

Amrutvahini Sheti and Shikshan Vikas Sanstha's

AMRUTVAHINI COLLEGE OF ENGINEERING, SANGAMNER

Amrutnagar, Ghulewadi, Sangamner – 422608



(An Autonomous Institute Approved by AICTE and Affiliated to SPPU, Pune)

Curriculum Structure & Syllabus of S.Y.B.Tech. Artificial Intelligence & Data Science



(Effective from Academic Year 2026-27)

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



**Chairman
Academic Council**

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**INDEX**

1.	Institute Vision	iii
2.	Institute Mission	iii
3.	Department Vision	iii
4.	Department Mission	iii
5.	EOMS Policy	iv
6.	Knowledge and Attitude Profile	iv
7.	Program Outcomes (POs)	vi
8.	Program Education Objectives (PEOs)	vii
9.	Program Specific Outcomes (PSOs)	vii
10.	Preface	viii
11.	Program Codes and Subject Code Policy	ix
12.	Guidelines for Examination Scheme	x
13.	Syllabus Structure – Semester III	xiii
14.	Syllabus Structure – Semester IV	xiv
15.	Syllabus Design Team	xv
16.	Data Structures	1
17.	Software Engineering	5
18.	Object Oriented Programming	9
19.	Foundational Mathematics for AI & DS	13
20.	Entrepreneurship Development	18
21.	Universal Human Values & Professional Ethics	25
22.	Data Structures Lab	29
23.	Object Oriented Programming lab	40
24.	Community Engagement Project	52
25.	Artificial Intelligence	61
26.	Data Science	65
27.	Database management Systems	69
28.	Embedded System & Internet of Things	73
29.	Artificial Intelligence & Data Science Lab	78
30.	Modern Indian Language and Technical Writing	88
31.	Technology Commercialization and Startup Development	91
32.	Database management Systems Lab	95
33.	Embedded System & Internet of Things Lab	104
34.	Sustainable Development	111

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**Chairman
Academic Council**

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



Abbreviation

AEC - Ability Enhancement Course

AICTE - All India Council for Technical Education

CEP - Community Engagement Project

EEM - Entrepreneurship/Economics/Management Courses

MDM - Multidisciplinary Minor

MOOC - Massive Open Online Course

NEP - National Education Policy

NPTEL - National Programme on Technology Enhanced Learning

OEC - Open Elective Course

PCC - Program Core Course

CCC – Cocurricular Course

PEO - Program Educational Objectives

PSO - Program Specific Outcomes

SWAYAM - Study Webs of Active-learning for Young Aspiring Minds

UGC - University Grants Commission

VEC - Value Education Course

VSE - Vocational and Skill Enhancement Course

WK - Knowledge and Attitude Profile

CCE - Comprehensive and Continuous Evaluation

ISO - International Organization for Standardization

EOMS - Educational Organization Management System

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

Chairman

Academic Council
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



Institute Vision

To create opportunities for rural students to become able engineers and technocrats through continual excellence in engineering education.

Institute Mission

Our mission is to create self-disciplined, physically fit, mentally robust and morally strong engineers and technocrats with a high degree of integrity and sense of purpose who are capable of meeting the challenges of ever advancing technology for the benefit of mankind and nature.

We, the management, the faculty and staff, therefore promise to strive hard and commit ourselves to achieve this objective through a continuous process of learning and appreciation of the needs of time.

Department Vision

To empower learners to become experts in Artificial Intelligence and Data Science through persistent excellence in learning and fostering innovations in a rapidly evolving technological world.

Department Mission

M1: To prepare graduates for successful careers in Artificial Intelligence and Data Science through a continuous process of learning and creating opportunities in the AI world.

M2: To cultivate innovations, ethical responsibilities and interdisciplinary collaborations among the students to address real-world problems and contribute to the betterment of society.

M3: We, the faculty and the staff strive hard to strengthen students with a strong academic background, experiential learning, effective communication skills, self-discipline and moral values to thrive in the ever-challenging computing world.

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

Chairman

Academic Council
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**ISO 21001:2018 - EOMS Policy**

We, at AVCOE Sangamner, are committed to:

- Provide quality technical education to the rural students
- Provide a conducive environment for all round development of the students.
- Develop as a premier institute for technical education & research as per the needs and expectations of all stakeholders.
- Comply with all applicable requirements.
- Develop students into technocrats having social responsibility.
- Keep abreast about the advances in technology and innovation.
- Enhance industry-institute interaction.
- Managing intellectual property.
- Maintain hygiene, health and safety for applicable requirements.
- Achieve continual improvement in academic and administrative processes.
- We shall strive to maintain an Educational Organizational Management System.

Knowledge and Attitude Profile

A Knowledge and Attitude Profile (KAP), often represented as WK (Knowledge and Attitude Profile) in some contexts, is a framework or assessment tool used to evaluate an individual's knowledge and attitudes related to a specific area, topic, or domain.

WK1	A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.
WK2	Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.
WK3	A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.
WK4	Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.
WK5	Knowledge, including efficient resource use, environmental impacts, whole-life cost, re-use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**Chairman
Academic Council**

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



K6	Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.
WK7	Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.
WK8	Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.
WK9	Ethics, inclusive behaviour and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**Chairman
Academic Council**

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**Program Outcomes (POs)**

PO1	Engineering Knowledge	Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.
PO2	Problem Analysis	Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4)
PO3	Design/ Development of Solutions	complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)
PO4	Conduct Investigations of Complex Problems	Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).
PO5	Engineering Tool Usage	Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)
PO6	The Engineer and The World	Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).
PO7	Ethics	Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)
PO8	Individual and Collaborative Team work	Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.
PO9	Communication	Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences
PO10	Project Management and Finance	Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.
PO11	Life-Long Learning	Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**Chairman
Academic Council**

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



Program Educational Objectives (PEOs):

PEO1: To develop competent graduates with a strong foundation in Artificial Intelligence and Data Science methodologies and algorithms, capable of solving complex real-world problems by applying appropriate modern tools and techniques to design and develop effective solutions.

PEO2: To exhibit leadership qualities of graduates by successfully guiding teams, managing projects, and leveraging communication skills to create a positive impact on organizations and society.

PEO3: To adopt a lifelong learning approach by consistently enhancing knowledge and skills to remain at the forefront of the rapidly evolving field, while adapting to new technologies, addressing challenges with environmentally sustainable solutions, and upholding ethical practices.

Program Specific Outcomes (PSOs):

A graduate of the Artificial Intelligence & Data Science Program will demonstrate-

PSO1: Apply the fundamental principles of Computer Science, Artificial Intelligence and Data Science to develop programming skills and implement computing solutions using modern software tools and technologies.

PSO2: Analyze, apply and solve complex real-world problems using engineering knowledge to develop innovative and sustainable solutions.

PSO3: Demonstrate innovation, ethics, and lifelong learning to use emerging technologies for career development and entrepreneurship.

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**Chairman
Academic Council**

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



Preface

It is a matter of great pride and satisfaction to introduce the autonomous syllabus for the Second Year B.Tech. programme in Artificial Intelligence & Data Science offered by the department. The curriculum has been thoughtfully developed to address the evolving demands of industry, research, higher education, competitive examinations, and emerging technological trends while strengthening the core academic competencies of students.

The syllabus is designed with a strong emphasis on conceptual clarity, practical exposure, analytical thinking, innovation, and problem-solving skills. It incorporates contemporary and industry-oriented subjects that encourage students to explore advanced domains of Artificial Intelligence, Data Science, and interdisciplinary technologies. The curriculum also promotes outcome-based education, experiential learning, project-based activities, and skill-oriented practices to prepare students for professional excellence and continuous learning.

I express my sincere gratitude to all faculty members, academic experts, industry professionals, and members of the Board of Studies for their valuable guidance, support, and contributions in designing this curriculum. Their collective efforts have helped create a balanced and forward-looking syllabus aligned with academic and industrial expectations.

I am confident that this autonomous curriculum will help students develop strong technical expertise, ethical values, research aptitude, and professional competence required to excel in the rapidly growing field of Artificial Intelligence & Data Science.

I extend my heartfelt best wishes to all students and faculty members for the effective implementation and successful execution of this curriculum.

Dr. A. R. Panhalkar

Head, Department of Artificial Intelligence & Data science

Amrutvahini College of Engineering, Sangamner

(An Autonomous Institute affiliated to Savitribai Phule Pune University, Pune)

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**Chairman
Academic Council**

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

Program codes:

The institute has assigned a unique program code to each branch.

Table 1 – Program codes

Sr. No.	Program code	Code
UG Programs		
1	Artificial Intelligence & Data science	AI

Subject code nomenclature:

The following nomenclature demonstrates the subject codes:

R-Year (of designing/revising syllabus) – **Program code** (as given in table 1) - **Course Type** (as per NEP) – **Course Number** (101 – FY B. Tech Subject 1, 201 - SY B. Tech Subject 1, 301 - TY B. Tech Subject 1, 401 - B. Tech Subject 1 etc.)



Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



Chairman

Academic Council

Amrutvahini College of Engineering, Amru
Sangamner-422608



Guidelines for Examination Scheme:

The evaluation of students shall be based on continuous academic performance in a course based on Comprehensive Continuous Evaluation (CCE) and End Semester Examination (ESE). Components of the Examination System:

1. Comprehensive Continuous Evaluation (CCE):

- CCE of 40 marks based on all the Units of course.
- To design a Comprehensive Continuous Evaluation scheme for a theory subject of 40 marks with the specified parameters, the allocation of marks and the structure can be detailed as follows:

Table 3: Allocation of marks & structure

Sr. No.	Examination	Conduction Marks	Marks	Coverage of Units
1	CCE –I (Class test – 1/Activities)	32	16	Units 1 & Unit 2 (8 Marks/Unit)
2	CCE –II (Assessment tools: Case Study, presentation, Activities, Open book test/Coding Challenges etc.)	32	16	Units 3 & Unit 4 (8 Marks/Unit)
3	CCE –III (Assessment tools: Assignments, seminar, quiz, course project, field visit/Coding Test etc.)	08	08	Unit 5

The distribution of marks across various assessment tools and their mapping to COs is detailed in the table below.

Table 4: Allocation of marks & structure

EXAM	Assessment tool	CO1	CO2	CO3	CO4	CO5
	CCE1	08	08			
	CCE2			08	08	
	CCE3					8

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

**Chairman
Academic Council**

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



CCE				Subject teacher should select any two of the assessment tools for conducting CCE2 & CCE3 (Assignments, seminar, quiz, course project, field visit, Case study, Open book test)		
	60	12	12	12	12	12
Total	100	20	20	20	20	20

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

Chairman**Academic Council**

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



1. Timeline for conduction of CCE:

- Weeks 1- 4: Completion of Unit 1 and 2
- Week 5: Conduct CCE1 1 (16 marks)
- Weeks 6 - 8: Completion of Unit 3 and 4
- Week 9: Conduct CCE2 as (16 marks)
- Weeks 10 -12: Completion of Unit 5
- Week 13: Conduct Seminar/Course project/Presentation/Quiz/Open Book test (8 marks)

Evaluation and Feedback:

- **Unit Test:** Student answer papers will be evaluated promptly, and constructive feedback will be provided highlighting strengths and areas for improvement.
- **Assignments / Case Study:** Student submissions will be assessed as per the given rubric, and meaningful feedback will be provided to help students understand their performance and scope for enhancement.
- **Seminar Presentation:** Students will be evaluated based on content accuracy, delivery, and responses during the Q&A session. Feedback will be shared on communication skills and conceptual understanding.
- **Open Book Test:** Evaluation will focus on the depth of analysis and the ability to apply concepts. Feedback will emphasize students' critical thinking and problem-solving skills.
- **Mini Project:** Students will be assessed on problem identification, design approach, implementation quality, and documentation. Innovation, teamwork, and practical application of concepts will be strongly considered.
- **Field Visit:** Evaluation will be based on the student's report and reflection, focusing on understanding of real-world practices and their relevance to the course outcomes. Marks will reflect clarity, insight, and active participation.
- **Activities:** Students will participate in discussions and reflective activities on human values, ethics, social responsibility, and real-life situations related to Universal Human Values (UHV). Evaluation will be based on active participation, clarity of thought, self-reflection, and the ability to relate concepts to personal and professional life experiences.
- **Coding Challenges:** Evaluation will focus on problem-solving and coding skills using platforms such as Code Chef, Hacker Rank, and Leetcode, with feedback on logical thinking, debugging, and optimization.

2. End Semester Examination (ESE):

The ESE is a compulsory examination for all students. The ESE examination will be conducted at the end of the semester by the institute Department of Examination and Evaluation (DEE). The ESE will be based on 100% syllabus (Unit 1 to Unit 5). The ESE shall be of 60 marks (12 Marks / Unit) conducted for all theory courses as mentioned in the programme structure.

Chairman

BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

Chairman

Academic Council

Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

Syllabus Structure

Course Code	Course Type	Name of the Subject	Teaching Scheme (Hrs./week)					Examination Scheme and Marks						Credit			
			T H	P R	T U T	SL	Tot al	C C E	ES E	T W	P R	O R	TO TA L	T H	P R	T U T	T O T A L
Semester-III																	
R25-AI-PCC-201	Program Core course - 1	Data Structures	3	-	-	45	90	40	60	-	-	-	100	3	-	-	3
R25-AI-PCC-202	Program Core course - 2	Software Engineering	3	-	-	45	90	40	60	-	-	-	100	3	-	-	3
R25-AI-PCC-203	Program Core course - 3	Object Oriented Programming	3	-	-	45	90	40	60	-	-	-	100	3	-	-	3
R25-AI-MDM-204	Multidisciplinary Course-1	Foundational Mathematics for AI & DS	3	-	-	45	90	40	60	-	-	-	100	3	-	-	3
R25-AI-OEC-205	Open Elective - 1	Open Elective-I	2	-	-	30	60	40	60	-	-	-	100	2	-	-	2
R25-AI-EEM-206	Entrepreneurship/Economics/ Management course	Entrepreneurship Development	1	2	-	15	60	25	-	25	-	-	50	1	1	-	2
R25-AI-VEC-207	Value Education Course	Universal Human Values & Professional Ethics	2	-	-	30	60	50	-	-	-	-	50	2	-	-	2
R25-AI-PCC-208	Program Core Course- Lab	Data Structures Lab	-	2	-	-	30	-	-	25	25	-	50	-	1	-	1
R25-AI-PCC-209	Program Core Course Lab	Object Oriented Programming lab	-	2	-	-	30	-	-	25	25	-	50	-	1	-	1
R25-AI-CEP-210	Community Engagement Project / Field Project	Community Engagement project	-	4	-	-	60	-	-	50	-	-	50	-	2	-	2
		Total	17	10	0	255	660	275	300	125	50	0	750	17	5	0	22



Chairman
BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



Chairman
Academic Council
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



Semester-IV																	
Course Code	Course Type	Name of the Subject	Teaching Scheme (Hrs./week)					Examination Scheme and Marks						Credit			
			TH	PR	TU	SL	Total	CEE	ESE	TW	PR	OR	TOTAL	TH	PR	TUT	TOTAL
R25-AI-PCC-211	Program Core course - 4	Artificial Intelligence	3	-	-	45	90	40	60	-	-	-	100	3	-	-	3
R25-AI-PCC-212	Program Core course - 5	Data Science	2	-	-	30	60	40	60	-	-	-	100	2	-	-	2
R25-AI-PCC-213	Program Core course - 6	Database management Systems	3	-	-	45	90	40	60	-	-	-	100	3	-	-	3
R25-AI-MDM-214	Multidisciplinary Course-2	Embedded System & Internet of Things	2	-	-	30	60	40	60	-	-	-	100	2	-	-	2
R25-AI-OEC-215	Open Elective - 2	Open Elective-II	2	-	-	30	60	40	60	-	-	-	100	2	-	-	2
R25-AI-VSC-216	Vocational Skill Course	AI & DS Lab	-	4	-	-	60	-	-	25	25	-	50	-	2	-	2
R25-AI-AEC-217	Ability Enhancement Course	Modern Indian Language & Technical Writing	1	2	-	-	60	-	-	50	-	-	50	1	1	-	2
R25-AI-EEM-218	Entrepreneurship/Economics/ Management course	Technology Commercialization & Startup	2	-	-	30	60	50	-	-	-	-	50	2	-	-	2
R25-AI-PCC-219	Program Core Course-Lab	Database management Systems Lab	-	2	-	-	30	-	-	-	25	-	25	-	1	-	1
R25-AI-PCC-220	Program Core Course Lab	Embedded System & Internet of Things Lab	-	2	-	-	30	-	-	25	-	-	25	-	1	-	1
R25-AI-VEC-221	Value Education Course	Sustainable Development	2	-	-	30	60	50	-	-	-	-	50	2	-	-	2
			17	10	0	240	660	300	300	100	50	0	750	17	5	0	22

Chairman
Chairman
 BoS, Artificial Intelligence & Data Science
 Amrutvahini College of Engineering, Amrutnagar
 Sangamner-422608

Chairman
Chairman
 Academic Council
 Amrutvahini College of Engineering, Amrutnagar
 Sangamner-422608



Department of Artificial Intelligence & Data Science

S.Y.B.Tech Syllabus Design Team(2026-27)

Course Code	Course Type	Subject	Faculty Co-ordinator	Supporting Faculty
S.Y. B. Tech 2026-27 (SEM III)				
R25-AI-PCC-201	Program Core course - 1	Data Structures	Er. J. R. Tambe	Er. R. R. Tribhuvan
R25-AI-PCC-202	Program Core course - 2	Software Engineering	Dr. A. R. Panhalkar	Er. N. A. Rahane
R25-AI-PCC-203	Program Core course - 3	Object Oriented Programming	Dr. A. R. Panhalkar	Er. R. R. Tribhuvan
R25-AI-MDM-204	Multidisciplinary Course-1	Foundational Mathematics For AI&DS	Er. D. N. Kalokhe Er. S. S. Deshmukh	Er. J. R. Tambe
R25-AI-OEC-205	Open Elective - 1	Road Safety&Civil Responsibility	Er. S. G. Dighe	Er. V. A. Naikwadi
R25-AI-EEM-206	Entrepreneurship/Economic s/ Management course	Entrepreneurship Development	Er. N. A. Rahane	Er. S. C. Patil
R25-AI-VEC-207	Value Education Course	Universal Human Values & professional ethics	Er. J. R. Tambe	Er. S. P. Shinde
R25-AI-PCC-208	Program Core Course- Lab	Data Structures Lab	Er. J. R. Tambe	Er. R. R. Tribhuvan
R25-AI-PCC-209	Program Core Course Lab	Object Oriented Programming Lab	Er. S. P. Shinde	Er. N. A. Rahane
R25-AI-CEP-210	Community Engagement Project / Field Project	Community engagement project	Er. R. B. Pandit	Er. S. P. Shinde
S.Y. B. Tech 2026-27 (SEM IV)				
R25-AI-PCC-211	Program Core course - 4	Artificial Intelligence	Er. R. B. Pandit	Er. S. C. Patil
R25-AI-PCC-212	Program Core course - 5	Data Science	Er. S. N. Navale	Er. R. B. Pandit
R25-AI-PCC-213	Program Core course - 6	Database Management System	Er. S. P. Shinde	Er. N. A. Rahane
R25-AI-MDM-214	Multidisciplinary Course - 2	Embedded System & Internet of Things	Er. S. G. Dighe	Er. S. C. Patil
R25-AI-OEC-215	Open Elective -2	Digital Marketing & E-business	Er. S. C. Patil	Er. S. N. Navale
R25-AI-VSC-216	Vocational Skill Course	AI&DS lab	Er. R. S. Pandit	Er. S. N. Navale

Chairman
BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

Amrutvahini College of Engineering, Sangamner
Department of Artificial Intelligence & Data Science

R25-AI-AEC-217	Ability Enhancement Course	Modern Indian Language for professional skill development	Er. V. A. Naikwadi	Er. S. G. Dighe
R25-AI-LEM-218	Entrepreneurship/Economic s/ Management course	Technology Commercialization & Startup	Er. N. A. Rahane	Er. R. B. Pandit
R25-AI-PCC-219	Program Core Course- Lab	AI & DS Lab	Er. R. B. Pandit	Er. S. N. Navale
R25-AI-MDM-220	Multidisciplinary Course-Lab	Embedded System & Internet of Things Lab	Er. S. C. Patil	Er. S. G. Dighe
R25-AI-VEC-221	Value Education Course	Sustainable Development	Er. V. A. Naikwadi	Er. N. A. Rahane

Er. J. R. Tambe
Coordinator

Dr. A. R. Panhalkar
H.O.D

Dr. M. A. Venkatesh
Principal

Chairman
Academic Council
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



Semester-III

Chairman
BoS, Artificial Intelligence & Data Science
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608

Chairman
Academic Council
Amrutvahini College of Engineering, Amrutnagar
Sangamner-422608



Program:	S. Y. B. Tech			Semester:	III		
Course:	Data Structures			Course Code:	R25-AI-PCC-201		
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
03	-	-	03	40	60	-	-
Course Description: This course introduces the fundamental concepts of Data Structures and their role in efficient data organization, storage, and processing. It covers linear and non-linear data structures along with algorithm analysis and practical implementation. The course emphasizes problem-solving techniques and applications relevant to programming, technical interviews, and competitive examinations.							
Course Relevance: This course aligns with industry and academic requirements by introducing core data structures and algorithmic thinking used in software development and system design. It prepares students for technical interviews, competitive programming, and exams such as GATE by strengthening logical thinking, coding ability, and problem-solving skills.							
Prerequisite: 1. Programming & Problem Solving 2. Programming in C							
Bridge Content: Basics of Programming (C/C++/Python)							
Course Objectives: 1. To Develop an understanding of data organization techniques and the role of data structures in efficient computation. 2. To Build programming skills for implementing fundamental data structures and performing operations on them. 3. To Analyse the performance of algorithms using time and space complexity concepts. 4. To Design appropriate data structure solutions for different computational problems. 5. To Strengthen logical thinking and coding ability required for software development, competitive programming and technical interviews.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome					Bloom's Level	
CO1	SELECT the data structure that minimizes complexity and maximizes processing speed.					4	
CO2	APPLY stacks and queues for sequential task scheduling and to balance orderly processing					3	
CO3	IMPLEMENT linked lists to develop solutions for real world problems.					3	
CO4	COMPARE outcome of various collision resolution techniques while implementing hash tables to manage and retrieve data efficiently.					4	



