Sort, Quick Sort, Merge sort **Hashing**: hash table, hash function, basic operations, and collision resolution techniques. Case Study: Inventory Management System in a Retail Store.	
3.	**Unit III - Linked Lists** **Linked list:** Static memory allocation vs Dynamic Memory Management, linked list using dynamic memory management. Introduction of Linked Lists, Primitive Operations on Linked List- Create, Traverse, Search, Insert, Delete, Sort, and Concatenate. Types of Linked List: Singly linked, linear and Circular Linked Lists, Doubly Linked List. Case study: Undo-Redo Functionality in Text Editors**.**	7Hrs
4.	**Unit IV- Stacks** **Stacks:** Stack operations, Multiple Stacks, need for infix, prefix and Postfix expressions, Expression Evaluation and Conversion, Postfix expression evaluation, Linked Stack and Operations. Recursion- concept, variants of recursion- direct, indirect, head, tail. Case study: Web Browser Navigation – Back and Forward Buttons (Stack).	7Hrs
5.	**Unit V- Queue** **Queue:** Queue as abstract data type, Representation of queue using sequential organization, Queue operations, Circular Queue and its advantages, Multi-queue, Linked queue and operations, Deque, Priority queue. Case Study : Task Scheduler with Priority Handling (Priority Queue)	7Hrs
6.	**Unit VI- Graphs and Trees** **Trees:** General tree and its representation: sequential and linked organization, Binary tree- properties, converting tree to binary tree, binary tree traversals (recursive and non-recursive) - inorder, preorder, post order, Operations on binary tree. Binary Search Tree (BST) and its operation. Traversals-depth first and breadth first, Minimum spanning Tree, Greedy algorithms for computing minimum spanning tree- Prims and Kruskal Algorithms. Case study1**:** Product Category Organization using Trees. Case study2: Delivery Route Planning using Graphs.	7 Hrs
	TOTAL	42Hrs

Experiment List:

1. Write a program to manage student attendance records (number of days present). Implement the following:

a) Compute the average attendance of all students.
b) Find the student with highest and lowest attendance.
c) Count the number of students with zero attendance.
d) Display the most common attendance count (mode).

e) Also, determine the time and space complexity of each operation.

2. A hospital maintains patient records using unique patient IDs stored in a list. Write a program to:
a) Linear Search: Check if a particular patient ID exists in the list.
b) Binary Search: Use binary search to quickly locate a patient ID.
Display an appropriate message if the ID is not found

3. An E-Commerce website has a range of products having Product id, name, manufacturer, price and quality rating out of 5. Write a C++ Program to display products as…
a) In Increasing order of Product id (Use Bubble Sort)
b) In Increasing order of Product price (Use Selection Sort)
c) In decreasing order of Product Quality Rating (Use Insertion Sort).

4. Library Book Management using Linked List. Use a singly/doubly linked list to maintain the library catalog. Book Structure: Book ID, Title, Author, Availability Status
a) add Book (): Add a new book to the list.
b) remove Book (book ID): Remove a book using ID.
c) search Book(title/author): Search for a book by title or author.
d) update Book (book ID): Update details like availability.
e) sort Books by (Title/Book ID): Sort and display book records.

5. The ticket booking system of Theater has to be implemented using C++ program. There are M rows and N seats in each row. Doubly circular linked list has to be maintained to keep track of free seats at rows. Assume some random booking to start with. Use an array to store pointers (Head pointer) to each row.On demand
a) The list of available seats is to be displayed.
b) The seats are to be booked.
c) The booking can be cancelled.

6.Implement a real-time browser navigation system using two stacks to simulate back and forward navigation.
The system should support the following operations:
a) Visit New Page: Open a new webpage and update the current state.
b) Back Navigation: Go back to the previous page and store the current one for forward use.
c) Forward Navigation: Move forward to the next page (if back was used before).
d) Display Current Page: Show the currently open webpage.
e) Display Navigation History: Show the back and forward stack contents for debugging or display.
7. Design a print spooling system where multiple print jobs are queued and processed in FIFO order using a queue data structure. The system should support:
a) Add Print Job: A new document is added to the print queue.
b) Process Next Job: The first job in the queue is sent to the printer and removed.
c) Cancel Print Job: Cancel a specific print job if it's still pending.
d) Display Print Queue: List all jobs currently waiting to be printed.

e) Display Processed Jobs: Maintain and display a list of already printed documents (optional – can use a second queue or stack).

8. A call center receives incoming calls, and each call is assigned a unique customer ID. The calls are answered in the order they are received. Your task is to simulate the call queue of a call center using a queue data structure.

a) add Call (customer ID, call Time): Add a call to the queue with the customer ID and the call time (in minutes).
b) answer Call (): Answer and remove the first call from the queue.
c) view Queue (): View all calls currently in the queue without removing them.
d) is Queue Empty (): Check if the queue is empty.

9. Exploring City Landmarks Using Graph Traversals. In a chosen part of your city, consider key landmarks (e.g., Park, Hospital, School, etc.) as nodes in a graph. If there is a path between two landmarks, represent this as an edge. Represent the graph using an adjacency matrix and perform a Depth First Search (DFS) starting from a selected landmark. Also, represent the graph using an adjacency list and perform a Breadth First Search (BFS). Print the order in which each location is visited for both algorithms.

10. Efficient Route Planning for Pizza Delivery Using Graphs. A pizza delivery service receives multiple orders from nearby neighborhoods, which are modeled as nodes in a graph. The travel time between each pair of locations is represented by weighted edges. Write a C++ program that determines the shortest delivery route for the pizza delivery agent to ensure all customers are served in minimum total time. Use appropriate graph traversal algorithms and data structures like priority queues, adjacency lists, or matrices.

11. Managing Data with Binary Search Tree Operations. Design a C++ program that implements a Binary Search Tree (BST) supporting the following operations:

a) Inserting a new element.
b) Deleting an existing element.
c) Searching for a specific value.

Displaying the entire tree in-order, pre-order, or post-order

12. Storing City Data in a Binary Search Tree. Maintain a list of cities and their populations using a Binary Search Tree. Implement functionalities in C++ to:

a) Add a new city.
b) Delete a city.
c) Update the population of a city.
d) Display city names in ascending and descending order.

Compute the maximum number of comparisons needed to find a city in the tree (worst-case search complexity).

Text Books:

1. Horowitz and Sahani, "Fundamentals of Data Structures in C++", University Press 2007, 2nd Edition, ISBN: 978-07-1678-292-6.
2. Data structures and algorithms in python by Michael T. Goodrich, ISBN-13: 978- 1118290279, ISBN-10: 1118290275, Publisher: Wiley; 1st edition (March 18, 2013).

3. Problem Solving with Algorithms and Data Structures Using Python by Bradley N Miller and David L. Ranum. ISBN-13: 978-1590282571, ISBN-10: 1590282574, Publisher: Franklin, Beedle & Associates; 2nd edition (August 22, 2011).

Reference Books:

- Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Pearson Education.
1. Seymour Lipschutz, "Data Structures", Schaum's Outline Series.

E-Resources:

- https://www.ebooks.com/en-us/book/95777110/python-data-structures-and-algorithms/benjamin-baka/
- https://www.ebookphp.com/advanced-data-structures-epub-pdf/
- https://www.ebookphp.com/data-structures-and-algorithms-professional-edition-beginners-guide-epub-pdf/

1. NPTEL archived course on "Programming in C++" by IIT Kharagpur : https://archive.nptel.ac.in/courses/106/105/106105151/

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)									**Semester:** III
Course:- Object Oriented Programming with C++									**Code:** ADPC303
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	02	-	03	40	60	-	50	-	150

Prerequisites:

Knowledge of C programming. Computer Programming in C++

Course Objectives:

1. To learn the object-oriented programming paradigm, focusing on the definition and use of classes along with the fundamentals of object-oriented design.
2. To learn the syntax and semantics of the C++ programming language.
3. To understand the concept of data abstraction and encapsulation, how to design C++ classes for code reuse, how to implement copy constructors and class member functions, to overload functions and operators in C++.
4. To learn how inheritance and virtual functions implement dynamic binding with polymorphism.
5. To learn how to design and implement generic classes with C++ templates and how to use exception handling in C++ programs.

Course Outcomes: After completion of this course, student shall be able to

CO1	Apply constructs- sequence, selection and iteration; classes and objects, inheritance, use of predefined classes from libraries while developing software.
CO2	Design object-oriented solutions for small systems involving multiple objects.
CO3	Use virtual and pure virtual function and complex programming situations.
CO4	Apply object-oriented software principles in problem solving.
CO5	Analyze the strengths of object-oriented programming.
CO6	Develop the application using object oriented programming language(C++).

Course Contents:

Unit	**Fundamentals of Object Oriented Programming**	**Duration (Hrs.)**
1.	**Introduction** Comparison between procedural programming paradigm and object-oriented programming paradigm, Basic data types, Derived data types, Constants, Tokens, Keywords, Identifiers and variables, Concepts of an object and a class, difference between structure and class, control structures, Arrays and Strings **Functions**- Function, function prototype, accessing function and utility function, Constructors and destructor, Types of constructor, Objects and Memory requirements, Static members: variable and functions, inline function, friend	7Hrs

	function.	
2.	**Inheritance** Base Class and derived Class, protected members, relationship between base Class and derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Class Hierarchies, Public and Private Inheritance, Types of Inheritance, Ambiguity in Multiple Inheritance Virtual Base Class, Abstract class, Friend Class, Nested Class.	7Hrs
3.	**Polymorphism** Introduction to Polymorphism, Types of Polymorphism, Operator Overloading-concept of overloading, operator overloading, Overloading Unary Operators, Overloading Binary Operators, Function overloading, Run Time Polymorphism-Pointers to Base class, virtual function and its significance in C++, pure virtual function and virtual table, virtual destructor, abstract base class.	7Hrs
4.	**Files and Streams** Introduction to File Handling, Types of file,streams,header files, File Operations-Opening files, reading form file, Writing to files, File Pointers and Navigation, Error Handling.	7Hrs
5.	**Exception Handling and Templates** Exception handling in c+ try-catch block, throwing exceptions, built-in types, standard exceptions, custom exceptions templates :define templates, template variables, default template arguments, template non-type arguments, template argument deduction function template arguments deduction, class template arguments deduction function templates, class templates, variable templates.	7Hrs
6.	**Standard Template Library (STL)** Basic of STL, STL Components, STL Iterators, Container Adapters Containers-Sequence containers: Array, Vector, Queue,Deque,Forward_list, Associative containers: Set, Multiset, Map, Multimap Unordered associative containers:Unordered_set,Unordered_multiset,Unordered_multimap, Applications, STL Algorithms: Sorting algorithms, searching algorithms, Copying algorithms Counting algorithm	7 Hrs
	TOTAL	42Hrs

List of Experiments:

1. Write a C++ program to find the sum of individual digits of a positive integer
2. Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.
3. Write a program Illustrating Class Declarations, Definition, and Accessing Class Members
4. Write a C++ Program to illustrate default constructor, parameterized constructor and copy constructors.

5. Write a Program to Implement a Class STUDENT having following members:

Data members	
Member	**Description**
sname	Name of the student
Marks array	Marks of the student
total	Total marks obtained
Tmax	Total maximum marks

Member functions	
Member	**Description**
assign()	Assign Initial Values
compute()	to Compute Total, Average
display()	to Display the Data.

6. Write a program to demonstrate the i) Operator Overloading ii) Function Overloading
7. Write a Program to demonstrate friend function and friend class
8. Write a C++ Program that illustrates all types of inheritance.
9. Write a template based program to sort the given list of elements.
10. Write a C++ program containing a possible exception. Use a try block to throw it and a catch block to handle it properly.
11. Write a C++ program to demonstrate the catching of all exceptions.

Text Books:

1. Deitel, "C++ How to Program", 4th Edition, Pearson Education, ISBN:81-297-0276-2
Robert Lafore, "Object-Oriented Programming in C++", fourth edition, Sams Publishing, ISBN:0672323087 (ISBN 13: 9780672323089

Reference Books:

1. Herbert Schildt, "C++-The complete reference" , Eighth Edition, McGraw Hill Professional, 2011, ISBN:978-00-72226805
2. Matt Weisfeld, "The Object-Oriented Thought Process", Third Edition Pearson ISBN-13:075-2063330166
3. E. Balagurusamy, "Object-Oriented Programming with C++", 7th edition, Raw-Hill Publication, ISBN 10: 9352607996 ISBN 13: 9789352607990
Cox Brad, Andrew J. Novobilski, "Object –Oriented Programming: An Evolutionary Approach", Second Edition, Addison–Wesley, ISBN:13:978-020-1548341

E-Resources:

1. **MOOC / NPTEL/YouTube Links:**
2. http://www.nptelvideos.in/2012/11/internet-technologies.html
3. https://freevideolectures.com/course/2308/internet-technology/25video lecture by Prof. Indranil Sengupta, IIT, Kharagpur
4. https://www.digimat.in/nptel/courses/video/106105191/L01.html
5. https://www.w3schools.com/html/html_blocks.asp
6. https://www.javatpoint.com/html-frame-tag

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)									**Semester:** III
Course: Computer Graphics									**Code: ADPC304**
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites: Basic Knowledge of Computer

Course Objectives:

- Remembering: To acquaint the learner with the basic concepts of Computer Graphics.
- Understanding: To learn the various algorithms for generating and rendering graphical figures.
- Applying: To get familiar with mathematics behind the graphical transformations.
- Understanding: To understand and apply various methods and techniques regarding projections, animation, shading, illumination and lighting.
- Creating**:** To generate Interactive graphics using OpenGL.

Course Outcomes:

CO1	Identify the basic terminologies of Computer Graphics and interpret the mathematical foundation of the concepts of computer graphics.
CO2	Apply mathematics to develop Computer programs for elementary graphic operations.
CO3	Illustrate the concepts of windowing and clipping and apply various algorithms to fill and clip polygons.
CO4	Understand and apply the core concepts of computer graphics, including transformation in two and three dimensions, viewing and projection.
CO5	Understand the concepts of color models, lighting, shading models and hidden surface elimination.

Course Contents:

Unit	**Introduction to Computer Graphics**	**Duration (Hrs.)**
1.	Introduction, graphics primitives - pixel, resolution, aspect ratio, frame buffer. Display devices, applications of computer graphics.IO devices **Scan conversion:** Line drawing algorithms: Digital Differential Analyzer (DDA), Bresenham. Circle drawing algorithms: DDA, Bresenham	7Hrs
2.	**Polygon** Introduction to polygon, types: convex, concave and complex. Inside test. **Polygon Filling:** flood fill, seed fill, scan line fill. **Windowing and clipping:** viewing transformations, 2-D clipping: Cohen – Sutherland algorithm line Clipping algorithm, Sutherland Hodgeman Polygon clipping algorithm, Weiler Atherton Polygon Clipping algorithm.	7Hrs
3.	**2-D and 3D transformations** **2-D transformations:** introduction, homogeneous coordinates, 2-D transformations - Translation, scaling, rotation and shear, rotation about an	7Hrs

	arbitrary point. **3-D transformations:** introduction, 3-D transformations - Translation, scaling, rotation and shear, rotation about an arbitrary axis. **Projections :** Parallel (Oblique: Cavalier, Cabinet and orthographic: isometric, diametric, trimetric) and Perspective (Vanishing Points – 1 point, 2 point and 3 point)	
4.	**Light and Colour models** **Colour models**: RGB, HSV, CMY. **Illumination Models**: Ambient Light, Diffuse reflection, Specular Reflection, and the Phong model, Combined diffuse and Specular reflections with multiple light sources, warn model, **Shading Algorithms**: Halftone, Gauraud and Phong Shading. **Hidden Surfaces** Introduction, Back face detection and removal, Algorithms: Depth buffer (z), Depth sorts (Painter), Area subdivision (Warnock).	7Hrs
5.	**Curves:** Introduction, Interpolation and Approximation, Blending function, B-Spline curve, Bezier curve, **Fractals**: Introduction, Classification, Fractal generation: snowflake, Triadic curve, Hilbert curve.	7Hrs
6.	**Introduction to Animation and Gaming** **Segment:** Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility. **Animation:** Introduction, Conventional and computer based animation, Design of animation sequences, Animation languages, Key- frame, Morphing, Motion specification.	7 Hrs
	TOTAL	42Hrs

Text Books:

1. S. Harrington, ―Computer Graphics‖, 2nd Edition, McGraw-Hill Publications, 1987, ISBN 0 – 07 – 100472 – 6.
2. D. Rogers, ―Procedural Elements for Computer Graphics‖, 2nd Edition, Tata McGraw-Hill Publication, 2001, ISBN 0 – 07 – 047371 – 4.
3. Donald D. Hearn, ―Computer Graphics with Open GL‖, 4th Edition, ISBN13: 9780136053583.

Reference Books:

1.J. Foley, V. Dam, S. Feiner, J. Hughes, ―Computer Graphics Principles and Practice‖, 2nd Edition, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.
2. D. Rogers, J. Adams, ―Mathematical Elements for Computer Graphics‖, 2nd Edition, Tata McGraw-Hill Publication, 2002, ISBN 0 – 07 – 048677 – 8.
3. Mario Zechner, Robert Green, ―Beginning Android 4 Games Development‖, Apress, ISBN: 978-81-322-0575-3.

E-Resources:

https://onlinecourses.nptel.ac.in/noc20_cs90/preview

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)								**Semester:** III	
Course: Statistics for AI								**Code:** ADMD301	
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Knowledge of probability and Statistics

Course Objectives:

To enable students to understand and apply statistical concepts and methodologies including bivariate distributions, hypothesis testing, regression analysis, time series modeling, and real-time AI applications, thereby preparing them to analyze and interpret data for AI-based decision-making and systems development.

Course Outcomes: After completion of this course, student will be able to -

CO1	Understand and apply the concepts of discrete and continuous random variables, including joint, marginal, and conditional probability distributions for bivariate cases.
CO2	Demonstrate the ability to construct and interpret statistical hypotheses, including null and alternative hypotheses, and evaluate their significance using statistical inference principles.
CO3	Apply various hypothesis testing techniques, including likelihood ratio tests and non-parametric methods, to draw meaningful conclusions from sample data.
CO4	Analyze relationships between variables using correlation and regression methods, and apply these techniques in AI applications such as feature selection and prediction models..
CO5	Develop and evaluate time series models such as ARIMA, SARIMA, and exponential smoothing for trend analysis and forecasting in real-world datasets.
CO6	Identify and explain the role of statistical models in real-time AI applications including autonomous systems, recommendation engines, and fraud detection systems.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Random Variables & Bivariate Probability Distribution** **I**ntroduction, Random Variable : discrete, continuous with example Bivariate random variable: discrete, continuous, Bivariate probability distribution, Marginal Probability Distribution, Independence of Two Discrete Random Variable, Conditional Probability Distribution Joint Distribution Function of Two Dimensional Discrete R.V.	7Hrs
2.	**Inferential Statistics: Hypothesis** Statistical Inference - Testing of Hypothesis, Non-parametric Methods and Sequential,Analysis:Introduction,Statistical Hypothesis (Simple and-Composite), Test of a Statistical Hypothesis, Null Hypothesis, Alternative Hypothesis, Critical Region, level of Significance, Power of the Test, Principle of likelihood	7Hrs

3.	**Inferential Statistics: Tests for Hypothesis** Most Powerful Test (MP Test), Uniformly Most Powerful Test, likelihood Ratio Test, Properties of Likelihood Ratio Test, Test for the Mean of a Normal Population, Test for the Equality of Means of Two Normal Populations, Test for the Equality of -Means of Several Normal Populations, Test for the Variance of a Normal Population, Test for Equality of Variances of two Normal Populations, Non-parametric Methods, Advantages and Disadvantages of Non-parametric Methods	7Hrs
4.	**Regression & Correlation** Concept of correlation, Types: Positive, Negative, Zero, AI Application: Correlation in feature selection Regression, types of regression (Simple linear regression, Logistic Regression,, Multiple Linear Regression)	7Hrs
5.	**Time Series Analysis** Introduction, Components of time series data MA model – basic and weighted MA model Time series models AR Model, ARIMA Model, SARIMA,SARIMAX,VAR,VARMAX, Simple exponential smoothing model.	7Hrs
6.	**Real-Time Applications in AI** 1.Autonomous Vehicles 2.Chatbots and Virtual Assistants 3.Facial Recognition Systems 4.Real-Time Fraud Detection 5.Personalized Recommendations 6.Industrial Robotics and Automation	7 Hrs
		42Hrs

Text Books:

1. Statistics (Dr.P.G. Dixit)
2. Statistics for Data Scientsists, An introduction to probability, statistics and Data Analysis, Maurits Kaptein et al, Springer 2022
3. Probability and Statistics for Engineering and Sciences,8th Edition, Jay L Devore, Cengage Learning
4. Introduction to Time Series and Forecasting, Second Edition, Peter J Brockwell, Richard A Davis, Springer.