CO5	APPLY tree and graph structures to solve complex structured and hierarchical data problems.	3
Course Contents		
Unit 1	Introduction to Data Structures & Algorithms	9 Hrs. CO1
Need of Data structure, Definition, Classification/Types of Data Structures Algorithm design strategies, Divide and Conquer Searching: Sequential Search, Binary Search (Recursive & Non-recursive) Sorting: Bubble Sort, Selection Sort, Insertion sort, Merge sort, Quick sort, Radix sort Complexity of algorithm: 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.		
Exemplars / Practical Applications and usage: E-commerce Product Sorting using Divide and Conquer strategy		
Unit 2	Stack & Queue	9 Hrs. CO2
Stack: Real world connection, Introduction, Representation of stack Using Sequential Organization, stack operations, Applications of Stack, Expression Evaluation and Conversion. Recursion- concept. Queue: Real world connection, Introduction of Queue, Representation of Queue using Sequential organization. Queue Operations. Circular Queue and its advantages, Deque-introduction, Priority Queue.		
Exemplars / Practical Applications and usage: Use of stack in backtracking, Job scheduling using priority queue.		
Unit 3	Linked organization: Linked List	9 Hrs. CO3
Linked List: Real-World Connection, program memory layout Introduction of Linked Lists with its storage representations, Types of Linked List: singly linked, linear and Circular Linked Lists, Doubly Linked List, Doubly Circular Linked List, Dynamic memory management function in C/C++, pointers, Types of pointers, Primitive Operations on Linked List-Create, Traverse, Search, Insert, Delete, Sort, Concatenate. Polynomial Manipulations-Polynomial addition using Linked list. Generalized Linked List (GLL): concept.		
Exemplars / Practical Applications and usage: Concept of Garbage collection, Growing employee database dynamically using LL.		
Unit 4	Hashing	9 Hrs. CO4
Hash Table: Real world connection, Concepts-hash table, hash function, basic operations, bucket, collision, probe, synonym, overflow, open hashing, closed hashing, perfect hash function, load density, full table, load factor, rehashing, properties of good hash function, Collision resolution strategies- Open Addressing (Linear Probing, Quadratic Probing, Double Hashing), Separate Chaining, Rehashing, Extendible Hashing.		
Exemplars / Practical Applications and usage: Using Cryptographic Hashing for Password Security and File Integrity Verification		
Unit 5	Non-Linear Data Structures: Trees and Graphs	9 Hrs. CO5



Tree: Real world Connection, Introduction of tree, Basic terminology (root, node, leaf, height, etc.) types of trees, Representations, Traversals (Recursive, On-recursive)- inorder, preorder, postorder, Binary tree, Binary search tree, BST operations, Concept of Threaded Binary Tree, Optimal Binary Search Tree (OBST), Height Balanced Tree-AVL tree, Heap Tree

Graph: Real world Connection, Introduction of graph, storage representation, Adjacency matrix, adjacency list, DFS, BFS, Greedy strategy, Minimum spanning Tree - Prims and Kruskal Algorithms

Exemplars / Practical Applications and usage: GPS/Navigation system that models a city map as a weighted graph and applies core graph algorithms, ZIP/GZIP file compression using frequency-based encoding using Huffman tree

Textbooks:

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, "Fundamentals of Data Structures in C", Publisher - Universities Press, 2nd Edition, 2008, ISBN-13: 978-8173716058, ISBN-10: 8173716056.
2. Reema Thareja, "Data Structures Using C", 2nd Edition, Oxford University Press, ISBN-13: 978-0-19-809930-7 ISBN-10: 0-19-809930-4

Reference Books:

1. Steven S. Skiena, "The Algorithm Design Manual", Springer, 2nd edition, ISBN: 978-1-84800-069-8
2. Yashavant Kanetkar, "Let Us C", 8th Edition, BPB Publications, ISBN: 9788183331777
3. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", 2nd Edition, Pearson Education, ISBN: 978-8177583588
4. Aaron M. Tenenbaum, "Data Structures Using C", 2nd Edition, Pearson Education, ISBN: 9788131711484
5. Varsha H. Patil, "Data Structures Using C++", Oxford University Press, 2nd edition, ISBN: 978-0198066231

MOOCs Links and additional reading material:

1. First Mile Foundation Program-Programming and Data Structure and Algorithm
<https://nptel.ac.in/courses/106106133>
2. Introduction to Data Structures and Algorithms
<https://nptel.ac.in/courses/106102064>
3. Data Structures and Program Methodology
<https://nptel.ac.in/courses/106103069>

CO-PO-PSO Mapping

CO	PO1 EK	PO2 PA	PO3 DD	PO4 CPI	PO5 ETU	PO6 EW	PO7 Eth	PO8 ITW	PO9 Com	PO10 PMF	PO11 LL	PSO1 PFS	PSO2 PSS	PSO3 SCE
CO1	2	2	2	2	-	-	-	-	-	-	2	2	2	2
CO2	2	2	2	2	-	-	-	-	-	-	2	2	2	2
CO3	2	2	2	2	-	-	-	-	-	-	2	2	2	2
CO4	2	2	2	2	-	-	-	-	-	-	2	2	2	2
CO5	2	2	2	2	-	-	-	-	-	-	2	2	2	2
Avg.	2	2	2	2	-	-	-	-	-	-	2	2	2	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**Unit wise Weightage:**

Exam	Unit1	Unit2	Unit3	Unit4	Unit5
CCE					
CCE1	8	8			
CCE2			8	8	
CCE3					8
ESE	12	12	12	12	12
Total	20	20	20	20	20

Assessment Pattern

BT Level	CCE			ESE				
	CCE1	CCE2	CCE3					
CO				CO1	CO2	CO3	CO4	CO5
L1								
L2	√	√	√	√	√	√	√	√
L3	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√
L5								
L6								

This course serves as a prerequisite for / maps with the following future courses:

1. Database Management Systems
2. Design & Analysis of Algorithm
3. Operating Systems
4. Artificial Intelligence & Data Science

Job Mapping:

Job opportunities that one can get after learning this course: Software Developer, Backend Developer, Data Engineer, Machine Learning Engineer. These roles frequently use arrays, trees, graphs, hashing, and algorithms in real-world systems.



Program:	S. Y. B. Tech			Semester :	III		
Course:	Software Engineering			Course Code:	R25-AI-PCC-202		
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
03	-	-	03	40	60	-	-
Course Description: The course presents the basic principles and methodologies used in software engineering, including software development processes, requirement specification, design engineering, project planning, testing methods, Agile and modern software development approaches.							
Course Relevance: This course provides essential knowledge and practical skills required for the systematic development of software systems. It enables students to apply software engineering principles for developing reliable applications such as web, mobile, desktop, and enterprise software.							
Prerequisite: 1. Programming & Problem Solving 2. Emerging Trends in AI&DS							
Bridge Content: Basic concepts of programming							
Course Objectives: 1. To Study software engineering principles, software process frameworks, Agile methods and other various development models used in software development. 2. To Acquaint students with requirement engineering techniques for identifying system requirements and preparing Software Requirement Specification (SRS) documents. 3. To Study software design concepts and object-oriented analysis and design techniques for modelling software systems. 4. To Study the fundamentals of software project planning, estimation, scheduling, and risk management. 5. To Understand software testing techniques and modern software engineering practices.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome					Bloom's Level	
CO1	ANALYZE appropriate software process models and approaches for development of software projects					4	
CO2	APPLY requirement engineering concepts to gather, organize, and document system requirements for software systems.					3	
CO3	DEMONSTRATE the use of design concepts and UML diagrams to develop software system models.					3	
CO4	DEVELOP project plans, perform effort estimation, and create project schedules using methods and tools.					3	
CO5	APPLY software testing techniques and modern software engineering practices to develop modern software.					3	
Course Content							



Unit 1	Introduction To Software Engineering	8 Hrs.	CO1
Software and Software Engineering Software Process Framework, Software Development Life Cycle, Umbrella Activities, Software Development Myths, Generic Process Model, Waterfall Model, Incremental (SDLC) Process Models, Spiral Model, Concurrent Models, Agile software development: Agility, Agile process, Agile principles, plan driven and agile development. Scrum, Extreme Programming, Kanban, Lean			
Exemplars / Practical Applications/ Case Study:			
1. A hospital hires a software company to develop a Healthcare Management System that manages patient records, appointments, billing, and telemedicine services. Discuss with reasons which models are suitable for it. 2. Study Agile project management tools such as Jira, Trello, and Azure DevOps. Explain how these tools support backlog management, sprint tracking, and team collaboration.			
Unit 2	Software Requirements Analysis	8 Hrs.	CO2
Functional and non-functional requirements, User Requirements, System Requirements, Requirements Engineering Tasks: Elicitation, Elaboration, Negotiation, Specification, and Validation. Requirement Elicitation Techniques, Establishing the Groundwork: Identifying Stakeholders, Recognizing Multiple Viewpoints, Working toward Collaboration, Asking the First Questions. Developing Use Cases: Building the Analysis Model, Elements of the Analysis Model, Software Requirement Specification (SRS) documentation.			
Exemplars / Practical Applications and usage:			
Use Agile Tool- JIRA for requirement management, sprint planning, task tracking, and collaboration.			
Unit 3	Design Engineering	8 Hrs.	CO3
Design Concepts: Abstraction, Architecture, Patterns, Modularity, Information Hiding, Functional Independence, Refinement, Aspects, Refactoring. Cohesion and Coupling Object-Oriented Analysis & Design, Unified Modeling Language (UML): Structural Modeling: Class Diagram, Object Diagram. Behavioral Modeling: Use Case Diagram, Sequence Diagram, Collaboration Diagram, Activity Diagram, State Chart Diagram.			
Exemplars / Practical Applications and usage:			
Use Lucidchart or StarUML to design any software project.			
Unit 4	Project Planning and Scheduling	8 Hrs.	CO4
Project Scheduling: Software project scheduling, Gantt Chart, PERT chart, CPM, Microsoft Projects, and Primavera Project Management Software Project Estimation: Introduction to project planning, Project planning process, Resources, Software Project Estimation, Decomposition Techniques, LOC Based Estimation, COCOMO models and Function Point (FP) Based Estimation. Risk Management: Reactive versus Proactive Risk Strategies, Software Risks, Identification, projection, Refinement, mitigation, RMMM plans).			
Exemplars / Practical Applications and usage:			
1. Demonstrate project planning and scheduling of an E-Commerce Website using Oracle Primavera with task dependencies and resource management. 2. Develop a complete project plan for an Online Shopping System including scheduling, estimation, resource planning, risk identification, and mitigation strategies using project management tools.			



Unit 5	Software Testing and Current Trends in Software Engineering	9 Hrs.	CO5
Software Testing, A Strategic Approach to Software Testing, Test Strategies for Conventional Software, Test Strategies for Object-Oriented Software, Test Strategies for WebApps, Validation Testing, System Testing, Debugging, Defect Life Cycle, Manual and Automated testing. Introduction to DevOps practices and DevOps lifecycle, Continuous Integration and Continuous Deployment (CI/CD) pipelines, DevSecOps and Secure Software Development Lifecycle (SSDLC), Modern software architecture patterns including Microservices, Event-Driven, Serverless and Reactive architectures, AI in Software Engineering(AI-assisted development, automated testing and code review) , cloud-native and distributed software systems, Low-Code and No-Code development platforms, software sustainability and green computing, ethical and responsible software engineering practices. Exemplars / Practical Applications and usage: Study of tools like Selenium, Cypress, Playwright, and Postman with CI/CD pipelines to enable continuous testing and faster software delivery. Textbooks: 1. Roger Pressman, “Software Engineering: A Practitioner’s Approach”, McGraw Hill, ISBN 0– 07– 337597–7 2. Ian Sommerville, “Software Engineering”, Addison and Wesley, ISBN 0-13-703515-2 Reference Books: 1. Carlo Ghezzi, “Fundamentals of Software Engineering”, PHI, ISBN-10: 0133056996 2. Rajib Mall, “Fundamentals of Software Engineering”, PHI, ISBN-13: 978-8120348981 3. Pankaj Jalote, “An Integrated Approach to Software Engineering”, Springer, ISBN 13: 9788173192715. 4. S K Chang, “Handbook of Software Engineering and Knowledge Engineering”, World Scientific, Vol I, II, ISBN: 978-981-02-4973-1 5. Tom Halt, “Handbook of Software Engineering”, Clanye International, ISBN- 10: 1632402939 MOOCs Links and additional reading material: 1. Software Engineering: https://swayam.gov.in/nd1_noc19_cs69/preview 2. Software Engineering: https://swayam.gov.in/nd2_ccc20_cs07/preview			

CO-PO-PSO Mapping

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

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**Unit wise Weightage:**

Exam	Unit1	Unit2	Unit3	Unit4	Unit5
CCE					
CCE1	8	8			
CCE2			8	8	
CCE3					8
ESE	12	12	12	12	12
Total	20	20	20	20	20

Assessment Pattern

BT Level	CCE						ESE				
	CT1	CCE2	CCE3	CCE4			CO1	CO2	CO3	CO4	CO5
CO											
L1											
L2	√	√					√	√	√	√	√
L3	√	√	√				√	√	√	√	√
L4	√	√	√				√	√	√	√	√
L5											
L6											

This course serves as a prerequisite for / maps with the following future courses:

1. Software Testing and Quality Assurance
2. Web Application Development
3. Devops
4. Capstone / Major Project
5. Data Engineering and Big Data Systems

Job Mapping:

Job opportunities that one can get after learning this course: Software Developer / Software Engineer, Software Tester / QA Engineer, System Analyst, Web Developer



Program:	S. Y. B. Tech			Semester:	III		
Course:	Object Oriented Programming			Course Code:	R25-AI-PCC-203		
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
03	-	-	03	40	60	-	-
Course Description: This course provides a comprehensive introduction to Object-Oriented Programming (OOP) concepts and their implementation using the Java programming language. It covers core programming features, memory management, exception handling, multithreading, collections, and database connectivity through Java Database Connectivity.							
Course Relevance: The course equips students with essential Java and object-oriented design skills required in modern software development. It enables students to design scalable, maintainable, and efficient software applications. It prepares learners for careers in application development and advanced Java-based technologies.							
Prerequisite: 1. Programming & Problem Solving 2. Programming in C							
Bridge Content: Fundamentals of programming development.							
Course Objectives: 1. To comprehend object-oriented programming concepts and Java programming fundamentals. 2. To use access control, constructors, arrays, strings, and memory management concepts for developing applications using Java. 3. To utilize inheritance, polymorphism, abstraction, and interfaces for designing reusable, maintainable, and hierarchical programs in Java. 4. To demonstrate the use of packages, exception handling, and multithreading in developing efficient programs using Java. 5. To apply generic programming, collections framework, database connectivity using Java Database Connectivity, and framework concepts in developing real world applications.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome					Bloom's Level	
CO1	APPLY object-oriented programming concepts for developing basic programs in Java.					3	
CO2	DEMONSTRATE the use of constructors, arrays, and memory management techniques to develop efficient programs in Java.					3	
CO3	USE inheritance, polymorphism, abstraction, and interfaces to develop reusable and optimized software.					3	
CO4	APPLY packages, exception handling, and multithreading concepts to design structured and reliable programs.					3	
CO5	ANALYZE generics, collections, basic frameworks and Java-based data science techniques to create modern and robust software applications.					4	



Course Content			
Unit 1	Introduction to Java	8 Hrs.	CO1
Introduction to Procedure oriented and Object-oriented programming need of OOP, characteristics/features of object-oriented programming: objects, classes, message passing, encapsulation, abstraction, information hiding, inheritance, polymorphism, static and dynamic binding. Java as an Object-Oriented Language: The Java buzzwords, Java Virtual Machine (JVM), byte code, Java Runtime Environment (JRE), Java Development Kit (JDK), Java editors. Java program structure, compilation and execution process, comments Classes and Objects: Defining classes, field declaration, method declaration and definition, object creation, accessing class members, Access Control: Access specifiers (public, private, protected, default), datatypes, variables, control statements, argument passing, object as a parameter, returning objects, Input and Output: Byte Stream vs character stream, writing console output, use of scanner class. Exemplars / Practical Applications/ Case study: Study the features such as code assistance, debugging support, plugin availability, framework integration, and ease of development to determine which IDE (IntelliJ IDEA, Eclipse IDE, Apache NetBeans, and Visual Studio Code) best supports efficient Java application development.			
Unit 2	Constructors and Arrays	8 Hrs.	CO2
Constructors: Concept, Characteristics of constructors, types of constructors, constructor overloading, use of constructors in object initialization. this keyword, static data members and static methods, scope and lifetime of variables. Arrays: One-dimensional and multidimensional arrays, passing arrays to methods, array of objects, variable-length argument lists, command-line arguments, Strings: String class, StringBuffer, StringBuilder Memory Management in Java: new keyword, garbage collection mechanism, finalize () method. Exemplars / Practical Applications/ Case study: Explain how to use Project Lombok (Lombok) a Java library to reduce boilerplate code in Java programs.			
Unit 3	Inheritance and Polymorphism	8 Hrs.	CO3
Inheritance: Concept, need, types of inheritance, benefits and limitations, use of super keyword, constructors in inheritance, Creating a Multilevel Hierarchy, final variables and methods, final classes, Polymorphism: Compile-time and runtime polymorphism, method overloading, role of polymorphism in reusable and maintainable software design. method overriding, dynamic method dispatch, Abstraction: Abstract classes, abstract methods, Interfaces for Multiple Inheritance: Multiple inheritance concept, defining interfaces, extending interfaces, implementing interfaces, accessing interface variables, Functional interface, Lambda Expression Exemplars / Practical Applications/ Case study: Explain sealed class and its features with examples to provide better control over class hierarchies.			
Unit 4	Packages, Exception Handling and Multithreading	8 Hrs.	CO4
Packages: Java API packages. creating packages, accessing packages, adding classes to packages Exception handling: Errors and exceptions, types of exceptions, exception-handling fundamentals, try, catch, multiple catch blocks, nested try blocks, finally block, throw and throws keywords, user-defined exceptions, common built-in exceptions. MultiThreading: Processes and Threads, Java Thread model, main thread, creating a thread, creating multiple threads, Using isAlive() and join(), Thread priorities, synchronization, suspending, resuming, and stopping threads.			

**Exemplars / Practical Applications/ Case study:**

Demonstrate how Java Virtual Threads (Project Loom) can be used to handle thousands of concurrent tasks efficiently, without the overhead of traditional threads.

Unit 5	Introduction to Advanced Java	8 Hrs.	CO5
Generic Programming: Introduction to generics, advantages of generics, generic classes and methods. Collections Framework: Collection hierarchy, List interface, set interface, Map interface, collection classes – ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap, iterators. Database Connectivity: Basics of JDBC, connecting java with MYSQL/POSTGRES SQL, crud operations using JDBC, introduction to prepared statements Framework: Introduction, framework vs library, advantages of using frameworks, overview of popular java frameworks Java in Data Science: Overview, Reading and Writing Data (CSV, JSON), Basic Statistical Operations using Apache Commons Math, Introduction to JavaFX for Data Visualization			
Exemplars / Practical Applications/ Case study: Develop a cloud-ready REST API to manage employee data using Spring Boot for demonstrating modern Java development practices, dependency injection, and microservice-ready architecture.			
Textbooks: 1. E Balaguruswamy, (2023). Programming with JAVA: A Primer. 7th edition. India: McGraw Hill Education 2. Herbert Schildt, (2021). Java: The complete reference, 13th edition. McGraw-Hill Education.			
Reference Books: • Paul Deitel and Harvey Detail, Java: How to Program, Pearson's Publication, 9th Edition. • Horstmann, C. S. (2023). Core Java - Vol. I – Fundamentals (Vol. 12). Pearson Education.			
MOOCs Links and additional reading material: 1. Programming In Java: https://onlinecourses.nptel.ac.in/noc25_cs57/preview			

CO-PO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	2	2	2	-	-	-	-	-	3	2	2	3
CO2	2	3	2	2	2	-	-	-	-	-	3	3	3	3
CO3	2	3	2	2	2	-	-	-	-	-	3	3	3	3
CO4	2	3	2	2	2	-	-	-	-	-	3	3	3	3
CO5	2	3	3	3	3	-	-	-	-	-	3	3	3	3
Avg.	2	3	2	2	2	-	-	-	-	-	3	3	3	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Unit wise Weightage:

Exam	Unit1	Unit2	Unit3	Unit4	Unit5
------	-------	-------	-------	-------	-------



CCE					
CCE1	8	8			
CCE2			8	8	
CCE3					8
ESE	12	12	12	12	12
Total	20	20	20	20	20

Assessment Pattern

BT Level	CCE						ESE				
	CCE1	CCE2	CCE3	CCE4							
CO	CO1						CO1	CO2	CO3	CO4	CO5
L1											
L2	√	√	√		√	√	√	√	√	√	√
L3	√	√	√		√	√	√	√	√	√	√
L4			√						√		
L5											
L6											

This course serves as a prerequisite for / maps with the following future courses:

1. Machine Learning
2. Web Technologies
3. Major Project

Job Mapping:

Job opportunities that one can get after learning this course: Java Developer, Software Developer, Application Developer, System Programmer, Junior Software Engineer



Program:		S. Y. B. Tech		Semester:		III	
Course:		Foundational Mathematics for AI&DS		Course Code:		R25-AI-MDM-204	
Teaching Scheme (Hrs./Week)				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
3	--	--	03	40	60	--	--
Course Description: This course introduces the mathematical foundations required for Artificial Intelligence and Data Science. It covers sets, logic, functions, statistics, probability, hypothesis testing, and graph theory concepts. The course emphasizes analytical problem-solving and computational applications. It also provides the basis for advanced topics in machine learning, data analytics, and algorithm design.							
Course Relevance: The course provides mathematical foundations required in Artificial Intelligence and Data Science. It helps students develop logical reasoning and analytical thinking skills. Concepts of statistics and probability are widely used in machine learning and predictive analytics. Graph theory and functions are essential for algorithms, data structures, and network analysis.							
Prerequisite: 1. Basic knowledge of higher secondary mathematics. 2. Understanding of algebraic operations and equations. 3. Familiarity with elementary probability and statistics concepts. 4. Logical thinking and analytical problem-solving ability.							
Bridge Content: • Revision of basic algebra and arithmetic concepts. • Introduction to logical reasoning and mathematical notation. • Basics of graphs, probability, and statistical interpretation. • Practice on problem-solving techniques and computational thinking.							
Course Objectives: 1. To understand the concepts of sets, relations, functions, and propositional logic for mathematical problem-solving. 2. To learn statistical techniques and measures for analyzing and interpreting data effectively. 3. To apply probability concepts and probability distributions in uncertain and real-life situations. 4. To understand hypothesis testing methods and statistical tests for decision-making and data validation. 5. To study graph theory concepts and algorithms for solving computational and network-based problems.							
Course Outcomes: On completion of the course, learner will be able to :-							
CO	Course Outcome					Bloom's Level	
CO1	APPLY concepts of sets, relations, logic, and functions to solve mathematical and computational problems.					3	
CO2	ANALYZE and interpret statistical data using measures of central tendency, dispersion, correlation, and regression.					3	
CO3	APPLY probability concepts and distributions to model uncertain events and real-life scenarios.					3	
CO4	PERFORM hypothesis testing using statistical tests such as t-test, Z-test, F-test, and Chi-square test.					3	
CO5	USE graph theory concepts and algorithms to solve network and optimization problems.					3	