E-Resources:

MOOC / NPTEL/YouTube Links:

1. https://www.w3schools.com/html/html_blocks.asp

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence And Data Science)								**Semester:** III	
Course: Digital Marketing and Social Media								**Code:** ADMC301	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
01	-	01	02	-	-	25	-	-	25

Prerequisites:

1. Basic knowledge of internet usage and browsing.
2. Familiarity with communication tools such as email, blogs, or websites.
3. Interest in business, branding, or marketing principles.

Course Objectives:

1. To provide fundamental knowledge of digital marketing strategies and tools.
2. To familiarize students with social media platforms and their business applications.
3. To introduce concepts of SEO, SEM, email marketing, and analytics.
4. To enable the design and execution of basic digital marketing campaigns.
5. To explore ethical practices and emerging trends in online marketing.

Course Outcomes: After completion of this course, students will able to -

CO1	Understand the core concepts of digital marketing and its advantages over traditional marketing.
CO2	Apply SEO and SEM techniques to enhance online visibility.
CO3	Design social media strategies for different platforms (Facebook, Instagram, LinkedIn, etc.).
CO4	Create and manage email marketing and content marketing campaigns.
CO5	Use tools like Google Analytics to measure marketing effectiveness.
CO6	Analyze case studies to understand trends and challenges in the digital space.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Digital Marketing:** Definition, need and importance; Comparison with traditional marketing; Types of digital marketing; Digital marketing ecosystem; Buyer's journey and conversion funnel.	03
2.	**Search Engine Optimization (SEO):** On-page and off-page SEO; Keywords, backlinks, meta tags; Tools: Google Search Console, SEMrush, Moz; SEO performance metrics.	02
3.	**Search Engine Marketing (SEM) and Paid Ads:** Google Ads, Bing Ads; Ad formats, keyword research, bidding strategies, PPC campaigns; Landing page design and optimization.	03

4.	**Social Media Marketing (SMM):** Overview of platforms (Facebook, Instagram, Twitter, LinkedIn); Creating content calendars, running ad campaigns, influencer marketing; Engagement metrics and KPIs.	02
5.	**Email and Content Marketing:** Email campaign creation, tools like Mailchimp; Email list building, segmentation; Blogging, video marketing, storytelling; Content writing tips and tools.	02
6.	**Analytics and Trends in Digital Marketing:** Web analytics, Google Analytics basics; Conversion tracking, ROI measurement; Current trends: AI in marketing, chatbots, voice search, privacy laws (GDPR, etc.).	02
	TOTAL	**14**

List of Assignments: - (Perform Any 10 Practical's)

1. **SEO Audit Using Free Tools:** Perform an SEO audit of a website using tools like Uber suggest, Screaming Frog, or Google Search Console. Identify technical issues, keywords, and backlink data.
2. **Keyword Research and Content Optimization: Use** Google Keyword Planner or Ubersuggest to find keywords for a blog topic. Optimize sample content using SEO-friendly titles, meta tags, and keywords.
3. **Create and Simulate Google Ads Campaign:** Design a sample Google Ads search campaign using mock billing. Choose keywords, write ad copy, set budget, and identify target audience.
4. **Design and Run a Facebook/Instagram Ad (Simulation): Use** Meta Business Suite to design a mock or actual ad campaign. Define objective, target demographics, ad design, and expected reach.
5. **Create a Social Media Content Calendar: Plan** one week of social media posts for a chosen brand. Include post types, platforms, visuals, and caption samples using tools like Canva or Buffer.
6. **Create and Send Email Campaign Using Mailchimp: Design** and send a sample marketing email using Mailchimp or similar tools. Include subject line, body, image, call-to-action, and analyze open/click-through rate.
7. **Blog Setup and Optimization: Create** a blog using Blogger or WordPress. Write an SEO-optimized post including internal/external links, headings, and media.
8. **Google Analytics Simulation and Reporting:** Set up a Google Analytics demo account and generate basic reports. Interpret metrics like bounce rate, session duration, and traffic sources.
9. **Build a Business Page on LinkedIn/Facebook: Create** a professional brand page for a business/service. Add cover image, description, services, and one sample post.
10. **Reputation Management Simulation:** Track brand mentions online using Google Alerts or Brand24 (trial). Write a brief strategy to handle a mock reputation crisis.
11. **Case Study: Successful Digital Campaign Analysis:** Analyze a famous digital campaign (e.g., Amul, Zomato, Swiggy, Nike). Present objectives, tools used, target audience, campaign creatives, and success factors.

12. **Mini Project: Design a Complete Digital Strategy:** Choose a product/service and create a mini digital strategy. Include SEO plan, ad sample, social calendar, email draft, analytics dashboard (mock), and campaign goal. Submit final report with screenshots and justification of platform choices.

Text Books:

1. Seema Gupta, "Digital Marketing", McGraw Hill Education.
2. Ian Dodson, "The Art of Digital Marketing", Wiley

Reference Books:

1. Philip Kotler, Hermawan Kartajaya, Iwan Setiawan, "Marketing 4.0: Moving from Traditional to Digital", Wiley
2. Ryan Deiss & Russ Henneberry, "Digital Marketing for Dummies", Wiley.
3. Avinash Kaushik, "Web Analytics 2.0", Wiley

E-Resources:

1. Google Digital Garage: https://learndigital.withgoogle.com
2. HubSpot Academy: https://academy.hubspot.com/
3. Coursera – Digital Marketing Specialization: https://www.coursera.org/specializations/digital-marketing
4. Neil Patel Blog: https://neilpatel.com

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence And Data Science)									**Semester:** III
Course: Advanced Python Programming									**Code:** ADVS303
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	04	-	02	-	-	25	-	25	50

Prerequisites:

Fundamental Programming Concepts, Basic Knowledge of python Command Line, Logical Thinking, Basic Computer Literacy

Course Objectives:

- Understand Python syntax, data types, operators, and control structures, including loops and conditionals.
- Understand control structures, including loops and conditionals.
- Gain skills in defining and using functions, handling exceptions, and organizing code into modules.
- Learn to work with data structures such as lists, tuples, dictionaries, and sets, and perform operations on them.
- Learn Strings and perform operations on them.
- Apply Python programming skills to solve real-world problems and implement practical projects.

Course Outcomes:

CO1	Apply computer fundamentals, programming concepts and engineering problem-solving techniques to leverage Python's capabilities for real-world engineering applications.
CO2	Design and implement program logic by applying conditional statements, control statements, loops for constructing complex algorithms
CO3	Apply function concepts to decompose problems into smaller, reusable functions for modular code.
CO4	Students will be able to choose the appropriate data structures for various tasks and apply effective techniques to manipulate them efficiently.
CO5	Students will be able to demonstrate proficiency in string manipulation using various string operations.
CO6	Understand and utilize principles of object-oriented programming (OOP) in Python

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Object-Oriented Programming (OOP) in Python** features of object-oriented programming, classes and objects, methods and message passing, data abstraction and encapsulation, delegation and containership, Polymorphism, python OOP implementation**:** creating classes and objects, constructor and destructor, accessing and modifying attributes, built-in class attributes, instance vs. class variables, advanced oop features: python iterators,	7Hrs

	special methods	
2.	**Inheritance Basics:** Concept of Inheritance in Python: Single Inheritance, Multilevel Inheritance, Multiple Inheritance. Method Overriding, Using Super () Function, Composition vs. Inheritance, Practical Applications: Designing Class Hierarchies, Example Programs for Various Inheritance types.	7Hrs
3.	**File Handling** types of file access: text files, binary files , file operations: opening files: open(), reading files: read(), readline (), readlines(), writing files: write(), writelines(), closing files: close() file modes: read, write, append, read write, exception handling in file operations, working with file paths, file handling using with statement practical assignments: reading from and writing to files, file data processing examples	7Hrs
4.	**Error and Exception Handling** Understanding errors: syntax errors, runtime errors, logical errors, exception handling: try, except blocks, else, finally clauses, raising exceptions: raise statement, creating user-defined exceptions, multiple exception handling nested exception handling.	7Hrs
5.	**Data analysis libraries – numpy and pandas** Introduction to numpy: numpy arrays: creation, indexing, slicing, array operations and broadcasting, mathematical functions using numpy working with numpy: array attributes and methods, aggregations and vectorized operations introduction to pandas: series and dataframe: creation, indexing dataframe operations: selection, filtering, sorting importing and exporting data (csv, excel) handling missing data pandas data manipulation: groupby**,** merge, and pivot tables.	7Hrs
6.	**Data Visualization Libraries – Matplotlib and Seaborn** Introduction to matplotlib: plotting line, bar, scatter, and pie charts customizing plots: titles, labels, legends, colors, and grids, introduction to seaborn: creating attractive statistical graphics, using seaborn for histograms, boxplots, violin plots, heatmaps, advanced visualization: multiple subplots, plot styling and themes, data visualization projects: visualizing real-time data ,exploratory data analysis (eda) plots.	7 Hrs
	TOTAL	42Hrs

List of Experiments:

Unit 1: Object-Oriented Programming (OOP) in Python

1. Create a class Student with attributes like name, roll number, and branch. Create multiple objects and display their details.
2. Write a Python program to demonstrate the use of constructor and destructor.
3. Write a Python program to implement polymorphism using method overriding.