Course Content			
Unit 1	Sets and Functions	9 Hrs.	CO1
Sets-Set theory, Axiomatic set theory, Set Operations, Cardinality of set, Types of set Propositional logic, Relations -Relations & Properties, Representations of Relations, Closures of Relations. Equivalence Relations, Hasse Diagram, Lattices, Chains & Anti Chains, Warshalls Algorithm Functions & Types of Functions			
Exemplars / Practical Applications, Usage: An IT Company monitors its cloud servers by tracking API requests and their response times. The system treats API requests as a set and defines a function that maps each request to its response time. i. Collect / Simulate Data (Set Formation) ii. Apply Function Mapping iii. Check Properties of Function iv. Compute Numerical Measures (Using Function Output).			
Unit 2	Statistics	9 Hrs.	CO2
Study of statistical measures including central tendency, dispersion. Moments, skewness, kurtosis, correlation, regression, and interpretation of statistical data.			
Exemplars / Practical Applications, Usage: An IT company evaluates the performance and reliability of its cloud servers by analyzing API response times and success/failure rates. Students learn how statistics and probability are used in real-world performance monitoring and reliability analysis. i. Collect or Simulate Sample Data ii. Compute Statistical Measures iii. Apply Probability Concepts Iv. Use Normal Distribution (Approximation)			
Unit 3	Probability	9 Hrs.	CO3
Introduction to probability including sample space, events, addition and multiplication theorems, conditional probability, independent events, random variables, and basic probability distributions such as Binomial, Poisson, and Normal distribution, Markov inequality and Chebyshev inequality			
Exemplars / Practical Applications, Usage: These probability tools allow engineers to model events, analyze performance, and guarantee reliability under uncertainty in AIDS engineering systems.			
Unit 4	Hypothesis Testing	9 Hrs.	CO4
Introduction, Statistical Hypothesis (Simple and-Composite), Test of a Statistical Hypothesis, Null Hypothesis, Alternative Hypothesis, Critical Region, Type I and Type II errors, level of significance, hypothesis testing using Chi-square Test, t-test, F-test, Z-test			
Exemplars / Practical Applications, Usage: In AIDS (Applied, Information, Data, and Systems) engineering, Engineers test the hypothesis “the new routing algorithm reduces packet loss compared to the old one” by collecting performance data and applying statistical tests (such as t-tests or chi-square) to determine if the observed improvement is statistically significant.			
Unit 5	Trees & Graph Theory	9 Hrs.	CO5



Trees- Introduction, Properties, Decision Trees, Prefix codes and Huffman's coding, Topological ordering Graph terminology and Special types of Graphs, Representation of Graphs, Graph Isomorphism, Connectivity, Euler & Hamilton Paths, The Handshaking Lemma, Single Source shortest Path-Dijkstra's Algorithm, All Pair Shortest distance algorithm- Floyd Warshall's Algorithm

Exemplars / Practical Applications, Usage: In AIDS (Applied, Information, Data, and Systems) engineering, it is particularly useful for analyzing connectivity, optimizing resource allocation, and improving reliability in communication, medical, and computational systems.

Textbooks:

1. Kenneth H. Rosen, Discrete Mathematics and Its Applications, McGraw Hill.
2. Seymour Lipschutz and Marc Lipson, Discrete Mathematics, Schaum's Outline Series.
3. S.C. Gupta and V.K. Kapoor, Fundamentals of Mathematical Statistics, Sultan Chand Publications.
4. Narsingh Deo, Graph Theory with Applications to Engineering and Computer Science, PHI Learning.

Reference Books:

1. John P. Tremblay and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, McGraw Hill.
2. Sheldon Ross, A First Course in Probability, Pearson Education.
3. Richard Johnson and Dean Wichern, Applied Multivariate Statistical Analysis, Pearson.
4. Douglas B. West, Introduction to Graph Theory, Prentice Hall.

MOOCs Links and additional reading material:

1. Set and Function: By Prof. Dipanwita Roychoudhury, IIT Kharagpur.
<https://youtu.be/kLKIPOkSR8?si=snI3XdVseJA3kvC3>
2. Statistics: By Prof. S. Dharmaraja, IIT Delhi. <https://youtu.be/1NPBMF2T8Iw?si=NFIZByxt-xPgKlcm>
3. Probability: By Prof. Prof. S. Dharmaraja, IIT Delhi
<https://youtu.be/Tbdj75ByDIU?si=kUR7ObER7G7F5QiJ>
4. Hypothesis Testing: By Prof. Jitesh J Thakkar, IIT Kharagpur.
<https://youtu.be/6zslEdMGGiQ?si=2YDHzi8QUjmWXwYt>
5. Graph Theory: By Prof. Soumen Maity, IISER, Pune.
<https://youtu.be/E40r8DWgG40?si=RIHwvoGRJnMDzRQQ>

CO-PO-PSO Mapping

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

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**Unit wise Weightage:**

Exam	Unit1	Unit2	Unit3	Unit4	Unit5
CCE					
CCE1	8	8			
CCE2			8	8	
CCE3					8
ESE	12	12	12	12	12
Total	20	20	20	20	20

Assessment Pattern

BT Level	CCE						ESE				
	CCE1	CCE2	CCE3	CCE4							
CO	CO1						CO1	CO2	CO3	CO4	CO5
L1											
L2	√	√					√	√	√	√	√
L3	√	√	√				√	√	√	√	√
L4	√	√	√				√	√	√	√	√
L5											
L6											

This course serves as a prerequisite for / maps with the following future courses:

- Machine Learning
- Data Analytics
- Artificial Intelligence
- Data Structures and Algorithms
- Design and Analysis of Algorithms
- Database Management Systems
- Deep Learning
- Natural Language Processing
- Computer Networks
- Operations Research
- Big Data Analytics
- Predictive Modeling and Statistical Learning
- GATE Exam/ CAT Exam /Aptitude Tests

Acela

R. Ganesh

**Job Mapping:**

Job opportunities that one can get after learning this course:

- Data Analyst
- Data Scientist (Entry Level)
- Machine Learning Engineer (Junior)
- Business Intelligence Analyst
- Statistical Analyst
- AI Engineer (Associate Level)
- Research Analyst
- Operations Analyst
- Software Developer (Analytical roles)
- Data Engineer (Basic level)
- Quantitative Analyst (Finance sector entry roles)
- AI/ML Intern / Trainee roles



Program:		S. Y. B. Tech			Semester:		III	
Course:		Entrepreneurship Development			Course Code:		R25-AI-EEM-206	
Teaching Scheme				Examination Scheme				
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR	
01	02	-	02	25	-	25	-	
Course Description: This course introduces students to the fundamentals of entrepreneurship, focusing on opportunity identification, business model development, market validation, and startup execution. It integrates practical tools such as Business Model Canvas, customer validation techniques, and basic financial planning to enable students to conceptualize and develop viable startup ideas.								
Course Relevance: The course is highly relevant in today’s innovation-driven economy, as it equips students with entrepreneurial skills, problem-solving ability, and digital business knowledge. It prepares students for startup creation, self-employment, and industry roles requiring innovation, adaptability, and business acumen.								
Prerequisite: 1. Fundamentals of Management 2. Computer Fundamentals 3. Basic Economics								
Bridge Content: Students with basic knowledge of business, finance, communication, and digital tools								
Course Objectives: The course aims to: 1. Introduce the fundamental principles of entrepreneurship, forms of business organizations, and The startup ecosystem. 2. Enable students to identify, evaluate, and select viable business opportunities using structured Techniques. 3. Familiarize students with business models, financial planning, and market validation strategies. Expose students to key marketing strategies, customer acquisition techniques, and branding Essentials for startups 4. Develop student’s entrepreneurial mindset and their ability to communicate and pitch business Ideas effectively using structured storytelling techniques								
Course Outcomes: After learning the course, students will be able to								
CO	Course Outcome					Bloom’s Level		
CO1	DESCRIBE the role of entrepreneurship in economic growth and the startup ecosystem.					2		
CO2	APPLY creative techniques to viable business ideas based on customer needs.					3		
CO3	DEVELOP a basic business model using tools like the Business Model Canvas through					4		
CO4	INVESTIGATE marketing strategies for startups deliver a concise business pitch using storytelling and effective communication techniques					4		
Course Content								



Unit 1	Foundations of Entrepreneurship	3 Hrs.	CO1
Entrepreneurship: Definition and evolution, Role of entrepreneurship in economic development, Role in job creation, GDP, and innovation. Characteristics of an Entrepreneur: Key traits: Risk-taking, innovation, pro-activeness, Leadership, perseverance, and resilience. Types of Entrepreneurships: Startup entrepreneurship, social entrepreneurship, Intrapreneurship (corporate entrepreneurship), Lifestyle and small business entrepreneurship, Forms of Business Organization-Sole proprietorship, partnership, private limited, public limited. Entrepreneurial Mindset: Growth mindset and adaptability, Creativity and problem-solving, Opportunity recognition and initiative-taking, Ethical & Responsible Entrepreneurship Overview of the Startup Ecosystem: Key stakeholders: Incubators, accelerators, angel investors, vcs, Government support schemes (Startup India, Atal Innovation Mission, etc.), Global vs. Indian startup ecosystems			
Exemplars / Practical Applications and usage: 1. Siddharth A K – Waste-to-value innovation 2. Aman Gottumukkala – Built startup with small team, moved to AI ecosystem 3. Chetan Garg – edtech entrepreneur solving student problems 4. Kaivalya Vohra – One of the youngest unicorn founders			
Unit 2	Opportunity Identification & Design Thinking	4 Hrs.	CO2
Evaluating Opportunities: Difference between an “idea” and an “opportunity”. Basic filters: Desirability, feasibility, and viability. Tools: SWOT Analysis, Opportunity Matrix, Industry trends, market gaps. Understanding Customer Needs and Pain Points, Creativity Techniques for Idea Generation: Definition and importance of creativity in entrepreneurship Scamper, design thinking steps.			
Brainstorming: Rules of effective brainstorming. Individual vs. Group brainstorming. Mind Mapping: Visual idea structuring using central themes and branches. Tools (manual and digital) for mind mapping.			
Exemplars / Practical Applications and usage: • Quick commerce (Zepto) • Social commerce (Meesho) • EV startups (blusmart) • Creator-led startups (physicswallah)			
Unit 3	Business Models & Lean Startup	4 Hrs.	CO3
Introduction to Business Model Canvas: Definition and purpose of a business model, Overview of The Business Model Canvas by Osterwalder, Benefits of using BMC for startups.			
Key Components of BMC: Value Proposition: Defining what unique value the product/service offers. Addressing customer pain points. Customer Segments: Identifying target customers. Creating Customer personas Revenue Models: Direct sales, subscriptions, freemium, licensing, etc.			
Basic Market Research for Validation: Importance of market research in early-stage business development. Designing effective surveys and customer feedback forms. Conducting basic interviews And analyzing responses. Introduction to MVP (Minimum Viable Product) and feedback loops.			
Exemplars / Practical Applications and usage: Map the BMC for a well-known startup (e.g., zepto, CRED, ather energy, perplexity AI)			
Unit 4	Market Research & Funding	4 Hrs.	CO4



Basic of Funding, TAM, SAM, SOM Analysis, Customer Segmentation Techniques, Competitor Intelligence, SWOT into actionable strategies, Modern Tools & Methods (Digital Market Research Tools, Customer Validation Techniques, Data Analytics Basics), Costing, Pricing Strategies, Break-even Analysis, Funding Sources.

Search Engine Optimization (SEO): Basics of how search engines work, Keyword research and Content optimization, On-page vs. Off-page SEO Importance of Digital Presence – Website essentials, blogs, and analytics tools.

Customer Acquisition Strategies: Understanding the Customer Journey – Awareness, interest, decision, action. Early-Stage Customer Acquisition Tactics: Word-of-mouth & referrals, Influencer marketing(micro-influencers), Email marketing basics, building a landing page and collecting leads

Pitching & Startup Execution: Storytelling & Communication, Overview of Funding Sources.

Exemplars / Practical Applications and usage:

Mamaearth, physicswallah, EV Charging Startup, Canva.

Shark Tank India – Pitch Analysis (Any Season), Airbnb – The Original Pitch Deck

Suggested Projects:

1. Smart campus waste management startup converting waste into reusable products or energy.
2. IoT-enabled water leakage detection system for homes and industries.

Textbooks:

1. Bygrave, W.D., Zacharakis, A., & Corbett, A.C. Entrepreneurship, 6th Edition, Wiley, 2025.ISBN: 9781394262809.
2. Drucker, Peter F. Innovation and Entrepreneurship: Practice and Principles, Reprint Edition, Harper Business, 2006. ISBN: 9780060851132.
3. Osterwalder, Alexander & Pigneur, Yves. Business Model Generation: A Handbook for Visionaries,Game Changers, and Challengers, 1st Edition, Wiley, 2010. ISBN: 9780470876411.

Reference Books:

1. Ries, Eric. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses, 1st Edition, Crown Business, 2011. ISBN: 9780307887894.
2. Kawasaki, Guy. The Art of the Start 2.0: The Time-Tested, Battle-Hardened Guide for Anyone Starting Anything, Portfolio (Penguin Random House), 2015. ISBN: 9781591847847.

Moocs Links and additional reading material:

1. Entrepreneurship Essential https://onlinecourses.nptel.ac.in/noc20_ge08/preview
2. Overview of Entrepreneurship https://onlinecourses.nptel.ac.in/noc21_mg70/preview
3. Entrepreneurship as the Foundation https://onlinecourses.nptel.ac.in/noc20_mg35
4. Entrepreneurship Guide <https://www.coursera.org/learn/entrepreneur-guide-beginners>
5. Wadhvani Foundation1 <https://wadhwanifoundation.org/>
6. Wadhvani Foundation2 <https://www.youtube.com/@wadhvani-foundation/videos>

CO-PO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	1	–	–	2	1	–	1	1	2	1	1	3

20

Aeela

R. Ganesh



CO2	1	2	3	1	2	1	–	1	1	2	2	2	3	3
CO3	1	2	3	1	2	1	–	2	2	3	2	2	3	3
CO4	–	2	2	2	1	1	1	2	3	2	2	1	2	3
Avg.	1	2	2	1	1	2	1	1	2	2	2	2	2	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Unit wise Weightage:

Exam	Unit1	Unit2	Unit3	Unit4	
CCE					
CCE1	6	6			
CCE2			6	7	
CCE3					
ESE					
Total	6	6	6	7	

Assessment Pattern

BT Level	CCE						ESE				
	CT1	CCE2		CCE4							
CO							CO1	CO2	CO3	CO4	CO5
L1											
L2	√	√									
L3	√	√									
L4	√	√									
L5											
L6											

This course serves as a prerequisite for / maps with the following future courses:

1. Startup Management
2. Digital Marketing / Product Management

Job Mapping:

Job opportunities that one can get after learning this course: startup founder, business development executive, digital marketing associate, product assistant, or innovation analyst in startup and corporate environments.

Guidelines for Lab Conduction

Set of suggested assignment list is provided in list - Each student must perform at all assignments
Operating System recommended: - Microsoft Windows 10/11 (64-bit)
Tools recommended: - Canva or Any AI tool

Guidelines for Student's Laboratory Journal

Aeeha

R. Ganesh



Maintain the laboratory journal regularly and neatly. Write the experiment/activity title, date, and objectives clearly. Record observations, discussions, and outcomes of each activity. Include business ideas, case studies, and entrepreneurial concepts discussed in practical sessions. Prepare reports on activities, presentations, and field visits, if any. Complete all assignments and practical activities within the stipulated time.

- Obtain the faculty's signature after completing each activity.
- Submit the laboratory journal for assessment as per the academic schedule

Guidelines for Laboratory/Term Work Assessment

Continuous assessment of laboratory work should be done based on overall performance and Laboratory assignments performance of students. Each Laboratory assignment assessment should be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include Continuous assessment should focus on creativity, innovation, problem-solving ability, and entrepreneurial skills.

Experiment No. 1	Entrepreneurial Mindset Reflection	2 Hrs.	CO1
Objective: To encourage students to explore their personal views on entrepreneurship and recognize the key characteristics of an entrepreneurial mindset by studying the journey of a real-world entrepreneur Associated Tasks: Write a reflective essay (500–600 words) explaining what entrepreneurship means to you. • Identify an entrepreneur (Indian or global) you admire and explain the reasons for admiration. • Highlight key mindset traits such as risk-taking, resilience, innovation, and adaptability that contributed to their success. • Reflect on how these traits relate to your own strengths or areas for development			
Experiment No. 2	Idea Generation Challenge	2 Hrs.	CO1
Objective: To foster creativity, structured brainstorming, and the ability to identify potential business opportunities based on real-world problems. Associated Tasks: Generate 10 Business Ideas Use any structured brainstorming technique 1. Ideas can be tech-based, social impact, service-based, or product-based 2. Select One Idea- Choose the most promising idea from your list 3. Write a 1-page Concept Summary, include the following: • Problem Identified: Describe the specific problem or pain point your idea addresses. • Solution Overview: Briefly describe your business idea. • Target Audience: Identify the group of people or organizations that would benefit. • Market Potential: Discuss the viability and scalability of the idea.			
Experiment No. 3	Business Model & Customer Validation	2 Hrs.	CO2

**Part A: Business Model Canvas**

1. Choose a business idea (from Assignment 2 or a new one).
2. Create a Business Model Canvas with all 9 key blocks:
 - Customer Segments
 - Value Propositions
 - Channels
 - Customer Relationships
 - Revenue Streams
 - Key Resources
 - Key Activities
 - Key Partnerships
 - Cost Structure

3. Present the BMC in visual or tabular format

Part B: Customer Interviews & Insights

1. Identify 2–3 potential customers from your target segment.
 2. Conduct brief interviews (5–10 minutes each) to gather insights on:
 - Their pain points
 - Their reaction to your proposed solution
 - Willingness to pay or use your product/service
 3. Summarize findings in a 1–2 page report that includes:
 - Key customer quotes or paraphrased insights
 - A revised Value Proposition or Customer Segment block (if needed)
- A short reflection: key learnings and potential changes to your idea.

Experiment No. 4	Business Launch Plan – Marketing & Financial Snapshot	2 Hrs	CO3
-------------------------	--	--------------	------------

Objective: To develop a practical understanding of how marketing strategy and financial planning go hand-in-hand in launching a startup. Students will define a basic marketing campaign and align it with estimated costs, pricing, and projected revenue.

Associated Tasks:

You are preparing to launch your business idea.

Prepare a combined Marketing and Financial Snapshot including the following:

Part A: Marketing Campaign Plan

- Define your target market by identifying primary customers.
- Design a mini-campaign using one or more of the following channels:
 - Social media (e.g., Instagram, LinkedIn)
 - Print/digital flyers
 - Email marketing
- Describe the campaign content, including the message or offer to be promoted.
- Optionally, create 1–2 sample marketing materials.
- Write a 300-word explanation outlining your marketing strategy and expected impact.

Part B: Financial Snapshot

1. Startup Costs – Estimate your initial costs (fixed + Variable)
2. Pricing Strategy – State your pricing model and justification
3. Break-even Analysis – Basic cost vs. Sales estimate
4. 6-Month Revenue Projection – Expected sales and Income
5. Format: Use a simple table or spreadsheet (optional).

Experiment No. 5	Elevator Pitch Video	2 Hrs.	CO4
-------------------------	-----------------------------	---------------	------------



Objective: To help students develop confidence and clarity in presenting their business idea in a short, compelling format. The exercise simulates real-world investor or networking scenarios where entrepreneurs must grab attention quickly.

Associated Tasks:

Prepare a 90-second elevator pitch for your business idea (the same or refined idea used in earlier assignments).

Your pitch should cover the following elements:

- The Problem – Problem Identification
- The Solution – Description of your product/service.
- Value Proposition – The unique value proposition.
- Target Audience – Audience for your idea.
- Call to Action – E.g. Request for support, funding, feedback, etc.

Deliver Your Pitch:

- Record a video and submit it with written/video version of your pitch.
- Ensure clear speech, confident body language (for video), and persuasive tone.