Unit 2: Inheritance

4. Write a Python program to implement single inheritance with base class Employee and derived class Manager.
5. Write a Python program to demonstrate multiple inheritance using super() function.
6. Write a Python program to implement multilevel inheritance and access parent class properties from the grandchild class.

Unit 3: File Handling

7. Write a Python program to read a text file and count the number of vowels, consonants, and spaces.
8. Write a Python program to create a file, write student records, and then display the content of the file.
9. Write a Python program to copy the contents of one file to another file.

Unit 4: Error and Exception Handling

10. Write a Python program to handle division by zero exception using try-except blocks.
11. Write a Python program to handle multiple exceptions including file not found and invalid input.
12. Write a Python program to create and handle a user-defined exception.

Unit 5: Data Analysis Libraries – Numpy and Pandas

13. Write a Python program to perform basic array operations using Numpy: creation, addition, multiplication, transpose, and reshaping.
14. Write a Python program to load a CSV file using Pandas and perform filtering, sorting, and summary statistics.
15. Write a Python program to handle missing data in Pandas using fillna() and dropna() methods.

Unit 6: Data Visualization Libraries – Matplotlib and Seaborn

16. Write a Python program to plot line charts and bar graphs using Matplotlib with labels, titles, and legends.
17. Write a Python program to plot histograms and scatter plots using Seaborn and Matplotlib.
18. Write a Python program to create Seaborn heatmaps and boxplots using a sample dataset.

Text Books:

1. Automate the Boring Stuff with Python: Practical Programming for Everyday Tasks (2nd Edition) by Al Sweigart
2. Python Crash Course, 2nd Edition by Eric Matthes
3. Head First Programming by David Griffiths
4. Fluent Python by Luciano Ramalho
5. Think Python: How to Think Like a Computer Scientist by Allen B. Downey

Reference Books:

1. Python Cookbook by David Beazley and Brian K. Jones
2. Dive Into Python by Mark Pilgrim
3. Learning Python, 5th Edition by Mark Lutz
4. Fluent Python (2nd Edition) by Luciano Ramalho

E-Resources:

1. https://learning.edx.org
2. https://Python.org
3. edX – Python Courses
4. Automate the Boring Stuff with Python – Online Resource
5. W3Schools Python Tutorial

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)						**Semester:** III			
Course: Project Based Learning						**Code:** ADCE301			
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	25	-	-	25

Course Objectives:

- To develop critical thinking and problem solving ability by exploring and proposing solutions to realistic/social problem.
- To Evaluate alternative approaches, and justify the use of selected tools and methods.
- To emphasizes learning activities that are long-term, inter-disciplinary and student-centric.
- To engages students in rich and authentic learning experiences.
- To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.

Course Outcomes:

CO1	Identify the real-life problem from societal need point of view
CO2	Choose and compare alternative approaches to select most feasible one
CO3	Analyze and synthesize the identified problem from technological perspective
CO4	Design the reliable and scalable solution to meet challenges
CO5	Evaluate the solution based on the criteria specified
CO6	Inculcate long life learning attitude towards the societal problems

Course Contents:

Description

Selection of Project/Problem: The problem-based project oriented model for learning is recommended. The model begins with the identifying of a problem, This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated interdisciplinary or subject frame. A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry. There are no commonly shared criteria for what constitutes an acceptable project. Projects vary greatly in the depth of the questions explored, the clarity of the learning goals, the content, and structure of the activity.

• A few hands-on activities that may or may not be multidisciplinary.

• Use of technology in meaningful ways to help them investigate, collaborate, analyse, synthesize, and present their learning.

• Activities may include- Solving real life problem, investigation, /study and Writing reports of in depth study, field work

Assessment:

The institution, head, and mentor are committed to regularly assessing student performance and program

progress. PBL activities are reviewed weekly. Both individual and group performances are continuously monitored and evaluated by supervisors, mentors, and authorities.
Students are expected to maintain a culture of collaboration, self-motivation, peer learning, and personal responsibility. The institution should support this through guidance, orientation, and necessary resources.
Both students and mentors must actively participate in the assessment process. Groups should showcase their work through a public product, report, or presentation.

Assessment Components:

1. **Individual Assessment:** Evaluate each student's understanding, role, and contribution.
2. **Group Assessment:** Assess team roles, work distribution, communication, and teamwork.
3. **Documentation and Presentation:** Evaluate project records,Viva and final presentation.

Evaluation and Continuous Assessment

All PBL activities must be regularly recorded and assessed. Both students and mentors should maintain proper documentation, including the PBL workbook. Each mentor, department, and institute should also maintain a Continuous Assessment Sheet (CAS).

Assessment Parameters and Weightage:

1. **Idea Inception & Awareness** (10%)
   – Consider environment, social, ethical, safety, and legal aspects.
2. **PBL Outcomes & Problem Solving** (40%)
   – Individual and team performance, solution quality, final product.
3. **Documentation** (15%)
   – Requirement gathering, design, implementation, technology use, final report.
4. **Demonstration** (20%)
   – Presentation, user interface, and usability.
5. **Contest Participation / Publication** (15%)

The PBL workbook helps track accountability, punctuality, technical writing skills, and project workflow for students, mentors, and coordinators.

NOTE:

- Choose clear and appropriate assessment methods that are easy to understand for both students and faculty.

- Student groups must follow Software Engineering principles: clearly define the problem, plan the solution, implement, and maintain proper documentation.
- Problem research and exploring multiple solutions is essential and should be emphasized by mentors.
- Apply design thinking: focus first on understanding the user's problem, not just the technology.
- Tutors and mentors must guide students in scoping the project and validate the technology choices.
- Well-documented code can be continued by future batches, enabling the team to work on larger, long-term solutions.

Text Books:

1. A new model of problem based learning. By Terry Barrett. All Ireland Society for higher education (AISHE). ISBN:978-0-9935254-6-9; 2017
2. Problem Based Learning. By Mahnazmoallem, woei hung and Nada Dabbagh, Wiley Publishers. 2019.

3. 3. Stem Project based learning and integrated science, Technology, Engineering and mathematics approach. By Robert Capraro, Mary Margaret Capraro
Reference Books: 1) 1.De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007. 2) 2.Gopalan," Project management core text book", 2 Indian Edition 3) 3.James Shore and Shane Warden, " The Art of Agile Development"
E-Resources:
1.Software Engineering - Tutorials Point 2.Software Engineering – Geeks for Geeks

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)								**Semester:** III	
Course: Internship – II								**Code:** ADIN302	
Teaching Scheme (Hrs./week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	-	-	02	-	-	25	-	-	25

Preamble:

Internships serve as vital educational and career development experiences, offering practical exposure in a specific field. Employers seek individuals who possess the necessary skills and an understanding of industry environments, practices, and cultures. This internship is designed as a structured, short-term, supervised training program, often centered on specific tasks or projects with clear timelines. The primary goal is to immerse technical students in an industrial setting, providing experiences that cannot be replicated in the classroom. This exposure aims to develop competent professionals who understand the social, economic, and administrative factors influencing the operations of industrial organizations.

Course Objectives:

1. Exposure to students to the industrial environment, which cannot be provided in the classroom and hence creating deployable professionals for the industry.
2. Learn to implement the technical knowledge in real industrial situations.

Course Outcomes: After completion of this course, students will be able to -

CO1	Gain exposure to industry practices and understand how academic concepts are applied in professional settings.
CO2	Develop and demonstrate effective communication and teamwork skills within a work environment.
CO3	Improve your problem-solving and time management skills by working in real-world industry settings.

Internship Requirements

1. **Internship Duration:** It is mandatory for all students to undergo an internship after every semester during vacations for the duration of 4 weeks. Internships completed during this period will be considered for the assessment of Term Work (TW).
2. **Internship Opportunities:** Students can explore various opportunities for internships at:
    a. Industries
    b. Research labs or organizations
    c. Collegiate clubs
    d. In-house research projects
    e. Online internships
3. **Support and Assistance:** Students can seek assistance for securing internships from:
    a. The Training and Placement cell, along with departmental coordinators
    b. Department or institute faculty members
    c. Personal contacts
    d. Directly connecting with industries or organizations

4. **Request Letter:** Once an industry, research organization, or collegiate club is identified, students must obtain a request letter from the concerned department or placement office. This letter, in the standard format must be duly signed by the authority, should be addressed to the HR manager or relevant authority.
5. **Confirmation Letter:** Students must submit the confirmation letter from the industry, research organization, or collegiate club to the Internship Coordinator and the Head of Department (HOD) office.
6. **Joining Report:** Upon commencing the internship, students must submit the joining report, joining letter, or a copy of the confirmation email to the Internship Coordinator and the HOD office.
7. **Faculty Mentor:** A faculty member will be assigned as a mentor to a group of students. The mentor will be responsible for monitoring, evaluating, and assessing student internship activities. The faculty mentor is also required to visit the internship location and submit formal feedback to the Internship Coordinator.
8. **Faculty Visits:** Faculty members are advised to visit the internship site once or twice during the internship period to monitor progress.
9. **Progress Report:** Students must submit progress report fortnightly to their faculty guide and the final internship report to the Internship Coordinator and department office.
10. **Evaluation Report:** After the completion of the internship, the mentor, along with the assessment panel members, should submit the evaluation report of the students to the department office and the Internship Coordinator.
11. **Internship Certificate:** Students must receive the Internship Certificate from the industry and submit it to the Internship Coordinator and department office.
12. **Presentation and Assessment:** Students are required to give a presentation on their internship work as part of the term work. The internship diary and report will also be verified and assessed.