Reflection (Short Write-up):

- Share what you learned about communicating your idea
- Describe challenges or rewards you experienced in the process



Program:	S. Y. B. Tech			Semester:		III	
Course:	Universal Human Values & Professional Ethics			Course Code:		R25-AI-VEC-207	
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
02	-	-	02	50	-	-	-
Course Description: This course provides a blueprint for long-term fulfilment by balancing technical skills with life-long ethical values. The syllabus focuses on Self-Exploration, helping you distinguish between material success and lasting mental peace to prevent future burnout. It covers Harmony in the Self, Family, and Society, providing the emotional intelligence needed to navigate high-pressure careers and build trust-based relationships. By exploring Harmony in Nature and Professional Ethics, it prepares you to lead with integrity and social responsibility. Ultimately, it equips you with a clear moral compass to handle adulthood’s complexities with stability and purpose.							
Course Relevance: This course serves as a life-long compass for adulthood, moving you from simply "earning" to truly "living." It helps you distinguish between material success and mental peace, preventing burnout by teaching you to manage your internal state during high-pressure situations. By fostering Trust and Respect, it ensures your personal and professional relationships are built on harmony rather than ego. In your career, it provides the ethical framework needed to lead with integrity, securing a reputation that lasts longer than any job title. Ultimately, it equips you with the emotional intelligence to navigate life's complexities with clarity and purpose.							
Prerequisite: Student Induction Program (SIP)							
Bridge Content: Basic Human Values							
Course Objectives: This course aims to 1. Help the students develop a holistic, humane world-vision, and appreciate the essential complementarity between values and skills to ensure mutual happiness and prosperity 2. Elaborate on ‘Self-exploration’ as the process for Value Education 3. Facilitate the understanding of harmony at various levels starting from self and going towards family and society. 4. Elaborate on the salient aspects of harmony in nature and the entire existence 5. Explain how the Right understanding forms the basis of Universal human values and definitiveness of Ethical human conduct. 6. Provide the vision for a holistic way of living and facilitate transition from chaotic life to an orderly life.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome					Bloom’s Level	
CO1	EXAMINE their own potential to explore and apply this understanding in daily life					4	



CO2	DIFFERENTIATE the human being as the coexistence of self and body to identify their real needs and basic aspirations in daily life.	4	
CO3	RELATE the relationship between one self and the other self as an essential part of harmony in the family and apply it in daily life interactions	4	
CO4	ANALYSE the interconnectedness, harmony, and mutual fulfilment in nature and existence to develop ethical, holistic, and responsible approaches for daily life, technology, production systems, and societal management practices.	4	
Course Content			
Unit 1	Introduction to Value Education	7 Hrs.	CO1
(i) Understanding Value Education (ii) Self-exploration as the Process for Value Education (iii) Continuous Happiness and Prosperity - the Basic Human Aspirations and their Fulfilment (iv) Right Understanding, Relationship and Physical Facility (v) Happiness and Prosperity - Current Scenario (vi) Method to Fulfil the Basic Human Aspirations			
Exemplars / Practical Applications and usage: Discussion on modern lifestyle stress despite material growth			
Unit 2	Harmony in the Human Being	7 Hrs.	CO2
(i) Understanding Human being as the Co-existence of the Self and the Body (ii) Distinguishing between the Needs of the Self and the Body (iii) The Body as an Instrument of the Self (iv) Understanding Harmony in the Self (v) Harmony of the Self with the Body (vi) Programme to Ensure self-regulation and Health			
Exemplars / Practical Applications and usage: Prepare a personal health and self-regulation plan to improve their mental and physical well-being in daily life.			
Unit 3	Harmony in the Family and Society	8 Hrs.	CO3
(i) Harmony in the Family - the Basic Unit of Human Interaction "Trust" - the Foundational Value in Relationship (ii) 'Respect' - as the Right Evaluation (iii) Values in Human-to-Human Relationship (iv) Understanding Harmony in the Society (v) Vision for the Universal Human Order			
Exemplars / Practical Applications and usage: List the important relationships in their life (family, friends, teachers) and identify values like trust and respect in each relationship			
Unit 4	Harmony in the Nature (Existence) & Professional Ethics	8 Hrs.	CO4



- (i) Understanding Harmony in the Nature
- (ii) Interconnectedness, self-regulation and Mutual Fulfilment among the Four Orders of Nature
- (iii) Realizing Existence as Co-existence at All Levels
- (iv) The Holistic Perception of Harmony in Existence
- (v) Professional Ethics in the light of Right Understanding
- (vi) Strategies for Transition towards Value-based Life and Profession

Exemplars / Practical Applications and usage:

Students should conduct group activities for tree plantation/campus garden maintenance/waste segregation campaigns/water conservation awareness, and submit a brief report with photographs and observations demonstrating their contribution towards harmony with nature.

Textbooks:

1. A Foundation Course in Human Values and Professional Ethics, RR Gaur, R Asthana, GP Bagaria, 3rd revised edition, UHV Publications, 2023, ISBN: 978-81-957703-7-3 (Printed Copy), 978-81-957703-6-6 (e-book)
2. Teachers' Manual for A Foundation Course in Human Values and Professional Ethics, RR Gaur, R Asthana, GP Bagaria, 3rd revised edition, UHV Publications, 2023, ISBN: 978-81-957703-5-9 (Printed Copy), 978-81-957703-0-4 (e-Book)

Reference Books:

1. P. L. Dhar, R. R. Gaur, 1990, Science and Humanism, Commonwealth Publishers.
2. A. Nagaraj, 1999, Jeevan Vidya: Ek Parichaya, Jeevan Vidya Prakashan, Amarkantak
3. B. P. Banerjee, 2005, Foundations of Ethics and Management, Excel Books.
4. A. N. Tripathy, 2003, Human Values, New Age International Publishers.
5. E. G. Seebauer & Robert L. Berry, 2000, Fundamentals of Ethics for Scientists & Engineers, Oxford University Press
6. B. L. Bajpai, 2004, Indian Ethos and Modern Management, New Royal Book Co., Lucknow. Reprinted 2008.
7. M. Govindrajran, S Natrajan & V.S. Senthil Kumar, Engineering Ethics and Human Values, Eastern Economy Edition, Prentice Hall of India Ltd.
8. M. K. Gandhi, "The Story of my Experiments with Truth", Discovery Publisher

MOOCs Links and additional reading material:

1. Swayam Course on "Understanding Human Being Nature and Existence Comprehensively" by Dr. Kumar Sambhav, Director, UP Institute of Design (UPID), Noida. https://onlinecourses.swayam2.ac.in/aic22_ge23/preview
2. NPTEL Course on "Exploring Human Values: Visions of Happiness and Perfect Society" by Prof. A. K. Sharma, Department of Humanities and Social Sciences, IIT Kanpur. <https://nptel.ac.in/courses/109104068>
3. Gaur Gopal Das: <https://www.youtube.com/@GaurGopalDas>
4. Sadguru : <https://www.youtube.com/@sadhguru>

CO-PO-PSO Mapping:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	2	1	1	1	-	2	1	-	-
CO2	-	-	-	-	-	2	1	1	1	-	2	1	-	-



CO3	-	-	-	-	-	2	1	1	1	-	2	1	-	-
CO4	-	-	-	-	-	2	1	1	1	-	2	1	-	-
Avg.	-	-	-	-	-	2	1	1	1	-	2	1	-	-

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Unit wise Weightage:

Exam	Unit1	Unit2	Unit3	Unit4
CCE				
CCE 1	12	13		
CCE 2			12	13
Total	12	13	12	13

Assessment Pattern

BT Level	CCE		ESE				
	CCE1	CCE2	CO1	CO2	CO3	CO4	CO5
CO							
L1							
L2	√	√	√	√	√	√	√
L3	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√
L5							
L6							

This course serves as a prerequisite for / maps with the following future courses:

1. Sustainable Development
2. Humanities and Social Sciences (HSS)

Job Mapping:

Job opportunities that one can get after learning this course:

1. Human Resource (HR) Specialist
2. Public Relations (PR) Manager
3. Counselor / Life Coach
4. Sustainability Consultant



Program:	S. Y. B. Tech			Semester:		III	
Course:	Data Structures Lab			Course Code:		R25-AI-PCC-208	
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
-	2	-	01	-	-	25	25
Course Description: This course covers fundamental data structures including arrays, stacks, queues, linked lists, trees, graphs, and hashing techniques. Students learn to implement, manipulate, and analyze these structures to efficiently store, retrieve, and process data, with emphasis on time and space complexity and solving real-world problems							
Course Relevance: The course provides the foundation for programming, algorithms, and database systems, enabling efficient management of large datasets in software development and analytics. It prepares students for real-world applications, coding interviews, and system design while developing skills to analyze and select appropriate data structures for optimal performance.							
Prerequisite: 1. Programming & Problem Solving 2. Programming using C							
Bridge Content: Basics of problem solving & Programming (C/C++/Python)							
Course Objectives: 1. To apply searching, sorting, and hashing techniques to efficiently solve data storage and retrieval problems. 2. To implement and manipulate singly and doubly linked lists to manage dynamic real-world data and analyse performance. 3. To apply stack and queue data structures to solve practical problems. 4. To analyse hashing techniques to evaluate the efficiency of data storage and retrieval using methods such as hash tables, linear probing, chaining, and open addressing. 5. To implement Binary Search Trees and Graph data structures to efficiently store, retrieve, and analyse hierarchical and networked data for real-world applications.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome					Bloom's Level	
CO1	ANALYZE performance of searching, sorting, and hashing techniques to solve data storage and retrieval problems efficiently.					4	
CO2	APPLY stack and queue data structures to solve specific real-world problems					3	
CO3	USE singly and doubly linked lists to manage real-world applications based on requirements.					3	
CO4	ANALYZE hashing techniques to evaluate the efficiency of data storage and retrieval using methods such as hash tables, linear probing, chaining and open addressing					4	



CO5	IMPLEMENT binary search trees and graph data structures to efficiently store, retrieve, and analyze hierarchical and networked data, modeling real-world problems.	4
Guidelines for Lab Conduction		
Set of suggested assignment list is provided in groups- A, B, C, D, and E. Each student must perform at least 10 assignments as at least 2 from group A, 2 from group B, 2 from group C, 2 from group D and 2 from group E and one mandatory mini project based on data structure enlisted in the syllabus.		
- Operating System recommended: - Microsoft Windows 10/11 (64-bit) - Programming tools recommended: - Visual Studio Code, Dev-C++, Code: Blocks, Eclipse IDE, Visual Studio IDE, or other suitable development environments as applicable to the course.		
Guidelines for Student's Laboratory Journal		
The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of prologue, Certificate with table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Theory Concept in brief, algorithm, test cases, conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, Attaching printed papers as part of write-ups and program listing to journals may be avoided. Students' programs maintained on cloud or college server by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory for accreditation purposes.		
Guidelines for Laboratory/Term Work Assessment		
Continuous assessment of laboratory work should be done based on overall performance and Laboratory assignments performance of students. Each Laboratory assignment assessment should be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include timely completion performance, innovation, efficient codes, punctuality, and neatness.		
Guidelines for Practical Examination		
Both examiners should set the problem statements together, and assessment should focus mainly on correct implementation. Supplementary questions may be asked to check deeper understanding and efficiency. A fair, transparent, and encouraging evaluation process will build student confidence and support a strong academic start.		
Group-A(Searching & Sorting) Any 2		
Experiment No. 1	Write a menu driven program to store roll numbers of students in an array who attended a session in random order. Write a function for searching whether a particular student attended a session or not, using Linear search and Binary Search. Analyse algorithm by printing total number of comparisons done for searching element in respective algorithm.	2 Hrs.
		CO1



Objective: To compare Linear Search and Binary Search by searching student roll numbers and analyzing the number of comparisons required in each method.

Associated Tasks:

1. Store roll numbers of students in an array in random order.
2. Implement a menu-driven program to choose Linear Search or Binary Search.
3. Write a function to perform Linear Search for a given roll number.
4. Sort the array before performing Binary Search and write a function to locate the roll number.
5. Display whether the student attended the session or not.
6. Display whether the student attended the session and the total comparisons made by each algorithm

Exemplars and Utility:

Exemplars:

1. Roll numbers: [12, 45, 23, 67, 34]
Search key: 23 → Output: Student attended the session, Comparisons: Linear = 3, Binary = 2
2. Roll numbers: [12, 45, 23, 67, 34]
Search key: 50 → Output: Student did not attend the session, Comparisons: Linear=5, Binary = 3

Utility: Students can apply this knowledge to develop programs for fast data retrieval, record searching, and other real-world applications requiring efficient search operations

Experiment No. 2	Write a program to store a set of numerical data (such as student scores, employee salaries, or task priorities) in an array and find the Kth largest and Kth smallest values.	2 Hrs.	CO1
-------------------------	---	---------------	------------

Objective: To find the Kth largest and Kth smallest elements in a dataset and understand their application in real-world scenarios like rankings, salaries, or task prioritization.

Associated Tasks:

1. Accept numerical data (scores, salaries, or priorities) from the user and store them in an array.
2. Input the value of K.
3. Sort the array in ascending or descending order.
4. Identify and display the Kth largest and Kth smallest elements.
5. Ensure that K is valid ($1 \leq K \leq$ number of elements).

Exemplars and Utility:

Exemplars:

1. Finding the 3rd highest score in a class of students.
2. Identifying the 2nd lowest salary in a company.
3. Determining the 5th priority task in a task management system.
4. Check filter function of Excel sheet where data is sorted as per expected order

Utility: Students can apply this knowledge to rank items, identify thresholds, or prioritize tasks in real-world scenarios like academic performance analysis, payroll processing, and project/task management systems.

Experiment No. 3	Write a program to store the first-year percentage of students in an array. Implement functions to sort the array of floating-point numbers in ascending order using: Selection Sort & Bubble Sort.	2 Hrs.	CO1
-------------------------	--	---------------	------------



	After sorting, display the top five scores. Analyse the algorithm by printing total number of comparisons done for searching element in respective algorithm.		
Objective: To sort student percentages using Selection Sort and Bubble Sort and identify the top five performers in the class. Associated Tasks: 1. Accept the number of students and their percentages as input. 2. Store the percentages in a floating-point array. 3. Implement Selection Sort and Bubble Sort to sort the array in ascending order. 4. Display the sorted array. 5. Display the top five percentages from the sorted array. 6. Count and display the total number of comparisons performed in each sorting method Exemplars and Utility: Exemplars: Sorting percentages [78.5, 91.2, 66.8, 88.0, 73.5, 95.6] → Top five scores: 95.6, 91.2, 88.0, 78.5, 73.5 Sorting percentages [65.0, 82.3, 74.5, 90.0, 85.5, 77.0] → Top five scores: 90.0, 85.5, 82.3, 77.0, 74.5 Utility: Students can apply this knowledge to analyze academic performance, rank students, and compare sorting algorithms for efficiency in real-world datasets.			
Experiment No. 4	Write a program to store a set of numerical data (such as student scores, employee salaries, or product prices) in an array and sort the data using Quick Sort. Elaborate on analysis of an algorithm.	2 Hrs.	CO1
Objective: To implement Quick Sort for sorting numerical data and understanding the efficiency and analysis of the algorithm. Associated Tasks: 1. Accept the number of elements and numerical data as input. 2. Store the data in an array. 3. Implement Quick Sort to sort the array in ascending order. 4. Display the sorted array. 5. Analyze the algorithm by determining: a. Number of comparisons performed b. Time complexity in best, average, and worst cases c. Space complexity due to recursion stack Exemplars and Utility: Exemplars: Sorting scores [78, 91, 66, 88, 73, 95] → Output: [66, 73, 78, 88, 91, 95] Sorting salaries [45000, 78000, 32000, 90000, 61000] → Output: [32000, 45000, 61000, 78000, 90000] Utility: Students can apply Quick Sort to: Efficiently organize and rank data in real-world scenarios such as academic performance, payroll systems, or inventory management.			
Group-B (Stack & Queue)			



Experiment No.5	In any language program mostly syntax error occurs due to unbalancing delimiter such as (), {}, []. Write a program using stack to check whether a given expression is well parenthesized or not.	2 Hrs.	CO2
Objective: To use a stack to check whether an expression is well-parenthesized and understand the importance of stack in syntax validation. Associated Tasks: 1. Accept an expression as input from the user. 2. Implement a stack to store opening delimiters (, {, [. 3. Traverse the expression: • Push opening delimiters onto the stack. • Pop and match closing delimiters), },] with the top of the stack. 4. Determine whether the stack is empty at the end and all delimiters match. 5. Display whether the expression is well-parenthesized or not. Exemplars and Utility: Exemplar: 1. Input: (a+b) *(c-d) → Output: Well parenthesized 2. Input: {[a+b] *(c-d) → Output: Not well parenthesized 3. Input: ((a+b) *[c-d]) → Output: Well parenthesized. Utility: Students can apply this knowledge to: • Detects syntax errors in programming languages. • Validate expressions in calculators and parsers. • Understand stack operations and their practical use in real-world programming.			
Experiment No.6	Queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write a program for simulating job queue using an array. Write functions to add jobs and delete jobs from the queue.	2 Hrs.	CO2
Objective: To implement a queue using a linked list and simulate first come, first served processing of jobs or orders. Associated Tasks: 1. Define a node structure to store job/order information. 2. Implement enqueue operation to add a job at the rear of the queue. 3. Implement dequeue operation to remove a job from the front of the queue. 4. Display all current jobs/orders in the queue. 5. Ensure the queue does not exceed the maximum limit M. 6. Create a menu-driven program to allow interactive operations. Exemplars and Utility: Exemplar: 1. Maximum orders M = 5 2. Enqueue: Order1 → Order2 → Order3 → Order4 → Order5 3. Dequeue: Remove Order1 → Queue now has Order2 → Order5 4. Enqueue: Order6 → Queue updated: Order2 → Order5 → Order6			



Utility: Students can apply this knowledge to:

- Simulate real-world queues like order management or service lines.
- Understand queue operations (enqueue/dequeue) and their applications in job scheduling.
- Learn dynamic memory management with linked lists for flexible queue implementation.

Group-C (Linked List)

Experiment No.7	The Department of AI&DS has a student's club named 'Pinnacle Club'. Students of the second, third and final year of the department can be granted membership on request. Similarly, one may cancel the membership of the club. The first node is reserved for the president of the club and the last node is reserved for the secretary of the club. Write a program to maintain club member's information using singly linked lists. Store student PRN and Name. Write functions to: a) Add and delete the members as well as president or even secretary. b) Compute total number of members of club c) Display members d) Two linked lists exist for two divisions. Concatenate two lists.	2 Hrs.	CO3
------------------------	--	---------------	------------

Objective: To implement a singly linked list to manage dynamic club member information and perform operations such as insertion, deletion, counting, display, and concatenation.

Associated Tasks:

1. Define a node structure to store PRN and Name.
2. Implement functions to:
 - a. Add members at any position, including President (first node) or Secretary (last node).
 - b. Delete members from any position.
 - c. Compute total number of members in the club.
 - d. Display all members.
 - e. Concatenate two linked lists for two divisions.

Create a menu-driven interface to perform all operations interactively.

Exemplars and Utility:

Exemplars:

Division A members: President – PRN 101, Alice; Members – PRN 102, Bob; PRN 103, Carol; Secretary – PRN 110, Eve Division B members: President – PRN 201, Frank; Members – PRN 202, Grace; Secretary – PRN 210, Heidi

After concatenation: Combined list shows all members of Division A followed by all members of Division B in order.

Utility: Learn the importance of linked lists for real-world applications where arrays are inefficient.

Experiment No.8	Write a program to find the 1's complement, 2's compliment and addition of a given binary number using doubly linked list	2 Hrs.	CO3
------------------------	--	---------------	------------

Objective: To understand and implement a Doubly Linked List and perform various operations on it

Associated Tasks:

1. Accept a binary number from the user.
2. Perform 1's complement by changing all 1s to 0s and all 0s to 1s.
3. Perform binary addition by adding 1 to the obtained 1's complement value.

Heela

R. G. Jain



4. Generate the 2's complement using the result of the binary addition.
5. Display the original binary number, 1's complement, binary addition process, and 2's complement result.

Exemplars and Utility:**Exemplar:**

1. Input: 1010
1's Complement: 0101
2. $0101 + 1 = 0110$
2's Complement: 0110
3. Input:
First Binary Number: 1010
Second Binary Number: 0101

$$\begin{array}{r} 1010 \\ + 0101 \\ \hline 1111 \end{array}$$

Output: 1111

Utility: Students can apply this knowledge to:

- perform binary arithmetic operations,
- represent negative numbers in computer systems,
- understand low-level data representation used in digital electronics and computer architecture.

Group D(Hashing)

Experiment No.9	Consider the telephone book database of N clients. Make use of a hash table implementation to quickly look up a client's telephone number. Make use of linear probing, quadratic collision handling techniques	2 Hrs.	CO4
------------------------	---	---------------	------------

Objective: To implement hashing techniques for efficient storage and retrieval of client telephone numbers using linear probing and quadratic probing to handle collisions.

Associated Tasks:

1. Create a hash table to store client records (Name, Telephone Number).
2. Implement linear probing to resolve collisions during insertion.
3. Implement quadratic probing as an alternative collision resolution method.
4. Implement a function to search for a client's telephone number efficiently using the hash table.

Exemplars and Utility:



Exemplars: Demonstrate collision handling using linear and quadratic probing.

Utility: Students can apply this knowledge to quickly look up records in real-world databases such as telephone directories or employee databases

Experiment No.10	Hash Table Implementation for Efficient Data Storage and Retrieval Using Chaining	2 Hrs.	CO4
-------------------------	--	---------------	------------

Objective: To analyze and implement hashing techniques for efficient data storage and retrieval, handling collisions using chaining

Associated Tasks:

1. Create a hash table to store student records (PRN, Name, Marks).
2. Implement a suitable hash function for indexing records.
3. Handle collisions using the chaining technique (linked list at each index).
4. Implement insertion, searching, and deletion of records.
5. Display the contents of the hash table along with chains formed due to collisions.
6. Analyze the efficiency of hashing with chaining for data storage and retrieval

Exemplars and Utility:

Exemplar:

Demonstrate collision handling using separate chaining, where multiple records mapping to the same hash index are stored in a linked list attached to that index.

Utility: Hashing with chaining is used in databases, caching systems, symbol tables, and information management systems to enable fast data storage and retrieval while efficiently handling collisions.

Group- E (Tree & Graph)

Experiment No.11	Beginning with an empty binary search tree, construct a binary search tree by inserting the values in the order given. After constructing a binary tree - i. Insert new node ii. Find number of nodes in longest path from root iii. Minimum data value found in the tree iv. Change a tree so that the roles of the left and right pointers are swapped at every node v. Search a value.	2 Hrs.	CO5
-------------------------	--	---------------	------------

Objective: To implement and manipulate a Binary Search Tree for insertion, traversal, search, height calculation, and mirroring, and analyze its properties.

Associated Tasks:

1. Create a node structure for the BST.
2. Construct a BST by inserting values in the specified order.
3. Implement a function to insert a new node.
4. Implement a function to calculate the height or longest path from root.
5. Implement a function to find the minimum value in the BST.
6. Implement a function to mirror the BST by swapping left and right pointers at every node.
7. Implement a search function to locate a given value in the BST.
8. Display the tree contents using inorder, preorder, or postorder traversal

Exemplars and Utility:

Exemplars:

1. Input values: 50, 30, 70, 20, 40, 60, 80



2. Insert 25 → Updated BST
3. Longest path from root → 4 nodes
4. Minimum value → 20
5. Mirror tree → Left and right subtrees swapped
6. Search 60 → Found, Search 90 → Not Found

Utility: Students can apply this knowledge to manage hierarchical data such as employee records, product catalogs, academic results, in efficient data retrieval, decision-making systems, graphics, compiler design and network routing.

Experiment No. 12	There are flight paths between cities. If there is a flight between city A and city B then there is an edge between the cities. The cost of the edge can be the time that flight takes to reach city B from A or the amount of fuel used for the journey. Represent this as a graph. The node can be represented by the airport name or name of the city. Use adjacency list representation of the graph or use adjacency matrix representation of the graph. Check whether the graph is connected or not.	2 Hrs.	CO5
--------------------------	---	---------------	------------

Objective: To represent flight paths between cities as a weighted graph and apply adjacency list or adjacency matrix representation to analyze connectivity between airports/cities.