SYLLABUS
SEMESTER - IV

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)								**Semester:** IV	
Course: Operating System								**Code:** ADPC405	
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	02	-	04	40	60	-	25	-	125

Prerequisites:

Fundamentals of Computer Hardware and Software

Course Objectives:

- **To** understand the major components of an Operating System and its functions.
- **To** introduce the concept of a process and its management, including transitions and scheduling.
- **To** understand basic concepts related to Inter-Process Communication (IPC),
- **To** understand the concepts and implementation of memory management
- **To** understand the functions of an Operating System in storage management and device management.
- **To** study the need for and fundamentals of special-purpose operating systems in light of emerging technologies**.**

Course Outcomes: At the end of the course, Student will be able to -

CO1	Understand the basic concepts and functions of Operating Systems.
CO2	Describe process management policies and illustrate CPU scheduling of processes.
CO3	Explain and apply synchronization primitives and evaluate how Operating Systems handle deadlock conditions.
CO4	Describe and analyze memory allocation and management functions of Operating Systems.
CO5	Illustrate I/O and file management policies.
CO6	Compare the functions of various special-purpose Operating Systems.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to OS** What is OS, Structure, Goals of OS, OS functions and characteristics, Evolution of OS, Issues in OS design, Types of System Calls, Types of OS: Batch, time sharing, multiprogramming, distributed, network and real-time systems,	7Hrs
2.	**Process Management** Process concept, Process States, Process Control, Process Operations, Threads: Types, Multicore Programming, Multithreading, Issues, Process Scheduling: Types of process schedulers, Types of scheduling, Scheduling algorithms:	7Hrs
3.	**Process Coordination** Synchronization: Principles of Concurrency, Mutual Exclusion, Semaphores, Mutex, Classical synchronization problems: R/W Problem, Producer and Consumer problem,	7Hrs

	Deadlock: Principles, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.	
4.	**Memory Management** Memory Management Requirements, Memory Partitioning: Fixed, Dynamic Partitioning, Buddy System, Paging, Segmentation. Virtual Memory: Concepts, Paging, Segmentation, Page Replacement Policies.	7Hrs
5.	**I/O and File Management** I/O Devices, Organization of the I/O Function, I/O Buffering, Disk Scheduling Policies. File Management: Concepts, File Organization, Access methods, File Directories, File Sharing. Record Blocking, Secondary Storage Management.	7Hrs
6.	**Special-purpose Operating Systems** Open-source and Proprietary OS, Distributed OS, Network OS, Embedded OS, Cloud and IoT OS, Real-Time OS, Mobile OS, Multimedia OS, Comparison of Functions in Various Special-purpose OS.	7 Hrs
	TOTAL	**42hrs**
List of Experiments/Exercises: At least 10 Exercises/Activity to be Completed		

1. Case Study:
   a. Brief History of UNIX/Linux, Windows 9x, Mac OS, Chrome OS, Mobile OS
   b. Architecture UNIX/Linux, Windows 9x, Mac OS, Chrome OS, Mobile OS
   c. Installation of Unix/Linux Windows 9x, Mac OS, Chrome OS, Mobile OS
2. Use of Vi Editor with Any example
3. Use of Nano Editor Any example
4. Use of Power Shell Any example
5. Use of Vim/ Sublime Any example
6. Execution of Unix General Purpose Utility Commands
   a. echo, clear, exit, date, time, uptime, cal, cat, tty, man, which, history, id, pwd, whoami, ping, ifconfig, pr, lp, lpr, lpstat, lpq, lprm, cancel, mail, etc.
7. Study the Unix, windows & Mac OS file system (tree structure), file and directory permissions, and single-user and multi-user environments.
8. Execution of File System Management Commands
   a. ls, cd, pwd, cat, mkdir, rmdir, rm, cp, mv, chmod, wc, piping and redirection, grep, tr, echo, sort, head, tail, diff, comm, less, more, file, type, wc, split, cmp, tar, find, vim, gzip, bzip2, unzip, locate, etc.
9. Execution of User Management Commands
   a. who, whoami, su, sudo, login, logout, exit, passwd, useradd/adduser, usermod, userdel, groupadd, groupmod, groupdel, gpasswd, chown, chage, chgrp, chfn
10. Execution of Process Management Commands
    ps, pstree, nice, kill, pkill, killall, xkill, fg, bg, pgrep, renice, etc
11. Execution of Memory Management Commands
    free, /proc/ meminfo, top, htop, df, du, vmstat, demidecode, sar, pagesize, etc.
12. Study of Shell Programming
    Types of Shell, Variables and Operators, Control Structure,

1) Write a shell script to perform arithmetic operations.

2) Write a shell script to calculate simple interest.
3) Write a shell script to determine largest among three integer numbers
4) Execute the following advanced shell scripts using grep / sed commands:
   Write a script using grep command to find the number of words character, words and lines in a file.
5) Write a script using egrep command to display list of specific type of files in the directory.
   Write a script using sed command to replace all occurrences of particular word in given a file.

1. **Text Books:**

1. A. Silberschatz, P. Galvin, G. Gagne, Operating System Concepts, 10th ed., Wiley, 2018.
2. W. Stallings, Operating Systems: Internal and Design Principles, 9th ed., Pearson, 2018.
3. A. Tanenbaum, Modern Operating Systems, Pearson, 4th ed., 2015.
4. S. Das, Unix Concepts and Applications, 4th ed., McGraw Hill, 2017.
5. R. Michael, Mastering Unix Shell Scripting, 2nd ed., Wiley, 2008.
6. D. Ambawade, D. Shah, Linux Labs and Open-Source Technologies, Dreamtech Press, 2014.

Reference Books:

1. N. Chauhan, Principles of Operating Systems, 1st ed., Oxford University Press, 2014.
2. A. Tanenbaum and A. Woodhull, Operating System Design and Implementation, 3rd ed., Pearson.
3. R. Arpaci-Dusseau and A. Arpaci-Dusseau, Operating Systems: Three Easy Pieces, CreateSpace Independent Publishing Platform, 1st ed., 2018.
4. Y. Kanetkar, Unix Shell Programming, BPB Publications, 2003.
5. B. Forouzan and R. Gilberg, Unix and Shell Programming, Cengage Learning, 2003.

E-Resources:

1. https://repository.dinus.ac.id/docs/ajar/Operating_System.pdf
2. https://nptel.ac.in/courses/106/105/106105214/
3. https://nptel.ac.in/courses/106/106/106106144/

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)								**Semester:** IV	
Course: Database Management Systems								**Code:** ADPC406	
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
02	02	-	03	40	60	-	-	25	125

Prerequisites:

Discrete Mathematics, Data Structures

Course Objectives:

To describe fundamental database concepts, design principles, and develop ER/EER models.
2. **To develop** SQL and PL/SQL programs for effective data manipulation and procedural operations.
3. **To apply** normalization techniques for designing consistent and well-structured relational databases.
4. **To analyze** transaction processing, concurrency control mechanisms, and recovery techniques for reliable database systems.
5**. To evaluate** NoSQL database models and **assess** their suitability for managing unstructured and semi-structured data.

Course Outcomes: On completion of the course, learner will be able to -

CO1	Explain the fundamentals of database management systems, including data models, ER Modeling, and database design.
CO2	Construct **and** implement SQL and PL/SQL code to manage and manipulate data in relational databases.
CO3	**Apply** normalization rules to enhance database structure and **ensure** data consistency and integrity**.**
CO4	**Analyze** transaction properties and **evaluate** concurrency control techniques for ensuring database reliability**.**
CO5	Compare various NoSQL database models and **justify** their use for handling diverse and unstructured data formats**.**
CO6	Explain the fundamentals of database management systems, including data models, ER Modeling, and database design.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Unit I - Introduction to Database Management System.** Unit I: Introduction, Purpose of Database Systems, Database System Applications, View of Data, Database System Architecture. Data Models. Database Design and ER Model: Entity, Attributes, Relationships, Constraints, Keys, Design Process, Entity Relationship Model, ER Diagram, Extended E-R Features, converting ER and EER diagram into tables.	7 Hrs

2.	**Unit II – SQL Queries** Unit II: Database Languages. SQL Characteristics and Advantages, Data Types and Literals, DDL, DML, Select Queries and clauses, SQL Operators, Aggregate Functions. Case Study: Implementation of Unit 1 case study using SQL.	7 Hrs
3.	**Unit III: Advanced SQL Queries and PL/SQL** View, Indexes, Group by and Having Clause, Join Queries, Set operation, DCL, TCL, PL/SQL: Procedure, Function, Cursors, Trigger Case Study: Design and implement a Student Course Management System using SQL and PL/SQL to manage students, courses, and faculty members efficiently. The system should store and retrieve relevant data, ensuring integrity, security, and performance optimization.	7 Hrs
4.	**Unit IV - Relational Database Design.** Relational Model: Basic concepts, Attributes and Domains, CODD's Rules, Relational Integrity: Domain, Referential Integrities, Enterprise Constraints. Database Design: Features of Good Relational Designs, Normalization, Atomic Domains and First Normal Form, 2NF, 3NF, BCNF Case Studies: Design and Optimization of a Relational Database for a University Management System.	7 Hrs
5.	**Unit IV - Database Transactions** Basic concepts of a Transaction, ACID Properties, Concept of Schedule, serial schedule, Serializability – Conflict and View, Concurrency Control Protocols- Lock based and timestamp-based protocols, Recovery techniques. Case study: Design Online Shopping Cart Transaction Management In an e-commerce platform, multiple users simultaneously add, update, and purchase products. To ensure data consistency and reliability, the system must handle concurrent transactions effectively	7 Hrs
6.	**Unit V - NoSQL Database** Introduction to NoSQL Database, NoSQL data models, CAP Theorem and BASE Properties, Comparative study of SQL and NoSQL, MongoDB: CRUD Operations, Indexing and Aggregation, Map Reduce. Case study: Study NoSQL Database Selection for a Social Media Platform.	7 Hrs
	TOTAL	42 Hrs

Guidelines for Instructor's Manual

The instructor's manual should briefly cover the university/program background, course objectives, syllabus, lab conduction methods, and assessment criteria. It should provide topic-wise concepts, expected outcomes, and a list of practical assignments using languages such as **C, C++, or Java**. Sample code structures, algorithmic logic, and references should be included to support effective lab delivery. Standard evaluation rubrics, viva questions, and clarity on CO (Course Outcome) mapping must be provided. The manual is intended to ensure consistency in lab sessions and support instructors in aligning activities with the desired learning outcomes.