Associated Tasks:

1. Create a graph where each city or airport is represented as a node and each flight path is represented as an edge with cost such as travel time or fuel consumption.
2. Represent the flight network using an adjacency list by listing each city and its directly connected cities along with edge weights.
3. Represent the same flight network using an adjacency matrix where rows and columns represent cities and matrix values represent flight cost or travel time.
4. Apply BFS or DFS traversal starting from any city to check whether all cities are reachable and determine whether the graph is connected.

Exemplars and Utility:

Exemplar:

1. Create a flight network using cities such as Mumbai, Delhi, Pune, Bangalore, and Chennai, with edge weights representing flight time in hours.
2. Mumbai connected to Delhi and Pune, Delhi connected to Bangalore, Pune connected to Bangalore, and Bangalore connected to Chennai using an adjacency list.
3. Prepare an adjacency matrix for the same flight network and use DFS starting from Mumbai to verify that all cities are visited.

Utility: This activity helps students understand graph representation, weighted edges, and connectivity using a real-world airline network example. It develops problem-solving skills useful in route planning, transportation networks, logistics, communication networks, and optimization-based computing applications.

Mini Project:

Sample topics are listed here for reference. Students may choose any one topic from the given list or select a similar topic that covers the concepts learned in this subject.



1. Student Database Management using Linked Lists
2. Smart E-Commerce Inventory & Order Manager
3. Smart Logistics & Delivery Route Planner
4. College Event Registration & Participant Management System

Text Books:

1. Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, “Fundamentals of Data Structures in C”, Publisher - Universities Press, 2nd Edition, 2008, ISBN-13: 978-8173716058, ISBN-10: 8173716056
3. Reema Thareja, “Data Structures Using C”, 2nd Edition, Oxford University Press, ISBN-13: 978-0-19-809930-7 ISBN-10: 0-19-809930-4
4. Varsha H. Patil, “Data Structures Using C++”, Oxford University Press, 2nd edition, ISBN: 978-0198066231

Reference Books:

1. Steven S. Skiena, “The Algorithm Design Manual”, Springer, 2nd edition, ISBN: 978-1-84800-069-8
2. Yashavant Kanetkar, “Let Us C”, 8th Edition, BPB Publications, ISBN: 9788183331777
3. Mark Allen Weiss, “Data Structures and Algorithm Analysis in C”, 2nd Edition, Pearson Education, ISBN: 978-8177583588
4. Aaron M. Tenenbaum, “Data Structures Using C”, 2nd Edition, Pearson Education, ISBN: 9788131711484

MOOCs Links and additional reading material:

1. First Mile Foundation Program – Programming and Data Structure and Algorithm
<https://nptel.ac.in/courses/106106133>
2. Data Structures and Algorithms
<https://nptel.ac.in/courses/106102064>
3. Data Structures and Program Methodology
<https://nptel.ac.in/courses/106103069>
4. Data Structures – I (Virtual Lab)
<https://ds1-iiith.vlabs.ac.in/List%20of%20experiments.html>

CO-PO-PSO Mapping

CO	PO1 EK	PO2 PA	PO3 DD	PO4 CPI	PO5 ETU	PO6 EW	PO7 Eth	PO8 ITW	PO9 Com	PO10 PMF	PO11 LL	PSO1 PFS	PSO2 PSS	PSO3 SCE
CO1	2	2	2	2	2	-	1	-	-	-	2	2	2	2
CO2	2	2	2	-	2	-	1	-	-	-	2	2	2	2
CO3	2	2	2	2	2	-	1	-	-	-	2	2	2	2
CO4	2	2	2	-	2	-	1	-	-	-	2	2	2	2
CO5	2	2	2	2	2	-	1	-	-	-	2	2	2	2
Avg.	2	2	2	2	2	-	1	-	-	-	2	2	2	2

**Assessment Pattern**

BT Level	CCE based on ARCV									
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9	Exp. 10
CO	CO1	CO1	CO2	CO2	CO3	CO3	CO4	CO4	CO5	CO5
L1										
L2	√	√	√	√	√	√	√	√	√	√
L3	√	√	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√	√	√
L5										
L6										

This course serves as a prerequisite for / maps with the following future courses:

1. Database Management Systems
2. Design & Analysis of Algorithm
3. Operating Systems
4. Artificial Intelligence

Job Mapping:

Job opportunities that one can get after learning this course: Software Developer, Backend Developer, Data Engineer, Machine Learning Engineer. These roles frequently use arrays, trees, graphs, hashing, and algorithms in real-world systems



Program:	S.Y. B. Tech				Semester:	III		
Course:	Object Oriented Programming Lab (AI&DS)				Course Code:	R25-AI-PCC-209		
Teaching Scheme					Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR	
-	2	-	01	-	-	25	25	
Course Description: This practical course introduces core Object-Oriented Programming (OOP) concepts through hands-on coding. Students learn to design and develop applications using principles like classes, objects, inheritance, polymorphism, encapsulation, abstraction and advanced java concepts. The course builds strong problem-solving and software development skills essential for modern engineers.								
Course Relevance: The course develops strong programming and problem-solving skills essential in software development and IT industries. It builds a solid foundation in Object-Oriented Programming, preparing students for entry-level developer roles and advanced studies in software engineering, web development, and AI applications.								
Prerequisite: 1. Programming & Problem Solving 2. Programming in C								
Bridge Content: Programming Concepts and Logic Building								
Course Objectives: 1. To implement programs using basic Object-Oriented Programming Constructs. 2. To develop real-world applications using constructors, arrays and strings. 3. To apply inheritance, and polymorphism to develop robust Java programs. 4. To use exception handling, collections and advanced java concepts to create real time applications. 5. To implement various operations and visualization on data using advanced java concepts.								
Course Outcomes: After learning the course, students will be able to								
CO	Course Outcome						Bloom's Level	
CO1	USE fundamental programming constructs such as control statements to develop Java applications.						3	
CO2	IMPLEMENT robust Java programs using constructors, arrays and strings.						3	
CO3	APPLY object-oriented features such as inheritance, polymorphism, interfaces and exception handling to develop real time applications.						3	
CO4	APPLY multithreading, collections and advanced java concepts to build robust Java applications.						3	
CO5	DEVELOP database-driven mini applications using MySQL/PostgresSQL and implement CRUD operations for efficient data management.						3	
Guidelines for Lab Conduction								



Set of suggested assignment list is provided in groups- A, B and C. Each student must perform at least 01 assignment from group A, any 08 assignments from group B and 01 from group C. One mandatory mini project from suggested list.

- Operating System recommended: - Microsoft Windows 10/11 (64-bit) or any open-source OS.
- Programming tools recommended: - Visual Studio Code or any other suitable development environments as applicable to the course.

Guidelines for Student's Laboratory Journal

- The laboratory assignments are to be submitted by students in the form of a journal.
- Journal consists of prologue, Certificate with table of contents, and write-up of each assignment
- (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Theory Concept in brief, algorithm, test cases, conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted.
- Attaching printed papers as part of write-ups and program listing to journals may be avoided.
- For reference, one or two journals may be maintained with program prints at Laboratory for accreditation purposes.

Guidelines for Laboratory/Term Work Assessment

- Continuous assessment of laboratory work should be done based on overall performance of students.
- Each Laboratory assignment assessment should be assigned grade/marks based on parameters with appropriate weightage.
- Suggested parameters for overall assessment as well as each Laboratory assignment assessment include timely completion performance, innovation, efficient codes, punctuality, and neatness.

Guidelines for Practical Examination

Both examiners should set the problem statements together, and assessment should focus mainly on correct implementation. Supplementary questions may be asked to check deeper understanding and efficiency. A fair, transparent, and encouraging evaluation process will build student confidence and support a strong academic start.

Group-A (Any 1)

Experiment No. 1	Implement a robust Java calculator program that dynamically captures user input, evaluates mathematical expressions using conditional logic and incorporates efficient error handling for invalid or exceptional inputs.	2 Hrs.	CO1
-------------------------	---	---------------	------------

Objective: To implement simple mathematical functions using basic constructs in OOP

Associated Tasks:

1. Accept numbers from the user dynamically.
2. Validate user input to ensure correct data types and allowed operations.
3. Use conditional statement switch to determine operations.
4. Implement loops for repeated calculations or menu-driven interaction.
5. Display the calculated result clearly.

Exemplars and Utility:

Exemplars:

1. Basic Calculator Application
2. Scientific Calculator Software



Utility: Prepares students to design interactive and robust applications that handle dynamic user input and errors gracefully.

Experiment No. 2	Create a BankAccount class with account number, account holder name, and account balance. Restrict direct access to the balance using encapsulation and provide methods to deposit money, withdraw money, and check balance. The program should accept user input dynamically and validate transactions by preventing negative deposits and over-withdrawals.	2 Hrs.	CO1
-------------------------	--	---------------	------------

Objective: To implement encapsulation in Java by designing a BankAccount class and performing validated banking transactions.

Associated Tasks:

1. Create a BankAccount class with account number, account holder name, and balance as private data members.
2. Implement methods to deposit, withdraw, and check balance.
3. Accept user input dynamically using Scanner class.
4. Validate transactions by preventing negative deposits and over-withdrawals.

Exemplars and Utility:

Exemplars:

1. ATM System
2. Internet Banking System
3. Banking Management Software

Utility: Students can apply this knowledge to develop secure financial or transaction-based applications using OOP principles.

Experiment No. 3	Develop a Java program for an E-commerce order processing where some products are initialized through multiple constructors, users can input some product details manually, the system computes total order cost dynamically, applies discount policies based on conditions, and presents a detailed invoice summarizing the purchase.	2 Hrs.	CO1
-------------------------	---	---------------	------------

Objective: To develop a Java program for an E-commerce order processing system using constructors, user input, and conditional logic to generate invoices and calculate discounts.

Associated Tasks:

1. Create a Product class with multiple constructors.
2. Accept product details from users.
3. Calculate total order cost dynamically.
4. Apply discounts based on conditions.
5. Display a detailed purchase invoice.

Exemplars and Utility:

Exemplars:

1. Online Shopping Cart
2. Billing and Invoice System
3. Retail Order Processing System

Utility: Helps automate order processing and billing while demonstrating practical use of constructors, user input handling, and conditional statements in Java.

R. Ganesh



Group-B (Any 08)			
Experiment No. 4	Write a program to create a class Employee with data members: id, name, and salary. Use constructor overloading to initialize objects in different ways: Default constructor, Constructor with name only, Constructor with name and salary, Constructor with all details. Display employee details and calculate annual salary.	2 Hrs.	CO2
Objective: To implement constructor overloading in Java by creating an employee class with different ways of object initialization and calculating annual salary. Associated Tasks: 1. Create an employee class with id, name, and salary. 2. Implement multiple constructors using constructor overloading. 3. Initialize employee objects with different parameter combinations. 4. Display employee details. 5. Calculate and display annual salary. Exemplars and Utility: Exemplars: 1. Employee Payroll System 2. HR Management Application 3. Staff Record Management System Utility: Demonstrates the use of constructor overloading, object initialization, and salary calculations in Java applications.			
Experiment No. 5	Develop a Java program that uses multidimensional arrays to perform various matrix operations. The program should accept input for two matrices from the user and validate their dimensions to ensure compatibility for different operations. Implement matrix addition, matrix multiplication, transpose of a matrix, and calculation of row-wise and column-wise sums.	2 Hrs.	CO2
Objective: To develop a Java program using multidimensional arrays for performing matrix operations and validating matrix dimensions. Associated Tasks: 1. Accept two matrices from the user. 2. Validate dimensions for operations. 3. Perform matrix addition and multiplication. 4. Find transpose of matrices. 5. Calculate row-wise and column-wise sums. 6. Display results in proper matrix format. Exemplars and Utility: Exemplars: 1. Student Result Analysis System 2. Digital Image Processing Software 3. Scientific Matrix Calculator 4. Computer Graphics Application Utility: This program helps in understanding multidimensional arrays, matrix manipulation techniques,			



and logical problem-solving used in real-world computational and analytical applications.			
Experiment No. 6	Write a program to accept a string from user and perform multiple operations: reversal, palindrome check, counting of vowels, consonants, digits, whitespace characters, and special symbols	2 Hrs.	CO2
Objective: To develop a Java program for performing various string operations using string handling techniques. Associated Tasks: 1. Accept a string from the user. 2. Reverse the string. 3. Check whether the string is a palindrome. 4. Count vowels and consonants. 5. Count digits, whitespace characters, and special symbols. 6. Display all results properly. Exemplars and Utility: Exemplars: 1. Text Analysis System 2. Password Validation Application 3. Data Cleaning Tool 4. Text Processing Software Utility: This program helps in understanding string manipulation, character analysis, and text-processing techniques commonly used in data validation, search systems, and language processing applications.			
Experiment No. 7	Write a Java program to create a base class Product with attributes such as product ID and name. Derive subclasses such as Electronics and Clothing to store category-specific detail.	2 Hrs.	CO3
Objective: To develop a Java program demonstrating inheritance by creating a base class Product and derived classes for different product categories. Associated Tasks: 1. Create a base class Product with common attributes. 2. Derive subclasses such as Electronics and Clothing. 3. Add category-specific attributes and methods. 4. Accept and display product details. 5. Demonstrate inheritance and code reusability Exemplars and Utility: Exemplars: 1. E-Commerce Product Management System 2. Online Shopping Application 3. Inventory Management Software 4. Retail Billing System Utility: Helps in understanding inheritance, class hierarchy, and code reusability concepts used in product management and real-world business applications.			



Experiment No. 8	Write a Java program to demonstrate inheritance using geometric shapes. Create a base class Shape and derive subclasses such as Circle, Rectangle, and Triangle. Implement methods to calculate area and display results using method overriding	2 Hrs.	CO3
Objective: To develop a Java program demonstrating inheritance and method overriding using different geometric shapes. Associated Tasks: 1. Create a base class Shape. 2. Derive subclasses Circle, Rectangle, and Triangle. 3. Implement methods to calculate area in each subclass. 4. Use method overriding for area calculation. 5. Display area results for different shapes. Exemplars and Utility: Exemplars: 1. Geometry Calculator 2. Computer Graphics Application 3. CAD Design Software 4. Engineering Drawing System Utility: helps in understanding inheritance, polymorphism, and method overriding concepts used in graphical, engineering, and mathematical applications.			
Experiment No. 9	Write a Java program to create two interfaces, Academic and Sports, each containing methods to store and display respective details. Implement both interfaces in a class Student to demonstrate multiple inheritance and display combined information.	2 Hrs.	CO3
Objective: To develop a Java program demonstrating multiple inheritance using interfaces. Associated Tasks: 1. Create interfaces Academic and Sports. 2. Define methods to store and display details in each interface. 3. Implement both interfaces in the Student class. 4. Accept and display combined student information. 5. Demonstrate the concept of multiple inheritance using interfaces. Exemplars and Utility: Exemplars: 1. Student Performance Management System 2. College Activity Record System 3. Sports and Academic Tracking Application 4. School Information System Utility: Helps in understanding interfaces, multiple inheritance, and implementation of shared functionalities in Java applications.			



Experiment No. 10	Develop a Java application that simulates an ATM machine. Implement functionalities like checking account balance, withdrawing, and depositing money. Use try, catch, and finally blocks to handle potential exceptions such as insufficient funds (throwing Arithmetic Exception) and invalid input (throwing Illegal Argument Exception). Ensure that the application continues to run smoothly after handling exceptions.	2 Hrs.	CO3
Objective: To develop a Java ATM simulation application using exception handling mechanisms for secure and reliable banking operations. Associated Tasks: 1. Implement functionalities for balance inquiry, withdrawal, and deposit. 2. Accept user input for ATM operations. 3. Use try, catch, and finally blocks for exception handling. 4. Handle insufficient balance using Arithmetic Exception. 5. Handle invalid input using Illegal Argument Exception. 6. Ensure smooth execution after exceptions occur. Exemplars and Utility: Exemplars: 1. ATM Banking System 2. Online Banking Application 3. Financial Transaction System 4. Digital Payment Processing System Utility: It helps in understanding exception handling, input validation, and secure transaction processing used in banking and financial software applications.			
Experiment No. 11	Design and develop a Java application that uses multithreading to simulate a simple chat system, where each thread represents an individual user responsible for sending and receiving messages. The program should use isAlive() to monitor thread activity and join() to synchronize thread execution. It should also demonstrate the use of thread priorities for handling important messages and include mechanisms for suspending, resuming, and terminating threads	2 Hrs.	CO4

**Objective:**

To develop a Java multithreading application that simulates a chat system using thread management and synchronization techniques.

Associated Tasks:

1. Create multiple threads representing chat users.
2. Implement message sending and receiving functionalities.
3. Use `isAlive()` to monitor thread status.
4. Use `join()` for thread synchronization.
5. Assign thread priorities for important messages.
6. Demonstrate thread suspension, resumption, and termination.

Exemplars:

1. Online Chat Application
2. Messaging System
3. Real-Time Communication Platform
4. Customer Support Chat System

Utility:

This program helps in understanding multithreading, thread lifecycle management, synchronization, and concurrent communication used in real-time software applications.

Experiment No. 12	Develop a simple Java program to manage an Employee Directory using the Collections Framework. Store employee details using collection classes such as ArrayList, HashSet, and HashMap. Use an Iterator to display employee records. Demonstrate the basic collection hierarchy.	2 Hrs.	CO4
--------------------------	---	---------------	------------

Objective:

To develop a Java program for managing an Employee Directory using the Collections Framework and demonstrating collection hierarchy concepts.

Associated Tasks:

1. Store employee details using ArrayList, HashSet, and HashMap.
2. Perform basic operations like add, store, and retrieve employee data.
3. Use Iterator to traverse and display records.
4. Demonstrate usage of different collection types.
5. Show basic collection hierarchy concepts.

Exemplars:

1. HR Employee Management System
2. Staff Directory Application
3. Company Resource Management System
4. Payroll Record System

Utility:

This program helps in understanding the Java Collections Framework, efficient data storage techniques, and real-world data management using dynamic data structures.

Group-C (Any 01)



Experiment No. 13	Develop a Java application that connects to a MySQL/PostgreSQL database to manage student information. Implement CRUD (Create, Read, Update, Delete) operations for student records using JDBC. Use prepared statements to handle SQL queries securely and ensure proper transaction management.	2 Hrs.	CO4
--------------------------	---	---------------	------------

Objective:

To develop a Java application using JDBC to connect with MySQL/PostgreSQL and perform secure database operations on student records.

Associated Tasks:

1. Establish database connectivity using JDBC.
2. Implement CRUD operations (Create, Read, Update, Delete) for student records.
3. Use Prepared Statement for secure SQL query execution.
4. Manage database transactions properly.
5. Handle exceptions and ensure data integrity.

Exemplars:

1. Student Database Management System
2. College Admission System
3. Library Management System
4. Academic Record Maintenance System

Utility:

This program helps in understanding database connectivity, secure data handling using JDBC, and real-world CRUD operations used in enterprise-level applications.

Experiment No. 14	Create a Java application that reads weather data from a CSV file and performs basic statistical operations using the Apache Commons Math library. Use JavaFX to create interactive charts and graphs to visualize temperature trends, humidity levels, and other weather parameters.	2 Hrs.	CO4
--------------------------	--	---------------	------------

Objective:

To develop a Java application for weather data analysis using CSV files, statistical methods, and JavaFX visualization.

Associated Tasks:

1. Read weather data from CSV file
2. Perform basic statistical analysis
3. Create charts using JavaFX
4. Visualize temperature and humidity trends

Exemplars:

1. Weather Forecast System
2. Climate Data Analysis Tool
3. Environmental Monitoring App
4. Data Visualization Dashboard

Utility:

This application helps in learning file handling, statistical analysis, and graphical data visualization using Java.



Experiment No. 15	Develop a Java application that reads patient data from a CSV file and calculates basic statistics such as average age, median heart rate, and standard deviation of blood pressure using the Apache Commons Math library. Implement functionality to visualize the data using JavaFX charts.	2 Hrs.	CO4
Objective: To develop a Java application for patient data analysis using CSV files, statistical computation, and JavaFX visualization. Associated Tasks: 1. Read patient data from CSV file 2. Compute basic statistics using Apache Commons Math 3. Visualize data using JavaFX charts Exemplars: 1. Patient Data Analysis System 2. Healthcare Monitoring Dashboard 3. Medical Statistics Tool 4. Clinical Data Visualization App Utility: This application helps in analyzing medical data using statistical techniques and visualizing results for better healthcare decision-making.			
Mini Project (Any 01)			
Design and Implement one Mini project using JSP/Servlet and JDBC. Maintain proper documentation and push code on GitHub repositories. 1. Student Management System to perform CRUD operations on student records with marks entry and result display. 2. Library Management System to manage books and members with CRUD operations, issue/return tracking, and fine calculation. 3. Banking System to perform CRUD operations on accounts with deposit, withdrawal, and transaction history. 4. Employee Management System to manage employee records with CRUD operations and salary calculation using JSP/Servlet. 5. Online Quiz System to perform CRUD operations on questions and display quiz results. 6. Hospital Management System to manage patient records and appointments using CRUD operations. 7. Inventory Management System to perform CRUD operations on products with stock tracking and updates. 8. Railway Reservation System to manage train, booking, and cancellation data using CRUD operations. 9. Hotel Booking System to perform CRUD operations on rooms, customers, and reservations. 10. E-Commerce System to manage products, customers, and orders with CRUD operations.		CO5	



Objective: To design and implement a mini-project using **JSP/Servlet** for user interface development and structured programming concepts to perform database-driven application development with CRUD operations.

Associated Tasks:

1. Design user-friendly **JSP/Servlet** interfaces for data input and display
2. Implement CRUD operations for managing application data
3. Validate and process user inputs efficiently
4. Maintain structured data flow between UI and backend logic
5. Display processed results in tabular or graphical format

Exemplars and Utility:

This mini-project helps in developing practical skills in GUI design, database interaction, and CRUD-based application development, which are widely used in real-world software systems for managing and processing data efficiently.

Over and Above (Suggested by Industry Expert)

Indian Bank has been in operation since 1900. All the work and transactions are recorded manually. Bank wants to get advanced and wants to use technology in order to provide users self-transactions services.

In Version 1 of their Banking Software, the bank wants to give support to the following functionalities:

- 1) Credentials verification (BankAccountNo, Password)
- 2) Deposit
- 3) Withdraw
- 4) Transfer
- 5) OTP generation

Write a program where all the above-mentioned functionalities are operational.

Instructions

1) Customers will have only two fields in the first version of the application bankAccountNo, password. All the values can be hardcoded.

2) The system will ask the user to enter the credentials, if credentials match, the code will display 4 options

1. Deposit
2. Withdraw
3. Transfer
4. Logout

3) Upon entering the option, the user should be able to perform the operations

4) For Transfer, the system will generate an OTP of 4 digits. Before transferring money, the user should verify the OTP.