Guidelines for Student's Lab Journal

The student's lab journal should follow a uniform structure containing the experiment title, objective, problem statement, algorithm or pseudocode, program code, sample input/output, and conclusion. It should reflect individual work and understanding. Each journal must include a content index and a summary of weekly progress. Code documentation and neat presentation are expected. This record serves as the student's learning evidence and must be duly signed and verified each week by the lab instructor.
Guidelines for Lab /TW Assessment
The term work should be continuously assessed throughout the semester. Marks should be allotted based on timely submissions, code functionality, understanding of concepts, journal completion, and viva performance. Experiments should be graded using standardized rubrics focusing on logic design, correctness, output, and documentation. Bonus weightage may be given for case study analysis and the mini-project. The goal is to promote regular engagement, practical proficiency, and overall conceptual clarity.
Guidelines for Practical Examination
The practical examination should assess the student's coding ability, problem-solving skills, and conceptual understanding. It should consist of one programming problem from the performed experiments and an oral viva. Students must write and execute the solution independently within the given time frame. Evaluation should be based on execution correctness, code structure, and oral response, using standard rubrics. Both internal and external examiners must align on evaluation methods.
Guidelines for Laboratory Conduction
Laboratory sessions should begin with a brief explanation of the topic, objectives, and expected outcomes. Instructors should encourage active participation, individual coding practice, and peer learning. Faculty should monitor progress during the session and resolve student doubts. Weekly viva or discussion should be conducted for conceptual reinforcement. A lab timetable, attendance record, and assessment log should be maintained. The aim is to build consistent technical skills and analytical thinking through hands-on practice.
List of Experiments/Exercises: At least 10 Exercises/Activity to be Completed
Real-Time Application Design with ER Modeling— CO1
1. Decide a real time application and formulate a problem statement for the application to be developed. Propose a Conceptual Design using ER features using tools like ERD plus, ER Win etc. (Identifying entities, relationships between entities, attributes, keys, cardinalities, generalization, specialization etc.) Convert the ER diagram into tables on paper and propose an optimal database design using different normalization concepts.
DDL Operations in SQL — CO2,CO3
2. Write and execute SQL Data Definition Language (DDL) commands such as CREATE, ALTER, DROP, RENAME, and TRUNCATE to define and modify tables. Insert data into the tables and apply appropriate integrity constraints such as NOT NULL, UNIQUE, PRIMARY

KEY, FOREIGN KEY, and CHECK. (The application may vary as per the subject teacher's requirement.
Data and Access Control with SQL – CO2, CO4
Design and run SQL queries to demonstrate the following: a) Data Manipulation (DML): Use SQL statements to INSERT, UPDATE, and DELETE records. Apply arithmetic, logical, set operators, pattern matching, and string functions. b) Access Control (DCL): Use GRANT, REVOKE, and ROLE commands to manage user access. c) Transaction Control (TCL): Apply START TRANSACTION, COMMIT, ROLLBACK, and SAVEPOINT commands to manage transactions.
SQL Computation and Functions- CO2
Implement SQL queries for different SQL concepts for SQL computation (arithmetic operators, logical operators, pattern matching, IN and NOT IN predicates, and MySQL built-in functions).
Grouping and Views in SQL – CO2
5. Implement SQL queries for different SQL concepts such as grouping data using the GROUP BY clause, employing the HAVING clause, applying the EXISTS/NOT EXISTS operators, Creating and using Database Views.
Joins and Subqueries in SQL – CO2
6. Write SQL queries to demonstrate the different SQL concepts like subqueries, performing various join operations, and using set operators in MySQL.
Stored Procedure and PL/SQL Block — CO2
7. Write a Stored Procedure namely prograde for the categorization of student. If marks scored by students in examination is <=1500 and marks>=990 then student will be placed in distinction category if marks scored are between 989 and 900 categories is first class, if marks 899 and 825 category is Higher Second Class. Write a PL/SQL block to use procedure created with above requirement. Stud Marks (name, total marks) and Result (Roll, Name, Class)
Triggers for Salary Tracking— CO2
8. A Company wants to track employee salary changes, maintain company-wide statistics, and log employment history efficiently using row-level triggers. Implement BEFORE and AFTER triggers on EMPLOYEE, COMPANY_INFO, and EMP_LOG tables using INSERT, UPDATE and DELETE operations. EMPLOYEE (Emp_Id, First_Name, Last_Name, Email, Phone No, Hire Date, Job Profile, Salary, HRA) COMPANY_INFO (Emp_Count, Total_Salary_Expenses) EMP_LOG(Emp_Id, Old_Salary, New_Salary, Edit_Time, Job_Status)
Parameterized Cursor for Merging Data – CO2

9. Write a PL/SQL block of code using parameterized Cursor that will merge the data available in the newly created table Rollcall with the data available in the table O_RollCall. If the data in the first table already exist in the second table then that data should be skipped.

CRUD Operations in MongoDB — CO5

10. An institute maintains details of all teachers, including name, qualifications, department details, experience, and salary structure, date of joining, appointement_nature and area of expertise. Design and implement MongoDB queries to perform CRUD operations on the teacher's collection for various administrative tasks. Create the above collection, insert suitable documents and design updation and retrieval queries requiring comparison and logical operators, save() method, etc.

MongoDB Aggregation and Indexing — CO5

11. MongoDB - Aggregation and Indexing: Design and Develop MongoDB Queries using aggregation and indexing with suitable examples using MongoDB.

MongoDB Map-Reduce— CO5

12. MongoDB - Map reduce operations: Implement Map reduces operation with a suitable example using MongoDB.

Java MySQL Connectivity – CO2

Write a program to implement MySQL database connectivity with Java as a front-end language to implement Database navigation operations (add, delete, edit etc.)

Text Books:

1. SilberschatzA.KorthH., SudarshanS.,"Database System Concepts", McGrawHill Publishers,7th Edition, 2020 ISBN 978-0-07-802215-9.
2. Ivan Bayross, "SQL, PL/SQL the Programming Language of Oracle", BPB Publications, 2014 ISBN: 9788176569644.
3. Connally T,BeggC.,"DatabaseSystems-APracticalApproachtoDesign,Implementationand Management", Pearson Education, 5th Edition, 2010, ISBN 81-7808-861-4.
2. Pramod J. Sadalage and Martin Fowler, "NoSQL Distilled", Addison Wesley, ISBN 10: 0321826620, 2013, ISBN 13: 978-0321826626.

Reference Books:

1. C. J. Date, "An Introduction to Database Systems", Addison-Wesley, 8th Edition, 2004, ISBN 0321189566. S. K. Singh, "Database Systems: Concepts, Design and Application", Pearson Education, 2009,ISBN 9788177585674.Kristina Chodorow, Michael Dierolf, "MongoDB: The Definitive Guide", O"Reilly Publications, 3rd Edition, 2019 ISBN 9781491954461.
2. *Kevin Roebuck, "Storing and Managing Big Data - NoSQL, HADOOP and More", Emereo Pty Limited, 2011,ISBN. 1743045743, 9781743045749.*

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)									Semester: IV
Course: Data Science									Code:- ADPC407
Teaching Scheme									Evaluation Scheme
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Python Programming, Statistics, Linear Algebra

Course Objectives:

- To introduce the core concepts, methodologies, and tools of data science.
- To analyse, visualize, and interpret complex data.
- To apply statistical and machine learning models to solve real-world problems.
- To develop proficiency in tools like Python, pandas, NumPy, scikit-learn, and Jupyter.
- To build, evaluate, and deploy data-driven models and communicate insights effectively.

Course Outcomes:

CO1	Understand the fundamentals and lifecycle of data science with data types and sources.
CO2	Perform data cleaning, preprocessing, and exploratory data analysis using Python.
CO3	Apply statistical analysis and hypothesis testing for data inference.
CO4	Build and evaluate machine learning models for supervised and unsupervised learning.
CO5	Utilize advanced techniques such as ensemble learning, feature engineering, NLP, and time series.
CO6	Perform end-to-end data science projects using real-world datasets and deploy solutions.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to Data Science** What is Data Science? Roles of a data scientist, Data Science process/lifecycle, Data types: structured, semi-structured, unstructured. Data sources: APIs, Web scraping, databases, files. Introduction to Python for Data Science: Jupyter, Numbly, pandas, Working with data frames, basic operations	7Hrs
2.	**Data Wrangling, Cleaning, and Visualization** Handling missing data, duplicates, and outliers, Data normalization and transformation, Feature scaling, encoding categorical variables, Data visualization tools: matplotlib, seaborn, plotly, EDA techniques: histograms, boxplots, heatmaps, pairplots, Data imbalance handling (SMOTE, under sampling)	7Hrs
3.	**Statistical Analysis and Hypothesis Testing** Descriptive statistics: measures of central tendency & dispersion, Probability	7Hrs

	distributions: Normal, Binomial, Poisson, Inferential statistics: Sampling, confidence intervals, Hypothesis testing: t-test, ANOVA, chi-square, A/B Testing & its applications in real-world scenarios	
4.	**Machine Learning for Data Science** Machine learning workflow, Supervised learning: Linear regression, logistic regression, decision trees, random forest. Model evaluation: accuracy, precision, recall, F1-score, confusion matrix, ROC-AUC. Unsupervised learning: k-means, hierarchical clustering, PCA, Model selection and cross-validation	7Hrs
5.	**Real-time Data Processing & MLOps** Real-time Data and Stream Processing, Introduction to streaming data, Data Pipelines and Workflow Automation, Batch vs stream processing pipelines, Data pipeline design principles. Introduction to MLOps, MLOps lifecycle: model versioning, CI/CD for ML, MLFlow: tracking experiments, model registry. Data versioning with DVC, Model Deployment, Serving models via REST APIs using FastAPI or Flask, Model containerization with Docker.	7Hrs
6.	**Advanced Data Science Topics & Responsible AI** Feature Engineering, Feature transformation, binning, interactions. Feature selection: filter, wrapper, and embedded methods. Time Series Analysis, Components: trend, seasonality, noise. Forecasting models: ARIMA, Holt-Winters, Facebook Prophet, Model Explainability and Interpretability, SHAP, LIME, Partial Dependence Plots (PDPs). Ethics in AI and Data Science, Bias in datasets and models, Fairness, transparency, and accountability, Introduction to responsible AI frameworks (like IBM AI Fairness 360, Google PAIR)	7 Hrs
	TOTAL	**42Hrs**

Text Books:

1. Aurelian Géron, Hands-On ML with Scikit-Learn, Keras & TensorFlow, O'Reilly, ISBN 9781492032649
2.Wes McKinney, Python for Data Analysis, O'Reilly, ISBN 9781491957660

Reference Books:

1. Cathy O'Neil, Rachel Schutt, Doing Data Science, O'Reilly, ISBN 9781449358655
2. Joel Grus, Data Science from Scratch, O'Reilly, ISBN 9781492041139
3. Jake Vander Plas, Python Data Science Handbook, O'Reilly, ISBN 9781491912058

E-Resources:

1. https://nptel.ac.in/courses/106106179
2. https://nptel.ac.in/courses/106106226

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)									**Semester:** IV
Course: Internet of Things									**Code:**ADMD402
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
03	-	-	03	40	60	-	-	-	100

Prerequisites:

Basic Programming Knowledge, Familiarity with C or C++ Language

Course Objectives:

- Introduce how IoT has become a game changer in the new economy where the customers are looking for integrated value
- Bring the IoT perspective in thinking and building solutions
- Introduce the tools and techniques that enable IoT solution and Security aspects
- Implement machine learning and predictive maintenance models using IoT data.