5) Amount verification (withdrawal amount < balance) conditions must be addressed.

Text Books:

1. E Balaguruswamy, (2023). Programming with JAVA: A Primer. 7th edition. India: McGraw Hill Education
2. Herbert Schildt, (2021). Java: The complete reference, 13th edition. McGraw-Hill Education.

Reference Books:

1. Paul Deitel and Harvey Detail, Java: How to Program, Pearson's Publication, 9th Edition
2. Horstmann, C. S. (2023). Core Java - Vol. I – Fundamentals (Vol. 12). Pearson Education.

**MOOCs Links and additional reading material:**

1. Programming In Java: https://onlinecourses.nptel.ac.in/noc25_cs57/preview
2. Programming In Java: https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs36
3. Programming In Java: <https://nptel.ac.in/courses/106105191>
4. Java Database Connection: JDBC and MySQL: <https://www.udemy.com/course/how-to-connect-java-jdbc-to-mysql/>
5. java-the-complete-java-developer-course: <https://www.udemy.com/course/java-the-complete-java-developer-course/?couponCode=CM2600504IN>

CO-PO-PSO Mapping

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

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Assessment Pattern

BT Level	CCE based on ARCV									
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9	Exp. 10
CO	CO1	CO1	CO2	CO2	CO3	CO3	CO4	CO4	CO5	CO5
L1										
L2										
L3	√	√	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√	√	√
L5										
L6										

This course serves as a prerequisite for / maps with the following future courses:

1. Data science
2. Major Project
3. Web Technologies
4. Database Management Systems

Job Mapping:

Job opportunities that one can get after learning this course: Java Developer, Software Developer, Application Developer, Web Developer, Backend Developer, Software Tester, System Programmer, or Junior Software Engineer

Aeela

R. Ganesh



Program:	S. Y. B. Tech			Semester:			III
Course:	Community Engagement Project			Course Code:			R25-AI-CEP-210
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	OR
-	04	-	02	-	-	50	-

Course Description:

The purpose of this course is to empower students with the knowledge, skills, and mindset needed to leverage technology for addressing environmental and societal needs. Through a hands-on approach, students will learn to identify, analyze, and solve environmental issues faced by local communities using computer engineering and AI principles. By designing and implementing innovative solutions, students will not only gain technical proficiency but also develop teamwork, collaboration, and communication skills essential for making a positive impact on communities and the environment.

Course Relevance:

This course helps students apply their technical knowledge to solve real-world community problems, enhancing social responsibility, teamwork, and practical problem-solving skills required for professional and societal development.

Prerequisite:

Students should have prior knowledge of

1. Fundamental concepts of engineering and technology.
2. Basic understanding of social and ethical responsibilities.

Bridge Content:

- Understanding of simple survey methods, observation, and data recording
- Basic understanding of social issues, environmental concerns, and sustainable development.
- Problem-solving methodology.
- Computer and documentation Skills

Course Objectives:

- To develop social responsibility among students.
- To understand community problems and needs.
- To encourage students to apply their Artificial Intelligence knowledge and tools for solving real-life community issues.
- To improve communication, teamwork, and leadership skills.
- To promote sustainable development and community participation.

Course Outcomes: After learning the course, students will be able to

CO	Course Outcome	Bloom's Level
CO1	UNDERSTAND the concept and importance of community engagement.	2
CO2	IDENTIFY social and community problems through field visits and surveys.	3
CO3	IMPLIMENT problem-solving skills to address community issues.	3
CO4	APPLY Work effectively in teams and interact with community members.	3



CO5	IDENTIFY opportunities to contribute to the community's socioeconomic improvement.	3
Guidelines to Conduct the Course:		



1. A group of 3 to 4 students could be assigned for a particular habitation or village or municipal ward, as far as possible, in the near vicinity of their place of stay/college premise.
2. Each group /practical batch is allotted to a faculty member of the department as a mentor.
3. A division of 60 students can have 3 batches of minimum 20 students. Practical load of 4 hours to be allocated to each batch.
4. The group of students will be associated with a government official / village authorities /NGOs etc. concerned, allotted by the district administration, during the duration of the project.
5. The Community Engagement Project should be different from their regular programs of NSS/NCC/Green Club/Hobby Clubs, Special Interests Groups etc.
6. An activity book has to be maintained by each of the students to record the activities under taken/involved and will be countersigned by the concerned mentor/HoD.
7. An internal evaluation shall also be conducted by a committee constituted by the HoD. Evaluation to be done based on the active participation of the student and marks could be awarded by the mentor/HoD.
8. Student's groups can conduct an awareness program on Health and Hygiene or in Organic Farming or in Fisheries or in advocating prohibition of liquor or about renewable energy, e-waste management or any other activity in an area of their studies and as per his/her aptitude.
9. Final Assessment shall consist of presentation and demonstration of the project work carried out by the project groups.
10. Documentation and Reporting:
 - Students shall maintain a record of all meetings, remarks given by the guide/reviewers, and progress of the work in the project diary, presented during each review presentation.
 - For the final assessment, students shall complete the project report in all aspects, including formatting, duly signed by the project guide, Head of the Department.
11. Data Submission and Additional Outputs:
 - Students shall submit all data related to project work in soft copy to their guides, including the project report, A3 size poster, presentation, paper, etc., focusing on community engineering solutions.
12. AI Integration in Community Engagement Project
Students will use AI for Social Good to:
 - Identify community needs through data collection
 - Analyze problems using AI tools
 - Develop AI-assisted solutions
 - Create awareness on responsible AI
 - Promote digital literacy
 - Support agriculture, education, healthcare, and environmental sustainability
 - Assist citizens in accessing government services
 - Enhance community decision-making using AI insights

Suggestive list of topics under Community Engagement Project:



The below lists are not exhaustive and open for mentors to add, delete or modify. It is expected that the focus should be on specific local issues in their nearby areas. The students are expected to carry out these projects with involvement, commitment, responsibility and accountability. The mentors of a student/group of students shall

1. Interaction with Self Help Groups (SHGs) women members, and study their functions and challenges; planning for their skill-building and livelihood activities.
2. Visit Rural Schools/mid-day meal centres, study academic and infrastructural resources, digital divide and gaps or Visit local Anganwadi Centre and observe the services being provided.
3. Participate in Gram Sabha meetings, and study community participation.
4. Visit to local Nagarpalika office and review schemes for urban informal workers and migrants.
5. Visit local NGOs, civil society organizations and interact with their staff and beneficiaries.
6. Conduct soil health test, drinking water analysis, energy use and fuel efficiency surveys and building solar powered village.
7. Organize orientation programs for farmers regarding organic cultivation, rational use of irrigation and fertilizers, promotion of traditional species of crops and plants and awareness against stubble burning
8. Government Schemes Awareness - Educating citizens about different government welfare schemes.
9. Cyber Security Awareness - Awareness programs on online fraud, cybercrime, and safe internet practices
10. Environmental Awareness - Tree plantation drives and Awareness about plastic-free environment and waste management.
11. Education Support Activities - Organizing career guidance sessions, basic mathematics, science, or computer skills to school students
12. Traffic and Road Safety Awareness - Campaigns on helmet use, traffic rules, and accident prevention
13. Smart Agriculture Awareness - Educating farmers about AI-based agriculture tools and weather prediction apps.
14. Development of Simple Mobile App for Community Services - Designing a basic mobile app concept for reporting community issues or accessing services.

Course Content

Experiment No.1	Introduction to the Community Engagement Project (CEP)	2 Hrs.	CO1
Objective: Orientation on the course objectives, syllabus, Course Outcomes (COs), project methodology, ethical practices, and the significance of community engagement. Associated Task:			



- Understand the course objectives, syllabus, and Course Outcomes (COs).
- Study the concept, significance, and ethics of community engagement.
- Understand project guidelines, assessment criteria, and documentation requirements.
- Familiarize yourself with safety measures and code of conduct during field visits.

Experiment No.2	Formation of Student Project Groups	2 Hrs.	CO1
Objective: Formation of groups comprising 3–4 students and allocation of a community, village, municipal ward, or habitation for project implementation. Associated Task: • Form groups of 3–4 students. • Assign roles and responsibilities within the group. • Select or allocate the community/village/municipal ward. • Collect preliminary information about the assigned community. • Prepare a visit schedule.			
Experiment No.3	Identification of Community Domain and Area of Interest	2 Hrs.	CO2
Objective: Survey the assigned community to identify relevant social, educational, environmental, healthcare, agricultural, or technology-based areas requiring intervention. Associated Task: • Conduct an initial field visit. • Interact with community members and local authorities. • Observe existing facilities and challenges. • Identify possible domains such as education, healthcare, agriculture, environment, sanitation, digital literacy. • Record observations with photographs and field notes.			
Experiment No.4	Community Need Assessment and Problem Identification	2 Hrs.	CO2
Objective: Conduct field visits, interact with stakeholders, and finalize a problem statement based on the identified community needs. Associated Task: • Conduct surveys or interviews. • Design and use questionnaires for data collection. • Analyze collected information. • Identify and prioritize community problems. • Finalize the project topic and problem statement with faculty approval.			



Experiment No.5	Preparation and Submission of Project Proposal (Synopsis).	2 Hrs.	CO2
Objective: Prepare a project proposal outlining the title, objectives, methodology, implementation plan, expected outcomes, and required resources. Associated Task: ● Define project objectives. ● Prepare the implementation methodology. ● Develop a timeline and work plan. ● Estimate required resources and budget (if applicable). ● Submit the synopsis for approval.			
Experiment No.6	Implementation of the Community Engagement Activity.	2 Hrs.	CO3
Objective: Execute the planned intervention by applying appropriate technical, social, or AI-enabled solutions to address the identified problem. Associated Task: ● Execute the proposed solution/activity. ● Coordinate with community members and stakeholders. ● Collect implementation data and evidence. ● Record observations, photographs, video and attendance (if applicable). ● Monitor project progress and document outcomes.			
Experiment No.7	Organization of a Community Outreach Program.	2 Hrs.	CO4
Objective: Conduct any one of the following activities related to the project: ● Awareness Program ● Workshop ● Expert/Guest Lecture ● Demonstration or Training Session Associated Task: ● Plan and organize one activity such as an awareness program, workshop, guest lecture, demonstration, or training session. ● Prepare educational materials and presentations. ● Conduct the activity. ● Collect participant feedback. ● Document the event with photographs, video and reports.			
Experiment No.8	Project Presentation and Outcome Dissemination.	2 Hrs.	CO5
Objective: Present the project work, implementation process, outcomes, community feedback, and lessons learned before the evaluation committee. Associated Task: ● Prepare presentation slides. ● Summarize methodology, implementation, and outcomes.			

57





• Present community impact and lessons learned. • Answer questions from faculty and evaluators. • Incorporate suggestions for improvement.			
Experiment No.9	Impact Assessment, Feedback Collection and Sustainability Evaluation.	2 Hrs.	CO5
Objective: To evaluate the effectiveness, usability, and sustainability of the implemented engineering intervention through user feedback, field observations, and impact assessment. Associated Tasks: • Conduct post-implementation observation of the completed activity or solution. • Collect feedback from beneficiaries through interaction, surveys, or interviews. • Evaluate the impact based on usefulness, effectiveness, user satisfaction, safety, cost, and environmental benefits. • Compare the community's condition before and after the intervention and record key observations. • Prepare a concise impact assessment report, feedback summary, and recommendations. • Identify opportunities for improvement, sustainability, and replication of the project in other communities.			
Experiment No.10	Preparation and Submission of Final Project Report.	2 Hrs.	CO5
Objective: Compile a comprehensive report including problem identification, methodology, implementation, results, impact assessment, photographs, and future scope. Associated Tasks: • Compile all project documents. • Prepare the final report with introduction, literature review, methodology, results, impact assessment, conclusion, and future scope. • Attach survey forms, photographs, attendance records, and supporting documents. • Submit the report and complete viva-voce, if applicable.			
Proposal Submission:			
CEP Group should Submit a two-page project proposal, preferably prior to the term commencement outlining the following: • Title of the project • Aim, Objective and expected outcome • Plan of execution (timeline and activities). • Place of the CEP and involvement of any local authority, NGP • Required resources (if any). • Get approval from the designated faculty mentor.			
Text Books:			
1. Waterman, A. Service-Learning: A Guide to Planning, Implementing, and Assessing Student Projects. Routledge, 1997. 2. Beckman, M., and Long, J. F. Community-Based Research: Teaching for Community Impact. Stylus Publishing, 2016. 3. Design Thinking for Social Innovation. IDEO Press, 2015. 4. Dostilio, L. D., et al. The Community Engagement Professional's Guidebook: A Companion			



to The Community Engagement Professional in Higher Education. Stylus Publishing, 2017

Web Links:

1. UNESCO: Education for Sustainable Development <https://www.unesco.org>
2. EPICS (Engineering Projects in Community Service) <https://engineering.purdue.edu/EPICS>
3. Ashoka: Innovators for the Public <https://www.ashoka.org>
4. Design for Change <https://www.dfcworld.com>

MOOCs Links and additional reading material:

1. NPTEL course: Ecology and Society, https://onlinecourses.nptel.ac.in/noc20_hs77/preview

CO-PO-PSO Mapping

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

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Assessment Pattern

BT Level	CCE based on ARCV									
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9	Exp. 10
CO	CO1	CO1	CO2	CO2	CO3	CO3	CO4	CO4	CO5	CO5
L1										
L2	√	√	√	√	√	√	√	√	√	√
L3	√	√	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√	√	√
L5										
L6										

This course serves as a prerequisite for / maps with the following future courses:

This course provides practical exposure to real-world problem-solving and project implementation, which supports advanced courses such as Capstone Project, Project Management, Entrepreneurship Development, and Sustainable Development.

Job Mapping:

This course prepares students for roles such as Social Innovation Engineer, CSR Project Coordinator, Community Development Officer, Project Associate, and Entrepreneur by developing skills in problem-solving, teamwork, and real-world project implementation.



Semester - IV



Program:	S. Y. B. Tech			Semester:			IV
Course:	Artificial Intelligence			Course Code:			R25-AI-PCC-211
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
03	-	-	03	40	60	-	-

Course Description:

This course introduces the fundamental concepts, techniques, and applications of Artificial Intelligence systems. It covers intelligent agents, problem-solving methods, search algorithms, knowledge representation, reasoning, and basic learning techniques. The course provides both theoretical foundations and practical exposure to designing and implementing AI-based systems for real-world applications.

Course Relevance:

Artificial Intelligence is a core technology driving modern advancements in data science, automation, and intelligent decision-making systems. This course is highly relevant for Artificial Intelligence and Data Science engineering students as it builds a strong foundation for advanced subjects such as Machine Learning, Deep Learning, Natural Language Processing, and Computer Vision. The knowledge gained enables students to develop intelligent solutions across domains like healthcare, finance, robotics, and cybersecurity, thereby enhancing industry readiness and research capabilities.

Prerequisite:

1. Probability and Statistics.
2. Data structures and Algorithms.

Bridge Content:

Review of prerequisite concepts including programming basics, logical reasoning, and introductory data structures relevant to Artificial Intelligence.

Course Objectives:

1. To study the concept of Artificial Intelligence through the study of various intelligent tasks.
2. To implement problem-solving approaches in AI using different search strategies.
3. To understand the functioning of multi-agent systems in competitive environments.
4. To study the fundamentals of knowledge representation and reasoning.
5. To design effective plans of action for achieving goals as a key component of AI.

Course Outcomes: After learning the course, students will be able to

CO	Course Outcome	Bloom's Level
CO1	EXPLAIN the Fundamentals, Scope, and Applications of Artificial Intelligence and Intelligent Systems.	2
CO2	APPLY Problem-Solving Techniques and Search Strategies to Solve Well-Defined AI Problems.	3
CO3	APPLY Adversarial Search Techniques and Constraint Satisfaction Methods to Solve Game-Playing and Constraint-Based Problems.	3
CO4	EXPLAIN Knowledge Representation Techniques to Model Real-World Problems in Artificial Intelligence.	2

Acela

R. Ganesh



CO5	APPLY Reasoning Mechanisms and Basic Inference Techniques to Solve Artificial Intelligence Problems.	3
Course Content		
Unit 1	Introduction to Artificial Intelligence	9 Hrs. CO1
Introduction: What is AI, The History of AI, Types of AI, Narrow AI, General AI, and Super AI, Applications of AI, Limits of AI, Ethics of AI, AI Tools & Frameworks, The State of Art, Future of AI? AI bias, Explainable AI (XAI), Responsible AI, AI hallucinations Intelligent Agent: Introduction of agents, Structure of Intelligent Agent, Characteristics of Intelligent Agents, Types of Agents, Types of Environments, Generative AI, Large Language Models (LLMs), Prompt Engineering. Exemplars / Practical Applications and usage: Study How Recommendation Systems Suggest Products, Movies, Music, short videos etc Based on User Behavior, Preferences, and Feedback.		
Unit 2	Problem Solving	9 Hrs. CO2
Solving Problems by Searching: Problem-Solving Agents, Example Problems, Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions. Beyond Classical Search: Local Search Algorithms and Optimization Problems, searching with nondeterministic actions, Searching with partial observation, Online search agent and unknown environment. Exemplars / Practical Applications and usage: Demonstrates uninformed and informed search strategies, evaluates solution optimality, and analyzes the complexity of problem spaces using examples such as the 8-puzzle problem, water jug problem, travelling salesman problem, and maze/pathfinding problems etc.		
Unit 3	Adversarial Search and Constraint Satisfaction	9 Hrs. CO3
Adversarial Search: Games Theory, Optimal Decisions in Games, Heuristic Alpha–Beta Tree Search, Monte Carlo Tree Search, Stochastic Games, Partially Observable Games, and Limitations of Game Search Algorithms. Constraint Satisfaction Problem: Constraint Satisfaction Problems (CSP), Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs, Local Search for CSPs. Exemplars / Practical Applications and usage: Design an AI system to generate an Exam Timetable for a college such that: No student has overlapping exams, A teacher is not assigned to two exams at the same time, Exam halls are not double-booked, Exams are evenly distributed across available time slots		
Unit 4	Knowledge	9 Hrs. CO4
Logical Agents: Knowledge-Based Agents, The Wumpus World Problem, and Propositional Logic, A Very Simple Logic, Propositional Theorem Proving, Effective Propositional Model Checking, Agents Based on Propositional Logic. First Order Logic: First-Order Logic, Representation Revisited, Syntax and Semantics of First Order Logic, Ontologies in AI, Using First-Order Logic, Knowledge Engineering in First-Order Logic. Exemplars / Practical Applications and usage: Study and Write Case Study for Netflix: AI-Driven Personalized Content Recommendation Using Knowledge Representation and Inference.		
Unit 5	Reasoning and Planning	9 Hrs. CO5



Inference in First-Order Logic: Inference in First-Order Logic, Propositional vs. First-Order Inference, Unification and First-Order Inference, Forward Chaining, Backward Chaining, Resolution, Knowledge Representation.

Planning: Automated and Classical Planning, Hierarchical Planning, planning under Uncertainty, planning to real business and Engineering workflows, Analysis of Planning Approaches.

Exemplars / Practical Applications and usage:

Case Study on AI-Based Smart Agriculture Planning using AI planning techniques in emerging sustainable technologies.

Textbooks:

1. Stuart Russell and Peter Norvig, “Artificial Intelligence: A Modern Approach”, Third Edition, Pearson, 2003, ISBN :10:0136042597
2. Deepak Khemani, “A First Course in Artificial Intelligence”, McGraw Hill Education (India), 2013, ISBN :978-1-25-902998-1
3. Elaine Rich, Kevin Knight and Nair, “Artificial Intelligence”, TMH, ISBN-978-0-07-008770-5

Reference Books:

1. George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education, 6th ed., 2009.
2. K.R.Chowdhary, “Fundamentals of Artificial Intelligence”, Springer, 2020, ISBN-13 978-8132239703.

MOOCs Links and additional reading material:

1. Introduction to AI.
https://youtu.be/pKeVMlkFpRc?list=PLwdnzlV3ogoXaceHrrFVZCJkbm_laSHcH
2. Informed Search.
https://youtu.be/-Rf2hOyjZB8?list=PLwdnzlV3ogoXaceHrrFVZCJkbm_laSHcH
3. Uniformed Search.
https://youtu.be/te1K8on1Pk0?list=PLwdnzlV3ogoXaceHrrFVZCJkbm_laSHcH
4. Constraint Satisfaction Problems.
https://youtu.be/-6FRXK7RwJE?list=PLwdnzlV3ogoXaceHrrFVZCJkbm_laSHcH
5. Game Playing.
https://youtu.be/DeVA3ccObkA?list=PLwdnzlV3ogoXaceHrrFVZCJkbm_laSHcH

CO-PO-PSO Mapping:

CO	P01	P02	P03	P04	P05	P06	P07	P08	P09	P010	P011	PSO1	PSO2	PSO3
CO1	3	2	-	3	1	1	1	1	–	–	3	3	2	2
CO2	3	3	2	3	2	1	–	–	1	–	3	3	3	2
CO3	2	3	3	2	2	1	–	1	1	–	3	3	3	1
CO4	3	2	2	3	1	2	–	2	–	–	3	3	3	2
CO5	2	2	1	2	1	2	1	3	–	–	3	2	3	3
Avg.	2	2	2	2	1	1	1	2	1	-	3	3	3	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**Unit wise Weightage:**

Exam	Unit1	Unit2	Unit3	Unit4	Unit5
CCE					
CCE1	8	8			
CCE2			8	8	
CCE3					8
ESE	12	12	12	12	12
Total	20	20	20	20	20

Assessment Pattern

BT Level	CCE			ESE				
	CT1	CT2	CCE3					
CO				CO1	CO2	CO3	CO4	CO5
L1								
L2	√	√		√	√	√	√	√
L3	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√
L5								
L6								

This course serves as a prerequisite for / maps with the following future courses:

1. Machine Learning
2. Deep Learning
3. Natural Language Processing
4. Data Science and Analytics

Job Mapping:

Job opportunities that one can get after learning this course:

Database Administrator (DBA)

Database Developer

Data Analyst

Backend Developer

System Analyst



Program:	S. Y. B. Tech			Semester:	III		
Course:	Data Science			Course Code:	R25-AI-PCC-212		
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
02	-	-	02	40	60	-	-
Course Description: This course introduces the fundamentals of Artificial Intelligence (AI) and Data Science (DS), covering core concepts, tools, and real-world applications. It explains how AI and DS technologies work across various domains and addresses key ethical considerations.							
Course Relevance: This course aligns with current industry needs by introducing modern tools and methodologies used in AI and Data Science. It helps students understand the roles, responsibilities, and essential skills required for careers in this growing field, thereby preparing them for future opportunities in AI-driven industries and research.							
Prerequisite: 1. Computational Statics. 2. Data Structure 3. Emerging Trends							
Bridge Content: Program Problem Solving.							
Course Objectives: 1. To understand the fundamental concepts of Data Science, data types, and data collection methods. 2. To apply data preprocessing and data wrangling techniques to clean and prepare datasets for analysis 1. To analyze data using statistical methods and data visualization tools. 2. To analyze predictive models using machine learning algorithms. 3. To use modern data science tools and technologies to solve real-world problems and support data-driven decision making.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome						Bloom's Level
CO1	UNDERSTAND the fundamental concepts of Data Science, data types, and the data science lifecycle						2
CO2	PERFORM data preprocessing and data wrangling techniques to prepare datasets for analysis						3
CO3	ANALYZE life cycle of data analytics to provide a structure framework of data.						4
CO4	ANALYZE basic machine learning models for prediction and classification tasks.						4
CO5	USE modern Data Science tools and technologies to extract insights and support data-driven decision making.						3
Course Content							
Unit 1	Foundations of Data Science					6 Hrs.	CO1

Aeela

R. G. Gaurishankar



Definition of Data Science, Data Types in Data Science, Applications of data science, Business intelligence versus Data Science, Scales of measurement: Nominal, Ordinal, Interval, Ratio, Data quality dimensions: Accuracy, Completeness, Consistency, Timeliness, Difference Between Data, Information, Knowledge, wisdom. Metrics: Confusion Matrix, Accuracy, Precision-Recall, ROC-AUC, F1 Score, Mean Squared Error (MSE), R-squared. Exemplar/Case Studies: Create a multi-source student academic dataset containing missing values, duplicate records, inconsistent grading formats, and outliers. Perform data cleaning, transformation, and evaluation metric analysis to improve prediction reliability and analytical accuracy.			
Unit 2	Mathematical Foundations For DS	6 Hrs.	CO2
Vectors: Vector spaces, vector addition, scalar multiplication, Geometric interpretation of vectors (in 2D and 3D space). Dot product, Cross product, Vector norms, Linear combinations, linear independence, span, basis, Linear Transformations: Definition of linear transformations, Matrix representation of linear transformations, Change of basis. Exemplar/Case Studies: Create a retail sales dataset and perform statistical analysis, correlation analysis, and vector operations to identify customer purchasing patterns and sales trends.			
Unit 3	Data Science Life Cycle and Manipulation.	6 Hrs.	CO3
The Data Science Process/Lifecycle (e.g., CRISP-DM, Team Data Science Process (TDSP), SEMMA Methodology), Data Wrangling: Data collection, Data cleaning (missing values, outliers), Data transformation, Handling Missing Values, Data Integration, Data Reduction, Data Aggregation, Data Filtering and Sorting, Feature engineering, Data Validation. Exemplar/Case Studies: Create a smart healthcare dataset using patient records, laboratory reports, and sensor data, and perform data wrangling, integration, and feature engineering to support disease prediction analytics.			
Unit 4	Data Preprocessing and Visualization	6 Hrs.	CO4
Data Cleaning: Handling Missing Values , Outlier Detection and Treatment, Data Type Conversion, Data Consistency and Validation: Standardizing Values , Identifying and Correcting Errors, Data Transformation :- Feature Scaling and Normalization, Standardization (Z-score Normalization), Min-Max Scaling , Robust Scaling ,Data Visualization: Principal Component Analysis (PCA), Exploratory Data Analysis (EDA), Visualizing distributions, correlations, Visualization libraries: Matplotlib, Seaborn, Plotly. Exemplar/Case Studies: Create a customer behavior dataset for an online shopping platform and perform preprocessing, normalization, clustering, PCA, and visualization techniques to analyze purchasing patterns and customer segments.			
Unit 5	Automating AI Workflows with Pandas	6 Hrs.	CO5
ETL, ETL Challenges, ETL in Data Preprocessing for AutoML, AutoML: AutoML Workflow, AutoML Libraries: Auto-sklearn, H2O.ai, and TPOT, Benefits and Limitations of AutoML, AutoML vs Traditional Machine Learning, Pandas for data preprocessing for AutoML, data normalization and standardization for AutoML, Introduction to Big Data: Characteristics, Architecture and Ecosystem, Tools and Technologies, Applications and Challenges of Big Data in Data Science. AI-powered Data Cleaning: Using AI models to clean and structure raw data before analysis.			

**Exemplar/Case Studies:**

Create a customer behavior dataset for an online shopping platform and perform preprocessing, normalization, clustering, PCA, and visualization techniques to analyze purchasing patterns and customer segments.

Textbooks:

1. David Dietrich, Barry Hiller, “Data Science and Big Data Analytics”, EMC education services, Wiley publication, 2012, ISBN0-07-120413-X.
2. McKinney W., “Python for Data Analysis: Data Wrangling with Pandas, NumPy and I Python”. F2nd edition, O’Reilly Media.
3. Jiawei Han, Micheline Kamber, and Jian Pie, “Data Mining: Concepts and Techniques” Elsevier Publishers Third Edition, ISBN: 9780123814791, 9780123814807.

Reference Books:

1. EMC Education Services, “Data Science and Big Data Analytics- Discovering, analyzing Visualizing and Presenting Data” 1st Edition.
2. DT Editorial Services, “Big Data, Black Book”, DT Editorial Services, ISBN: 9789351197577, 2016 Edition.
3. Chirag Shah, “A Hands-On Introduction to Data Science”, Cambridge University Press, (2020), ISBN: ISBN 978-1-108-47244-9.

MOOCs Links and additional reading material:

1. Data Science. https://onlinecourses.nptel.ac.in/noc21_cs69/preview
2. Data Science. <https://archive.nptel.ac.in/courses/110/106/110106072/>

CO-PO-PSO Mapping:

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

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Unit wise Weightage:

Exam	Unit1	Unit2	Unit3	Unit4	Unit5
CCE					
CT1	8	8			
CCE2			8	8	
CCE3					8
ESE	12	12	12	12	12
Total	20	20	20	20	20

**Assessment Table:**

BT Level	CCE			ESE				
	CCE1	CCE2	CCE3	CO1	CO2	CO3	CO4	CO5
CO								
L1								
L2	√	√		√	√	√	√	√
L3	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√
L5								
L6								

This course serves as a prerequisite for / maps with the following future courses:

1. Machine Learning
2. Deep Learning
3. Artificial Neural Network

Job Mapping:

Job opportunities that one can get after learning this course: Data Scientist, Data Analyst, Data Engineer, Machine Learning Engineer, AI Specialist.

Acela

R. Ganesh



Program:	S. Y. B. Tech			Semester:		III	
Course:	Database Management Systems			Course Code:		R25-AI-PCC-213	
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
03	-	-	03	40	60	-	-
Course Description: This course introduces both fundamental and advanced concepts of Database Management Systems. It covers data models, Entity–Relationship and Extended ER design, Relational database concepts, Normalization techniques and the use of SQL for database definition and manipulation. The course also addresses transaction management, concurrency control, recovery mechanisms, and provides an introduction to NoSQL databases.							
Course Relevance: The course focuses on the principles of database modelling and design, the languages and data models supported by modern database management systems and essential database implementation techniques. It aims to provide an in-depth and contemporary understanding of core database system concepts, architectures and applications, equipping students with the skills required to design, develop, and manage efficient, secure and scalable data-driven systems.							
Prerequisite: 1. Principles of Problem Solving 2. Digital Technique Lab							
Bridge Content: Fundamentals of Programming Language							
Course Objectives: 1. To understand the fundamentals of various data modelling techniques used in database systems. 2. To design relational database management systems for real-world applications. 3. To analyze the theoretical foundations of database systems and the factors influencing their functionality and performance. 4. To use query languages effectively, incorporating semantic concepts for query optimization. 5. To familiarize students with emerging trends and recent advancements in data management systems.							
Course Outcomes: After learning the course, students will be able to:							
CO	Course Outcome					Bloom’s Level	
CO1	DESIGN effective data models using ER/EER concepts to meet organizational data requirements.					4	
CO2	DEVELOP relational database systems by writing efficient SQL queries and PL/SQL program units.					4	
CO3	ORGANIZE database tables using normalization methods to maintain data integrity in real-time databases.					4	
CO4	EVALUATE NoSQL with MongoDB operations for scalable intelligent data processing.					4	



CO5	APPLY Big Data concepts using Hadoop and Map-Reduce and analyze emerging database technologies for large-scale data management.	3	
Course Content			
Unit 1	Introduction to DBMS and Data Models	9 Hrs.	CO1
Introduction: Basic concepts, Advantage of DBMS over file processing system, Data Abstraction, Database Language, Structure of DBMS, Data Modeling, Database applications, Database-system Architecture. Database Design: Entity, Attributes, Relationships, Constraints, Keys, Design Process and Issues. Data Models: Entity Relationship (ER), ER model, Mapping ER Model to Relational Model, ER Diagram, Design issues. Relational and Object-Oriented Data Models, Integrity Constraints and Data Manipulation Operations, ER to Table Conversion.			
Exemplars / Practical Applications and usage: An illustration of a Banking and ATM system using ER diagrams and relational mapping to explain database design concepts to students.			
Unit 2	Advanced SQL And PLSQL	9 Hrs.	CO2
SQL: Introduction to SQL, SQL Data types and Literals, SQL Commands, SQL Select Query and Clauses. SQL Advanced Features: Set Operation, Aggregate Function, Null Values, Nested Sub Query, View, Joins, Sequence, Index, Introduction to Embedded and Dynamic SQL. Introduction to PL/SQL: Data types, Procedures, Functions, Cursor, Trigger, Package, Assertions, Roles and Privileges.			
Exemplars / Practical Applications and usage: An example demonstrating complex SQL queries using joins, sub queries, set operations, and aggregate functions to help students understand advanced database querying techniques.			
Unit 3	Relational Database Design and Transaction Management	9 Hrs.	CO3
Relational Database Design: Basic Concepts, Attribute and Domains, CODD’S Rules, Functional Dependency, Normalization: 1NF, 2NF, 3NF and BCNF Transactions: Basic Concepts, States, ACID properties, schedules. Serializability: Conflict and view serializability, cascaded aborts, recoverable and nonrecoverable schedules, Deadlock prevention, Deadlock Detection and Recovery from deadlock.			
Exemplars / Practical Applications and usage: Banking and ATM systems illustrating secure and consistent financial transactions in real-world database applications.			
Unit 4	NoSQL	9 Hrs.	CO4
Data types: Structured, unstructured, and semi-structured data. NoSQL databases: Types (Key value, Document, Graph, Wide-column), BASE properties, ACID vs. BASE, RDBMS vs. NoSQL MongoDB: CRUD operations, indexing, aggregation, MapReduce, Replication, Sharding			
Exemplars / Practical Applications and usage: E-commerce platforms using NoSQL databases to manage large-scale product catalogs, user sessions, and shopping carts.			
Unit 5	Emerging Database Technologies	9 Hrs.	CO5



Introduction to Big Data, handling large datasets using Map-Reduce and Hadoop.
Introduction to HBase data model and HBase region.

Introduction to emerging database technologies – Cloud Databases, PostgreSQL, Mobile Databases, SQLite Database, XML Databases, DynamoDB.

Exemplars / Practical Applications and usage:

XML databases for structured data representation and data exchange in web services and enterprise integration systems.

Textbooks:

1. Silberschatz A., Korth H., Sudarshan S., "Database System Concepts", McGraw Hill 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, Begg C., "Database Systems –A Practical Approach to Design, Implementation and Management", Pearson Education, 5th Edition, 2010, ISBN 81-7808-861-4.
4. Pramod J. Sadalage and Martin Fowler, "NoSQL Distilled", Addison Wesley, ISBN10: 0321826620, 2013, ISBN13: 978-0321826626.

Reference Books:

1. S. K. Singh, *Database Systems: Concepts, Design and Application*, Pearson Education, ISBN 978-81-317-6092-5.
2. Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining: Concepts and Techniques*, Elsevier.
3. *Big Data: Understanding How Data Powers Big Business*, Bill Schmarzo, Wiley 3 *Hadoop: The Definitive Guide*, Fourth Edition, Tom White O'Reilly.
4. *HBase: The Definitive Guide*, Fourth Edition, O'Reilly Lars George O'Reilly edidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum - *Data Structures Using C and C++*, Pearson Education.

MOOCs Links and additional reading material:

1. Introduction to Database Management System
https://onlinecourses.nptel.ac.in/noc26_cs39/course
2. Introduction to NoSQL
<https://www.mongodb.com/resources/basics/databases/nosql-explained%20>
3. Unified AI Database: <https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/modeling-data>
4. Data Base Management System: https://onlinecourses.nptel.ac.in/noc22_cs91/preview

CO-PO-PSO Mapping:

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

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**Unit wise Weightage:**

BT Level	CCE based on ARCV				
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
CO	CO1	CO1	CO2	CO2	CO3
L1					
L2	√	√	√	√	√
L3	√	√	√	√	√
L4	√	√	√	√	√

Assessment Pattern

BT Level	CCE						ESE				
	CCE1	CCE2	CCE3	CCE4							
CO							CO1	CO2	CO3	CO4	CO5
L1											
L2	√	√					√	√	√	√	√
L3	√	√	√				√	√	√	√	√
L4	√	√	√				√	√	√	√	
L5											
L6											

This course serves as a prerequisite for / maps with the following future courses:

1. Data Warehousing and Data Mining
2. Big Data Analytics
3. Machine Learning
4. Cloud Computing and Cloud Databases

Job Mapping:

Job opportunities that one can get after learning this course: Database Administrator (DBA), Database Developer / SQL Developer, Data Analyst, Cloud Database Engineer, NoSQL Database Developer, Backend / Full-Stack Developer.

Heela

R. Ganesh



Program:	S. Y. B. Tech			Semester:	IV		
Course:	Embedded System & Internet of Things			Course Code:	R25-AI-MDM-214		
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
02	-	-	02	40	60	-	-
Course Description: This course focuses on the principles and applications of embedded systems and IoT, including hardware–software integration, sensor and device interfacing, programming with Arduino and Raspberry Pi, and implementing intelligent solutions using basic AI techniques.							
Course Relevance: This course is highly relevant for developing skills in designing and implementing embedded and IoT-based intelligent systems, which are widely used in automation, smart devices, robotics, and real-time applications across industries.							
Prerequisite: 1. Fundamentals of Electronics Engineering 2. Emerging Trends in AIDS							
Bridge Content: Foundational knowledge of basic electronics, digital logic, and programming with advanced concepts in embedded systems, IoT, and AI with C/Python programming to enable students to design intelligent, real-world embedded and IoT applications.							
Course Objectives: This course aims to ● Explain the fundamentals of embedded systems, including design methodologies and system modeling techniques. ● Describe embedded system architecture and demonstrate the selection of appropriate processors and memory, along with understanding the embedded software build process. ● Apply interfacing techniques and analyze communication protocols and networking methods used in embedded and IoT systems. ● Develop and implement IoT-based applications using Arduino and Raspberry Pi platforms. ● Explain basic AI concepts and evaluate their application in developing intelligent embedded and IoT-based solutions.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome						Bloom’s Level
CO1	EXPLAIN the fundamentals of embedded systems, including their architecture, design methodologies, and modeling techniques.						2
CO2	SELECT appropriate processors, memory components, and development tools, and demonstrate the embedded system build process.						3
CO3	INTERFACE memory, I/O devices, sensors, and implement communication protocols in embedded and IoT systems while analyzing their performance.						4
CO4	DEVELOP and program embedded and IoT applications using Arduino and Raspberry Pi with effective sensor and device						3



	integration.		
CO5	DEMONSTRATE the use of RTOS and AI-enabled processors for intelligent and time-critical applications for real-time embedded systems.	3	
Course Content			
Unit 1	Introduction to Embedded Systems	6 Hrs.	CO1
Introduction to embedded system, application domain, Desirable Features and General Characteristics of Embedded Systems, Example of a Simple Embedded System, Figures of Merit for an Embedded System, Classification of MCUs: 4/8/16/32 Bits, History of Embedded Systems, classification of embedded system, Current Trends, example of embedded system: -Mobile Phone, Automotive Electronics, Radio Frequency Identification (RFID), Robotics			
Exemplars / Practical Applications and usage: Design and Configure programming model for GPIO in ARM7 Implement LED Blinking on pressing a switch using GPIO			
Unit 2	Microcontrollers in Embedded System	6 Hrs.	CO2
History of Microcontrollers, Introduction to Arduino, Components of Arduino, History of Arduino, Installing Software, Structure of Programming, Overview of Variables, Infinite Loops, Compiling, Linking and Debugging. 67 Raspberry Pi, About the Board, Linux on Raspberry Pi, Raspberry Pi Interfaces, Programming Raspberry Pi with Python. Controlling LED with Raspberry Pi, Interfacing an LED and Switch with Raspberry Pi, Interfacing a Light Sensor (LDR) with Raspberry Pi.			
Exemplars / Practical Applications and usage: Design and implement Temperature control circuit using LM 35 and DC motor (PWM Generation)			
Unit 3	Sensors, ADCs and Actuators	6 Hrs.	CO3
Basics of Sensors and Actuators, Active vs. passive sensors, Analog vs. digital sensors, Types of Sensors: Temperature Sensors, Light Sensors, Photo unction Devices, Proximity/Range Sensors, Humidity Sensors, pressure sensor, ultrasonic sensor, gas sensors, Analog to Digital Converters. Digital to Analog Converters, Types of Actuators, Displays-LEDs, Liquid Crystal Displays (LCDs), seven segment display, stepper motor, Relay			
Exemplars / Practical Applications and usage: Design and configure GPIO for Keypad and LCD interfacing with ARM7 Implement Keypad and LCD interfacing on LPC2148			
Unit 4	ARM Processor	6 Hrs.	CO4
Introduction, RISC design philosophy, ARM design philosophy, Embedded system hardware– AMBA bus protocol, ARM bus technology, Memory, Peripherals, Operating System, Applications. ARM Processor Fundamentals, ARM core dataflow model, registers, current program status register, Pipeline, Exceptions, Interrupts and Vector Table, Core extensions. Real time Scheduling Algorithms, Rate Monotonic Algorithm, The Earliest Deadline First Algorithm			
Exemplars / Practical Applications and usage: Study of different operating systems for Raspberry-Pi /Beagle board/Arduino. Understanding the process of OS installation			
Unit 5	Emerging Processors for AI	6 Hrs.	CO5



Introduction to AI-focused processors: CPU (Central Processing Unit), GPU (Graphics Processing Unit), TPU(Tensor Processing Unit), NPU (Neural Processing Unit), VPU (Vision Processing Unit). Comparison: Intel Core Ultra, Apple M3, NVIDIA RTX, Google TPU, AMD Ryzen AI, Qualcomm Hexagon. Concepts of Edge AI & Cloud AI processing. Heterogeneous computing (CPU+GPU+NPU).

Exemplars / Practical Applications and usage:

Write a program to control the color of the LED by using potentiometer.

Textbooks:

1. Raj Kamal, “Embedded systems Architecture, Programming and Design”, Tata McGraw- Hill, 2016.
2. Wayne Wolf “Computers as components: Principles of Embedded Computing System Design”, The Morgan Kaufmann Series in Computer Architecture and Design, 2013.
3. “Arduino for beginners: Essential Skills Every Maker Needs”, John Baichtal, Person Education, Inc., 1st Edition.
4. “Programming with Raspberry Pi: Getting Started with Python”, Simon Monk, January 2012, McGrawHill Professional.

Reference Books:

1. Lyla B. Das, " Embedded Systems an Integrated Approach", Pearson Education, 2013.
2. Shibu K V, " Introduction to Embedded Systems", McGraw Hill Education (India) Private Limited, 2014
3. Sriram V Iyer, Pankaj Gupta " Embedded Real Time Systems Programming", Tata McGraw-Hill, 2012
4. Steve Heath, “Embedded Systems Design”, EDN Series, 2013.
5. “Arduino Cookbook”, Michael Margolis, O’Reilly Media, Inc., 1st Edition.
6. “Raspberry Pi User Guide”, Eben Upton and Gareth Halfacree, August 2016, 4th Edition, John Wiley & Sons.