Course Outcomes: On completion of the course, learner will be able to -

CO1	Have a thorough understanding of the structure, function and characteristics of computer systems and Understand the structure of various number systems and its application in digital design
CO2	Develop the skill set to build IoT systems and sensor interfacing.
CO3	Explain the concept of Internet of Things and identify the technologies that make up the internet of things
CO4	Analyze trade-offs in interconnected wireless embedded device networks. Select Appropriate Protocols for IoT Solutions
CO5	Design a simple IoT system comprising sensors by analyzing the requirements of IoT Application
CO6	Identify the Application of IoT in automation of Commercial and Real World examples

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Fundamentals of Computer Organization & Digital Electronics** Basic Organization of Computers, Classification Micro, Mini, Mainframe and Super Computer. System Bus and Interconnection, PCI, Computer Function, I-Cycle, Interrupt and Class of Interrupts. Number systems, Decimal Number system, Binary number system, Octal & Hexadecimal number system, 1's & 2's complement, Binary Fixed Point Representation.	7Hrs
2.	**Communication Interface** Basic Peripherals & their interfacing with 8086/8088, Semiconductor Memory Interfacing-Dynamic RAM Interfacing-Interfacing I/O ports-PIO-8255, Modes of operation-interfacing Analog-Digital Data converter stepper motor interfacing.	7Hrs

3.	**Introduction & IOT Technologies behind smart and intelligent devices** IoT Concepts, Introduction to IOT Communications, Telemetry vs IOT, Applications of IOT Communications, People, Processes and Devices. Automation, asset management, telemetry, transportation, telematics. Telemetry and Telemetric; Report location, logistics, tracking and remote assistance; Next generation kiosks, self-service technology; Cellular IOT connectivity services	7Hrs
4.	**IoT Systems, Network and Protocols** Study of RF Wireless Sensors; Wireless networks; Wireless Sensor Networking (WSN); Cellular Machine-to Machine (M2M) application networks; Computer Connected to Internet; Network Devices; Device configuration and management; Exchange information in real time without human intervention; IoT Protocols.	7Hrs
5.	**IOT Design and System Engineering** Discuss IOT Requirements; Hardware & Software; Study of IOT Sensors; Tagging and Tracking; Embedded Products; IOT Design; SIM Card Technology; IOT Connectivity and Management; IOT Security & IOT Communication	7Hrs
6.	**IOT Applications** IOT Verticals; IOT Hosted Services; IOT Application development, IOT Connectivity; IOT Software providers; Review of various IoT application domains including agriculture, healthcare, manufacturing, device management, and vehicle to vehicle communication and wearable computing devices.	7 Hrs
	TOTAL	42hrs

Text Book:

1.Daniel Minoli, "Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications", ISBN: 978-1-118-47347-4, Willy Publications

2. Bernd Scholz-Reiter, Florian Michahelles, "Architecting the Internet of Things", ISBN 978-3- 642-19156-5 e-ISBN 978-3-642-19157-2, Springer

Reference Book:

a. Hakima Chaouchi, "The Internet of Things Connecting Objects to the Web" ISBN : 978-1- 84821-140-7, Willy Publications

b. Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things: Key Applications and Protocols, ISBN: 978-1-119-99435-0, 2nd Edition, Willy Publications Inside the Internet of Things (IoT), Deloitte University Press

c. Internet of Things- From Research and Innovation to Market Deployment; By Ovidiu & Peter; River Publishers Series

d. Five thoughts from the Father of the Internet of Things; by By Phil Wainewright - Kevin Ashton 6. How Protocol Conversion Addresses IIoT Challenges: White Paper By RedLion.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)								**Semester:** IV	
Course: E-Commerce								**Code:** ADMC402	
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
01	-	01	02	-	-	25	-	-	25

Prerequisites:

1. Basic knowledge of the Internet and the World Wide Web
2. Familiarity with business or commerce principles
3. Interest in digital platforms and technologies

Course Objectives:

1. To provide an understanding of fundamental concepts and models in E-Commerce.
2. To familiarize students with digital business strategies and payment systems.
3. To develop awareness of legal, ethical, and security issues in E-Commerce.
4. To give exposure to digital marketing and E-Commerce tools.

Course Outcomes: After completion of this course, students will able to -

CO1	Understand the foundations and evolution of E-Commerce and its business models.
CO2	Analyze various payment systems, technologies, and infrastructure of E-Commerce.
CO3	Examine security and legal frameworks applicable to E-Commerce.
CO4	Evaluate the ethical, social, and regulatory concerns in digital commerce.
CO5	Apply digital marketing strategies to online businesses.
CO6	Develop and demonstrate a basic E-Commerce prototype or campaign.

Course Contents:

Unit	**Description**	**Duration (Hrs.)**
1.	**Introduction to E-Commerce:** Definition, history, and evolution; traditional vs. electronic commerce; E-Commerce framework and business impact; features and limitations of E-Commerce.	02
2.	**E-Commerce Business Models:** B2B, B2C, C2C, C2B, G2C; revenue models; case studies (Amazon, Flipkart, Meesho, etc.); advantages and challenges.	02
3.	**E-Commerce Infrastructure:** Internet and WWW, web hosting, domain registration, client-server architecture; overview of E-Commerce platforms (Shopify, Wix, Woo Commerce).	03
4.	**Electronic Payment Systems & Security:** **Payment methods:** credit/debit cards, wallets, UPI, net banking; digital signatures; SSL; common threats and prevention; e-fraud case studies.	02
5.	**Legal, Ethical & Regulatory Issues:** Overview of IT Act 2000, data protection laws (GDPR); privacy concerns; ethical and IPR issues in digital business.	03
6.	**Digital Marketing & Strategy:** Introduction to SEO, SEM, affiliate marketing,	02

	email campaigns, and social media marketing; use of Google Ads, Meta Ads, CRM tools.	
	TOTAL	**14**

Text Books:

1. P.T. Joseph, "E-Commerce: An Indian Perspective", PHI Learning
2. Kenneth C. Laudon & Carol Guercio Traver, "E-Commerce 2023", Pearson

Reference Books:

1. S.J. Joseph, "E-Commerce: A Managerial Perspective", Prentice Hall
2. Elias M. Awad, "Electronic Commerce: From Vision to Fulfilment", Pearson

E-Resources:

1. NPTEL: https://nptel.ac.in
2. Google Digital Garage: https://learndigital.withgoogle.com
3. HubSpot Academy: https://academy.hubspot.com

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)							**Semester:** IV		
Course: Quantitive Aptitude & Logical reasoning							**Code:** ADAE402		
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	02	-	01	-	-	**25**	-	-	**25**

Prerequisites:
Knowledge of logic and any programming.

Course Objectives:

To develop fundamental quantitative and logical reasoning skills among students through engaging and practical topics such as number systems, ratios, calendars, coding-decoding, blood relations, and data interpretation. The course aims to enhance problem-solving speed, analytical thinking, and data handling abilities essential for competitive exams and real-world decision-making.

Course Outcomes: After completion of this course, student will be able to -

CO1	Understand and apply number system techniques including simplification, divisibility, and advanced arithmetic shortcuts for fast calculations.
CO2	Analyze and solve problems involving ratio, proportion, and their real-world applications using concepts of duplicate and triplicate ratios.
CO3	Determine the day of the week for any given date using calendar rules and apply logical methods to solve calendar-based reasoning problems.
CO4	Interpret and solve problems based on blood relations using symbols, family trees, and relational logic.
CO5	Decode complex patterns in words, numbers, and symbols, and apply logical reasoning in various coding-decoding formats.
CO6	Interpret data from charts, tables, and graphs to derive statistical insights and apply reasoning skills to visual and analytical problems.

List Of Assignments:

Unit	**Description**	**Duration (Hrs.)**
1.	Number System: Simplification, Speed Math, Squaring and Cubing Techniques, Multiplication Tricks, Divisibility Rules, Number of Divisors, Number of Even & Odd Divisors, Sum of Divisors, Product of Divisors.	4 Hrs
2.	Ratio and Proportions- Concept Explanation, Duplicate Ratio, Triplicate Ratio, Direct Proportion, Indirect Proportion,	4 Hrs
3.	Calendars- Leap and non-leap years, Odd/Extra day concept, odd days' table, counting of odd days, Centennial year and non-centennial years, Date to day calculations, Last day of a year, Forward and backward counting of days, Repetition of a calendar year or	4 Hrs

	month.	
4.	Blood Relations: Relations defined, Generation Verticals, Family Tree, Single Person Blood Relations, Mixed/Chain Blood Relations, Symbol based Blood Relation.	4 Hrs
5.	Coding Decoding: Coding based on order: Letter to Letter Mapping, Letter to number mapping, Letter to digit mapping, Re-ordering sequences; Word sequencing, Match the word to code, Symbol Coding.	4 Hrs
6.	Data Interpretation and Reasoning: Raw and group data, Tabulation, Bar Graphs, Pie Charts, Mean median and Mode, Analytical reasoning, Mirror Image	4 Hrs
	Total	24 Hrs

List of Experiments:(Solve Any two from Each Unit)

Number System

1: Use Vedic math techniques to simplify 10 given arithmetic expressions quickly.
2: Find the sum, product, and count of all divisors for 5 given numbers (e.g., 36, 48, 100, 120, 225).
3: Design a calculator (non-programming) worksheet that can identify whether a number is divisible by 2 to 11.

Ratio and Proportions

1: Solve 5 real-life problems involving direct and indirect proportions (e.g., work-time, distance-speed).
2: Given ingredient ratios in recipes, adjust them for different quantities using duplicate and triplicate ratios.
3: Compare expenditure vs income ratios for 3 sample families and find the highest savings ratio.

Calendars

1: Write steps to calculate the day of the week for 5 given historical/birth dates.
2: Analyze 3 different century years and find which ones are leap years; justify with odd day calculations.
3: Identify all years between 2000 and 2100 when a calendar will repeat itself.

Blood Relations

1: Draw a family tree from a given paragraph describing relations, then answer 5 relation-based questions.
2: Given 5 symbolic relations (e.g., A % B means A is sister of B), decode and solve the relationship.
3: Create your own blood relation puzzle involving 3 generations and test it with a peer

Reference Books:

1. "Quantitative Aptitude for Competitive Examinations" by R.S. Aggarwal.
2. "Logical Reasoning" by S. Chand.
3. "How to Prepare for Logical Reasoning for CAT" by Arun Sharma.
4. "Magical Book on Quicker Maths" by M. Tyra.

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. (Artificial Intelligence and Data Science)									**Semester:** IV
Course: R Programming for Data Science									**Code:** ADVS404
Teaching Scheme				**Evaluation Scheme**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	04	-	2	-	-	25	50	-	75

Prerequisites:

Knowledge of Statistics, Basic Computer Science Knowledge

Course Objectives:

1. Learn Fundamentals of R.
2. Covers how to use different functions in R, how to read data into R, accessing R packages, writing R functions, debugging, and organizing data using R functions.
3. Cover the Basics of statistical data analysis with examples.
4. The whole syllabus will give an idea to collect, compile and visualize data using statistical functions.

Course Outcomes:
On completion of the course, learner will be able to -

CO1	Understand the basics of Fundamentals of R.
CO2	Understands the loading, retrieval techniques of data.
CO3	Understand how data is analyzed and visualized using statistic functions.
CO4	Understand the R – programming functions and data frames for data analysis.
CO5	Understand the descriptive statistics methods.
CO6	Application to implement R techniques for text analysis and understanding

Course Contents:

Unit	Description	**Duration (Hrs.)**
1.	**Introduction to R:** What is R Why R Advantages of R over Other Programming Languages - R Studio: R command Prompt, R script file, comments Handling Packages in R: Installing a R Package, Few commands to get started: installed packages, package Description, help, find package, library Input and Output Entering Data from keyboard Printing fewer digits or more digits Special Values functions : NA, Inf and –inf.	4Hrs

2.	**R Data Types:** Vectors, Lists, Matrices, Arrays, Factors, Data Frame R Variables: Variable assignment, Data types of Variable, Finding Variable ls, Deleting Variables - R Operators: Arithmetic Operators, Relational Operators, Logical Operator, Assignment Operators, Miscellaneous Operators R Decision Making: if statement, if – else statement, if – else if statement, switch statement R Loops: repeat loop, while loop, for loop - Loop control statement: break statement, next statement.	4 Hrs
3.	**R-Function:** function definition, Built in functions: mean, paste, sum, min, max, seq, user-defined function, calling a function, calling a function without an argument, calling a function with argument values R-Strings Manipulating Text in Data: substr, strsplit, paste, grep, toupper, tolower R Vectors Sequence vector, rep function, vector access, vector names, vector math, vector recycling, vector element sorting - R List - Creating a List, List Tags and Values, Add/Delete Element to or from a List, Size of List, Merging Lists, Converting List to Vector R Matrices Accessing Elements of a Matrix, Matrix Computations: Addition, subtraction, Multiplication and Division- R Arrays: Naming Columns and Rows, Accessing Array Elements, Manipulating Array Elements, Calculation Across Array Elements R Factors creating factors, generating factor levels gl.	6 Hrs
4.	**Data Frames in R:** Create Data Frame, Data Frame Access, Understanding Data in Data Frames: dim, nrow, ncol, str, Summary, names, head, tail, edit functions Extract Data from Data Frame, Expand Data Frame: Add Column, Add Row Joining columns and rows in a Data frame rbind and cbind Merging Data frames merge Melting and Casting data melt, cast. Loading and handling Data in R: Getting and Setting the Working Directory getwd, setwd, dir - R-CSV Files - Input as a CSV file, reading a CSV File, Analyzing the CSV File: summary, min, max, range, mean, median, apply Writing into a CSV File R -Excel File Reading the Excel file.	6 Hrs
5.	**Descriptive Statistics in R:** Data Range, Frequencies, Mode, Mean and Median: Mean Applying Trim Option, Applying NA Option, Median Mode Standard Deviation Correlation Spotting Problems in Data with Visualization: visually Checking Distributions for a single Variable R Pie Charts: Pie Chart title and Colors Slice Percentages and Chart Legend, 3D Pie Chart R Histograms Density Plot R Bar Charts: Bar Chart Labels, Title and Colors.	4 Hrs
6.	**3Advanced Data Analysis and Visualization with R**: Statistical Analysis: Descriptive stats, hypothesis testing. Data Visualization Libraries in R: ggplot2. Machine Learning Concepts: Introduction to ML, basic models in R. R Shiny: Building interactive web applications**.** Integrating R with Power BI: Using R scripts	4 Hrs

	and calculations. Data Visualization Ethics and Best Practices. Capstone Project: Applying skills using R and Power BI	
		TOTAL **28** Hrs

Text Books:

1) R Programming for Beginners, Sandip Rakshit ,McGraw Hill Education (India), 2017, ISBN : 978-93- 5260-455-5.

2) Data Analytics using R,Seema Acharya ,McGrawHill Education (India), 2018, ISBN: 978-93-5260-524-8.

Reference Books:

1. " R for Data Science: A Practical Introduction" by Hadley Wickham and Garrett Grolemund.
2. " Beginner's Guide for Data Analysis using R Programming " by Dr. Jeeva Jose
3." R for Dummies A Wiley Brand, 2nd Edition " by Andrie de Vries, JorisMeys,
4." R Programming " by Tutorials Point (I)

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

Program: B. Tech. ((Artificial Intelligence and Data Science)								**Semester:** IV	
Course: Internship – III								**Code:** ADIN403	
Teaching Scheme (Hrs/week)				**Evaluation Scheme (Marks)**					
Lecture	**Practical**	**Tutorial**	**Credit**	**CIE**	**ETE**	**TW**	**OR**	**PR**	**Total**
-	-	-	02	-	-	25	-	-	25

Preamble:

Internships serve as vital educational and career development experiences, offering practical exposure in a specific field. Employers seek individuals who possess the necessary skills and an understanding of industry environments, practices, and cultures. This internship is designed as a structured, short-term, supervised training program, often centered on specific tasks or projects with clear timelines. The primary goal is to immerse technical students in an industrial setting, providing experiences that cannot be replicated in the classroom. This exposure aims to develop competent professionals who understand the social, economic, and administrative factors influencing the operations of industrial organizations.

Course Objectives:

1. Exposure to students to the industrial environment, which cannot be provided in the classroom and hence creating deployable professionals for the industry.
2. Learn to implement the technical knowledge in real industrial situations.

Course Outcomes: After completion of this course, students will be able to -

CO1	Gain exposure to industry practices and understand how academic concepts are applied in professional settings.
CO2	Develop and demonstrate effective communication and teamwork skills within a work environment.
CO3	Improve your problem-solving and time management skills by working in real-world industry settings.

Internship Requirements

1. **Internship Duration:** It is mandatory for all students to undergo an internship after every semester during vacations for the duration of 3 to 5 weeks. Internships completed during this period will be considered for the assessment of Term Work (TW).
2. **Internship Opportunities:** Students can explore various opportunities for internships at:
    a. Industries
    b. Research labs or organizations
    c. Collegiate clubs
    d. In-house research projects
    e. Online internships
3. **Support and Assistance:** Students can seek assistance for securing internships from:
    a. The Training and Placement cell, along with departmental coordinators
    b. Department or institute faculty members
    c. Personal contacts
    d. Directly connecting with industries or organizations

4. **Request Letter:** Once an industry, research organization, or collegiate club is identified, students must obtain a request letter from the concerned department or placement office. This letter, in the standard format must be duly signed by the authority, should be addressed to the HR manager or relevant authority.
5. **Confirmation Letter:** Students must submit the confirmation letter from the industry, research organization, or collegiate club to the Internship Coordinator and the Head of Department (HOD) office.
6. **Joining Report:** Upon commencing the internship, students must submit the joining report, joining letter, or a copy of the confirmation email to the Internship Coordinator and the HOD office.
7. **Faculty Mentor:** A faculty member will be assigned as a mentor to a group of students. The mentor will be responsible for monitoring, evaluating, and assessing student internship activities. The faculty mentor is also required to visit the internship location and submit formal feedback to the Internship Coordinator.
8. **Faculty Visits:** Faculty members are advised to visit the internship site once or twice during the internship period to monitor progress.
9. **Progress Report:** Students must submit progress report fortnightly to their faculty guide and the final internship report to the Internship Coordinator and department office.
10. **Evaluation Report:** After the completion of the internship, the mentor, along with the assessment panel members, should submit the evaluation report of the students to the department office and the Internship Coordinator.
11. **Internship Certificate:** Students must receive the Internship Certificate from the industry and submit it to the Internship Coordinator and department office.
12. **Presentation and Assessment:** Students are required to give a presentation on their internship work as part of the term work. The internship diary and report will also be verified and assessed.