MOOCs Links and additional reading material:

1. Introduction to Embedded system Design:
https://onlinecourses.nptel.ac.in/noc20_ee98/preview
2. Introduction to Internet of Things: <https://courses.mooc.fi/org/uh-cs/courses/introduction-to-the-internet-of-things-mooc>
3. Internet of Things: Design Concepts and Use Cases:
https://onlinecourses.swayam2.ac.in/ntr24_ed01/preview
4. Embedded Systems Design and IoT Integration:
<https://www.eicta.iitk.ac.in/courses/embedded-system-course>
5. Introduction to IOT:
https://onlinecourses.nptel.ac.in/noc19_cs65/preview?utm_source=chatgpt.com
1. Embedded System, IoT & AIoT- <https://www.udacity.com/course/embedded-systems-IoT-and-AIoT--cd12870>

**CO-PO-PSO Mapping:**

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	1	1	2	1	--	--	--	3	2	1	1
CO2	3	2	3	1	3	1	1	--	--	1	3	3	3	2
CO3	3	2	2	1	2	2	1	--	--	--	3	2	2	1
CO4	3	2	1	1	1	1	1	--	--	1	3	2	1	1
CO5	3	3	2	2	1	1	1	--	--	1	3	2	2	2
Avg.	3	2	2	2	2	2	2	0	0	1	3	2	2	1

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Unit wise Weightage:

Exam	Unit1	Unit2	Unit3	Unit4	Unit5
CCE					
CCE1	8	8			
CCE2			8	8	
CCE3					8
ESE	12	12	12	12	12
Total	20	20	20	20	20

Assessment Table:

BT Level	CCE			ESE				
	CCE1	CCE2	CCE3	CO1	CO2	CO3	CO4	CO5
CO								
L1								
L2	√	√		√	√	√	√	√
L3	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√
L5								
L6								

This course serves as a prerequisite for / maps with the following future courses:

- Robotics and Automation
- Edge AI and Intelligent Systems
- Machine Learning for Embedded Systems

Job Mapping:

Students gain competencies relevant to careers in embedded systems & IoT as

1. Embedded & IoT Systems Engineer
2. Application Developer
3. Hardware Design Engineer
4. IoT Security Specialist Embedded Linux Developer

Aeeha

R. Ganesh



Program:	S. Y. B. Tech			Semester:	IV		
Course:	Artificial Intelligence & Data Science Lab			Course Code:	R25-AI-VSC-216		
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
-	04	-	02	-	-	25	25
Course Description: The Artificial Intelligence & Data Science Lab provides hands-on exposure to fundamental Artificial Intelligence algorithms and Data Science techniques through practical implementation. The course enables students to model and solve real-world problems using search strategies, heuristic methods, adversarial games, constraint satisfaction, and reasoning mechanisms. It also introduces practical data analysis, visualization, and the effective use of modern AI tools for intelligent decision-making.							
Course Relevance: Through laboratory exercises and tool-based assignments, students gain experiential learning that bridges theoretical concepts with real-world applications in domains such as automation, healthcare, education, and data-driven systems.							
Prerequisite: 1. Fundamentals of Programming (C / Python). 2. Fundamental of Data Structure Lab. 3. Probability and Statistics (basics)							
Bridge Content: 1. Introduction to Python for AI and Data Science 2. Revision of linear algebra and probability concepts 3. Basics of data preprocessing and visualization 4. Overview of search techniques and problem formulation							
Course Objectives: 1. To apply hands-on experience in implementing core Artificial Intelligence algorithms. 2. To design AI-based solutions using searching and data-driven approaches and study of CSP. 3. To study various search strategies, along with knowledge representation and reasoning methods. 4. To apply data preprocessing and analysis techniques to extract meaningful insights from datasets. 5. To analyze data using statistical methods and visualization tools to support informed decision-making.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcomes						Bloom's Level
CO1	APPLY fundamentals of AI algorithms such as search techniques and problem-solving methods.						3
CO2	IMPLEMENT adversarial search, constraint satisfaction, and optimization techniques to solve AI problems.						3
CO3	DESIGN and implement intelligent solutions using knowledge representation and reasoning methods.						4
CO4	APPLY data analysis, preprocessing, and visualization techniques using Data Science tools.						3



CO5	APPLY modern AI and Data Science tools responsibly to develop solutions for real-world applications.	3
Guidelines for Lab Conduction		
A suggested list of laboratory assignments is provided under Groups A and B. Each student shall complete a minimum of 10 assignments, including at least 5 assignments from Group A and 5 assignments from Group B, along with one mandatory mini project based on Artificial Intelligence and Data Science as specified in the syllabus. Recommended Operating System: Microsoft Windows 10/11 (64-bit) recommended Programming Tools: Jupyter Notebook, Visual Studio Code, PyCharm, or any other suitable Integrated Development Environment (IDE) appropriate for the course objectives.		
Guidelines for Student's Laboratory Journal		
The laboratory assignments are to be submitted by students in the form of a journal. Journal consists of prologue, Certificate with table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Theory Concept in brief, algorithm, test cases, and conclusion/analysis). Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, Attaching printed papers as part of write-ups and program listing to journals may be avoided. Students' programs maintained on cloud or college servers by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory for accreditation purposes		
Guidelines for Laboratory/Term Work Assessment		
Continuous assessment of laboratory work should be done based on overall performance and Laboratory assignments performance of students. Each Laboratory assignment assessment should be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include timely completion performance, innovation, efficient codes, punctuality and neatness.		
Guidelines for Practical Examination		
Both examiners shall jointly prepare the practical problem statements. The evaluation should primarily emphasize the correct implementation of the solution. Additional questions may be asked to assess the student's conceptual understanding, problem-solving approach, and implementation efficiency. The assessment process should be fair, transparent, and encouraging, fostering student confidence and providing a positive foundation for their academic journey.		
Any 5 Experiments from Group - A		
Group-A (Artificial Intelligence)		
Experiment No. 1	Develop an elementary ChatBot for any suitable customer interaction application.	2 Hrs. CO3

**Objective:**

To design and implement an elementary ChatBot that can interact with users, understand predefined queries, and provide appropriate responses using basic AI concepts such as pattern matching and rule-based logic.

Associated Tasks:

Study the fundamentals of ChatBot and conversational AI, implement a rule-based ChatBot using pattern matching techniques, Test the ChatBot with different user queries, Analyze ChatBot responses and improve accuracy.

Exemplars and Utility:**Exemplars:**

College Admission Inquiry ChatBot, Bank Customer Support ChatBot, Hospital Appointment Booking ChatBot etc.

Utility:

Demonstrates practical implementation of AI concepts such as knowledge representation and reasoning, helps students understand real-world customer interaction systems used in industry.

Experiment No. 2	Implement depth first search (DFS) algorithm and Breadth First Search (BFS) algorithm. Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.	2 Hrs.	CO1
-------------------------	---	---------------	------------

Objective:

To implement Depth First Search (DFS) and Breadth First Search (BFS) algorithms for traversing an undirected graph using recursive and iterative approaches.

Associated Tasks:

Represent an undirected graph using adjacency list or matrix, Implement DFS using a recursive approach, Implement BFS using a queue-based approach.

Exemplars and Utility:**Exemplars:**

Graph traversal in tree and network structures, Path exploration in maze or puzzle problems, Web page or social network link traversal etc.

Utility:

Builds foundation for AI search strategies, Enhances problem-solving and algorithmic thinking skills, Useful in AI, data structures, and real-world pathfinding applications.

Experiment No. 3	Implement A star (A*) Algorithm for any game search problem.	2 Hrs.	CO2
-------------------------	---	---------------	------------

Objective:

To implement the A* search algorithm for solving a game search problem using heuristic-based pathfinding.

Associated Tasks:

Define the problem state space and goal state, Implement the A* algorithm using cost and heuristic values, Apply the algorithm to find an optimal solution path.

Exemplars and Utility:**Exemplars:**

8-puzzle or sliding tile game, Maze pathfinding problem, Game map navigation etc

Utility:

Demonstrates informed search techniques in AI, Improves understanding of heuristic evaluation, Useful in game development, robotics, and navigation systems.

Aeela

R. Ganesh

80



Develops understanding of adversarial search strategies, Improves algorithm efficiency and optimization skills, Useful in game development and decision-making systems.			
Experiment No.7	Assignment and practice of AI tools like Chat GPT, SORA, Gemini, Perplexity, Canva AI and its usages.	2 Hrs.	CO5
Objective: To explore, use, and analyze modern AI tools such as ChatGPT, SORA, Gemini, Perplexity, and Canva AI for real-world problem-solving and productivity enhancement. Associated Tasks: Study features and capabilities of selected AI tools, perform hands-on tasks using each AI tool, Compare outputs, accuracy, and limitations of the tools. Exemplars and Utility: Exemplar: Content generation using ChatGPT, Video generation or simulation using SORA, Information retrieval using Perplexity, Design and presentation using Canva AI, Problem-solving and code assistance using Gemini Utility: Enhances awareness of emerging AI technologies, improves productivity and creativity skills, Prepares students for industry-relevant AI tools and workflows.			
Any 5 Experiments From Group - B			
Group- B (Data Science)			
Experiment No.1	Create an “Academic performance” dataset of students and perform the following operations using Python. 1. Scan all variables for missing values and inconsistencies. If there are missing values and/or inconsistencies, use any of the suitable techniques to deal with them. 2. Scan all numeric variables for outliers. If there are outliers, use any of the suitable techniques to deal with them. 3. Apply data transformations on at least one of the variables. The purpose of this transformation should be one of the following reasons: to change the scale for better understanding of the variable, to convert a non-linear relation into a linear one, or to decrease the skewness and convert the distribution into a normal distribution	2 Hrs.	CO4
Objective: To create an Academic Performance dataset of students and perform data pre-processing techniques such as: 1. Handling missing values and inconsistencies 2. Detecting and treating outliers 3. Applying data transformation for better analysis Associated Tasks: Create a dataset of students with academic details. 1. Scan and handle missing values. 2. Identify and treat inconsistencies. 3. Detects and handles outliers in numeric variables. 4. Apply transformation to improve distribution or scale. Exemplars and Utility: Exemplars: A college wants to analyse student performance. However:			



1. Some students forgot to submit marks → Missing values
2. Some marks were wrongly entered (200, 300) → Inconsistency
3. One student has very low attendance (40%) → Outlier
4. Scores have different scales → Need normalization

Utility: This process is useful in:

1. Educational analytics
2. Academic performance prediction
3. Machine learning model building
4. Student grading systems
5. Institutional decision-making

Experiment No.2	Descriptive Statistics - Measures of Central Tendency and variability Perform the following operations on any open-source dataset (e.g., data.csv) 1. Provide summary statistics (mean, median, minimum, maximum, standard deviation) for a dataset (age, income etc.) with numeric variables grouped by one of the qualitative (categorical) variables. For example, if your categorical variable is age groups and quantitative variable is income, then provide summary statistics of income grouped by the age groups. Create a list that contains a numeric value for each response to the categorical variable. 2. Write a Python program to display some basic statistical details like percentile, mean, standard deviation etc. of the species of 'Iris-setosa', 'Iris-versicolor' and 'Iris-versicolor' of iris.csv dataset.	2 Hrs.	CO4
-----------------	---	--------	-----

Objective: To perform descriptive statistical analysis on an open-source dataset using Python by:

1. Calculating summary statistics (Mean, Median, Min, Max, Standard Deviation)
2. Grouping numeric variables by a categorical variable
3. Computing percentile and statistical measures for different species in the Iris dataset

Associated Tasks: Load an open-source dataset.

1. Group numeric variables based on a categorical variable.
2. Compute summary statistics:
 6. Mean
 7. Median
 8. Minimum
 9. Maximum
 10. Standard Deviation
3. Create a list containing numeric values for each category.
4. Display percentile and statistical measures for different Iris species.

Exemplars and Utility:

Exemplar: A company wants to analyse income distribution by age group.

1. Young people earn lower income.
2. Adults earn the highest income.
3. Seniors have moderate income.
4. Standard deviation shows income variation within groups.



Utility: Descriptive statistics are useful in: 1. Data exploration (EDA) 2. Academic research 3. Machine learning preprocessing 4. Business analytics 5. Biological classification studies 6. Understanding variability & distribution			
Experiment No.3	Data Analytics: -Create a Linear Regression Model using Python/R to predict home prices using Boston Housing Dataset (https://www.kaggle.com/c/boston-housing).The Boston Housing dataset contains information about various houses in Boston through different parameters. There are 506 samples and 14 feature variables in this dataset. The objective is to predict the value of prices of the house using the given features.	2 Hrs.	CO4
Objective: To build a Linear Regression model using Python to predict house prices using the Boston Housing Dataset The dataset contains: 1. 506 samples 2. 14 variables (13 independent features + 1 target variable MEDV) 3. Objective: Predict median house value (MEDV) based on given features like crime rate, number of rooms, tax rate, etc. Associated Tasks: Import required libraries 1. Load the Boston Housing dataset 2. Perform exploratory data analysis (EDA) 3. Split dataset into training and testing sets 4. Build Linear Regression model 5. Predict house prices 6. Evaluate model using: 1. Mean Absolute Error (MAE) 2. Mean Squared Error (MSE) 3. R² Score Exemplars and Utility: Exemplar: A real estate company wants to estimate the price of a house based on: 1. Location crime rate 2. Number of rooms 3. Property tax 4. Pollution level Utility: Linear Regression on housing data is useful in: 1. Real estate price prediction 2. Property investment analysis 3. Bank loan risk assessment 4. Government urban planning 5. Machine learning model development practice			



Experiment No.4	Implement logistic regression using Python/R to perform classification on Social_Network_Ads.csv dataset. 2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset	2 Hrs.	CO4
Objective: To implement a Logistic Regression classification model using Python on the Social_Network_Ads.csv and evaluate its performance using a Confusion Matrix to calculate. Associated Tasks: Import required libraries 1. Load dataset 2. Select features (Age, EstimatedSalary) 3. Split dataset into training and testing sets 4. Apply feature scaling 5. Train Logistic Regression model 6. Predict test results 7. Compute confusion matrix 8. Calculate Accuracy, Error rate, Precision, Recall Exemplars and Utility: Exemplar: A company wants to predict whether a user will buy a product after seeing an advertisement. Utility: Logistic Regression classification is useful in: 1. Digital marketing purchase prediction 2. Credit card fraud detection 3. Medical disease prediction 4. Spam email detection 5. Customer behaviour analysis 6. Binary classification problems			
Experiment No.5	Implement Simple Naïve Bayes classification algorithm using Python/R on iris.csv dataset. 2. Compute Confusion matrix to find TP, FP, TN, FN, Accuracy, Error rate, Precision, Recall on the given dataset.	2 Hrs.	CO4
Objective: To explore patterns in the Titanic dataset using visualization techniques from Seaborn and analyze the distribution of ticket prices (fare) using a histogram. We use the inbuilt dataset: Titanic dataset Associated Tasks: Load Titanic dataset using Seaborn 1. Perform exploratory data analysis (EDA) 2. Visualize survival patterns 3. Plot histogram of ticket fare 4. Interpret patterns from visualizations Exemplars and Utility: Exemplar: From analysis: 1. Women and children were prioritized for rescue. 2. First-class passengers had better access to lifeboats. 3. High ticket fare correlates with higher survival probability. Example Insight: A female passenger in first class had a higher survival chance compared to a male passenger in third			



class. Utility: This analysis is useful for: 1. Exploratory Data Analysis (EDA) 2. Feature selection for machine learning 3. Understanding data distribution 4. Identifying social and economic patterns 5. Learning Seaborn visualization techniques			
Experiment No.6	Use the inbuilt dataset 'titanic'. The dataset contains 891 rows and contains information about the passengers who boarded the unfortunate Titanic ship. Use the Seaborn library to see if we can find any patterns in the data. 2. Write a code to check how the price of the ticket (column name: 'fare') for each passenger is distributed by plotting a histogram.	2 Hrs.	CO5
Objective: To explore patterns in the Titanic dataset using visualization techniques from Seaborn and analyze the distribution of ticket prices (fare) using a histogram. We use the inbuilt dataset: Titanic dataset Associated Tasks: Load Titanic dataset using Seaborn 1. Perform exploratory data analysis (EDA) 2. Visualize survival patterns 3. Plot histogram of ticket fare 4. Interpret patterns from visualizations Exemplars and Utility: Exemplar: From analysis: 1. Women and children were prioritized for rescue. 2. First-class passengers had better access to lifeboats. 3. High ticket fare correlates with higher survival probability. Example Insight: A female passenger in first class had a higher survival chance compared to a male passenger in third class. Utility: This analysis is useful for: 1. Exploratory Data Analysis (EDA) 2. Feature selection for machine learning 3. Understanding data distribution 4. Identifying social and economic patterns 5. Learning Seaborn visualization techniques			
Experiment No.7	Download the Iris flower dataset or any other dataset into a DataFrame. (e.g., https://archive.ics.uci.edu/ml/datasets/Iris). Scan the dataset and give the inference as: 1. List down the features and their types (e.g., numeric, nominal) available in the dataset. 2. Create a histogram for each feature in the dataset to illustrate the feature distributions. 3. Create a boxplot for each feature in the dataset. 4. Compare distributions and identify outliers.	2 Hrs.	CO5



Objective: To download and analyse the Iris dataset

by performing exploratory data analysis (EDA) including:

1. Identifying features and their types
2. Plotting histograms
3. Creating boxplots
4. Comparing distributions
5. Detecting outliers

Associated Tasks:

1. Load dataset into a Data Frame
2. List features and identify their types
3. Create histogram for each feature
4. Create boxplot for each feature
5. Compare distributions and detect outliers

Exemplars and Utility:

Exemplar: Botanical Research

A botanist measures unknown iris flowers:

1. If petal length < 2 cm → Likely *Setosa*
2. If petal length > 5 cm → Likely *Virginica*

Histogram clearly shows petal length separates species.

Utility: This EDA process helps in

Understanding data distribution

Feature selection for ML models

Detecting outliers

Improving model accuracy

Learning visualization techniques

Suggested Course Projects:

- 1) Predict student academic performance based on study hours, attendance, and other factors.
- 2) Predict diseases based on symptoms or medical data.
- 3) Analyse public sentiment from tweets or reviews.

Text Books:

1. Stuart Russell and Peter Norvig, “AI A Modern Approach”, 3 Edition, Pearson, 2003, ISBN :10:0136042597
2. Deepak Khemani, “A First Course in AI”, McGraw Hill Education (India), 2013, ISBN: 978-1-25-902998-1
3. Elaine Rich, Kevin Knight and Nair, “Artificial Intelligence”, TMH, ISBN - 978-0-07-008770-5
4. Chirag Shah, “A Hands-On Introduction to Data Science”, ISBN 978-1-108-47244-9.

Reference Books:

1. Nilsson Nils J, “AI A new Synthesis”, Morgan Publishers Inc. ISBN: 978-1-55-860467-4
2. Patrick Henry Winston, “AI”, Addison-Wesley Publishing Company, ISBN: 0-201- 53377-4
3. Wes McKinney, “Python for Data Analysis”, O' Reilly media, ISBN: 978-1-449-31979-3

Virtual Laboratory:

1. Artificial Intelligence1 <https://ai1-iiith.vlabs.ac.in/List%20of%20experiments.html>
2. Artificial Intelligence2 <https://ai2-iiith.vlabs.ac.in/List%20of%20experiments.html>

MOOCs Links and additional reading material:

Acela

R. Ganesh



1. Artificial Intelligence Search Methods <https://nptel.ac.in/courses/106106226>
2. Artificial Intelligence by Prof. S. Sarkar, Prof. P. Mitra <https://nptel.ac.in/courses/106105078>
3. Artificial Intelligence by Prof. P. Dasgupta. <https://nptel.ac.in/courses/106105079>
4. Python for Data Science https://onlinecourses.nptel.ac.in/noc21_cs33/preview

CO-PO-PSO Mapping

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

1: Slight (Low) 2: Moderate (Medium) 3: Substant

Assessment Pattern

BT Level	CCE based on ARCV									
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9	Exp. 10
CO	CO3	CO1	CO2	CO2	CO3	CO4	CO4	CO4	CO4	CO4
L1										
L2										
L3	√	√	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√	√	√
L5										
L6										

This course serves as a prerequisite for / maps with the following future courses:

This course provides foundational knowledge and practical skills in Artificial Intelligence and Data Science, which serve as a prerequisite for the following advanced courses:

- Machine Learning
- Deep Learning
- Big Data Analytics
- Data Mining and Warehousing

Job Mapping:

This course equips students with practical skills in Artificial Intelligence and Data Science, preparing them for the following job roles:

1. Data Analyst
2. Data Scientist
3. Machine Learning Engineer
4. AI Engineer

Acela

R. Ganesh



Program:	S. Y. B. Tech			Semester:	IV		
Course:	Modern Indian Language and Technical Writing			Course Code:	R25-AI-AEC-217		
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	OR
-	02	-	02	-	-	50	-
Course Description: This course develops communication skills in Marathi along with technical writing and presentation abilities. It includes practical activities like speeches, group discussions, and report writing using LaTeX.							
Course Relevance: The course helps students build confidence in communication, teamwork, and professional skills. It prepares them for academic, workplace, and social interactions effectively.							
Prerequisite: 1. Indian Knowledge System. 2. Professional Communication Skill.							
Bridge Content: The course connects basic communication knowledge with practical applications like presentations, technical writing, and digital tools such as LaTeX.							
Course Objectives: 1. To enhance students’ verbal, written, and interpersonal communication skills through practical activities. 2. To build confidence in public speaking, group discussions, and presentations. 3. To improve technical writing skills including letter writing, report writing, and use of tools like LaTeX. 4. To develop teamwork, leadership, and collaborative skills through group activities and team-building exercises.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome						Bloom’s Level
CO1	UNDERSTAND the importance, history, and cultural value of Marathi language.						2
CO2	ANALYZE and present ideas through book reviews, presentations, and group discussions.						4
CO3	DEMONSTRATE public speaking skills through prepared speech, extempore, storytelling, and oral presentations.						3
CO4	APPLY formal writing techniques to draft letters, applications, and structured reports using tools like LaTeX.						3
CO5	DEVELOP teamwork, leadership, and interpersonal skills through group activities, presentations, and collaborative tasks.						3
Guidelines for Lab Conduction							



1. Internet – Dr. Ashok Bande, Manorama Publication, Mumbai
2. Business Marathi – Dr. L. N. Tambollikar, Phadke Publication, Kolhapur
Reference Books:
1. Applied Marathi – Editors: Dr. Naresh Mehta, Santosh Gore, Sujata Gore
2. Introduction to Information Technology – Digish J. O. Limbaji
MOOCs Links and additional reading material:
1. Latex- A document preparation system https://www.latex-project.org

CO-PO-PSO Mapping

CO	PO1	PO2	PO3	PO4		PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	-	-	-	-		-	-	-	1	2	-	2	-	-	2
CO2	-	-	-	-		-	-	-	1	2	-	2	-	-	2
CO3	-	-	-	-		1	-	2	1	1	-	2	-	-	2
CO4	-	-	-	-		1	-	2	1	1	-	2	-	-	2
CO4	-	-	-	-		2	-	2	1	1	-	2	-	1	2
Avg.	-	-	-	-		1	-	2	1	1	-	2	-	1	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Assessment Pattern

BT Level	CCE based on ARCV									
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9	Exp. 10
CO	CO1	CO1	CO2	CO2	CO3	CO3	CO4	CO4	CO5	CO5
L1										
L2	√		√							
L3	√	√	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√	√	√
L5										
L6										

This course serves as a prerequisite for / maps with the following future courses:

1. Professional Communication and Technical Documentation
2. Project Work, Seminar, and Presentation Skills
3. Entrepreneurship Development and Team-based Projects.

Job Mapping:

Technical Writer, Content Writer, Communication Specialist, Business Analyst, Trainer, Customer Support Executive, and roles requiring presentation and documentation skills across industries.

Aeela

R. Gaurish



Program:	S. Y. B. Tech			Semester:		III	
Course:	Technology Commercialization and Startup Development			Course Code:		R25-AI-EEM-218	
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	OR
02	-	-	02	50	-		—
Course Description: This course introduces students to the process of converting innovative ideas and technologies into commercially viable products and startups, covering key aspects such as intellectual property rights, startup registration, market research, funding strategies, and go-to-market planning.							
Course Relevance: This course is highly relevant as it equips students with practical knowledge and skills required to transform technological innovations into successful business ventures, fostering entrepreneurship, innovation, and industry readiness in a competitive startup ecosystem.							
Prerequisite: 1. Entrepreneurship Development							
Bridge Content: real-world problem identification, along with introductory knowledge of market research, digital tools, and startup ecosystem basics.							
Course Objectives: 1. To understand the importance of technology commercialization and startup development. 2. To understand intellectual property rights for protecting inventions and product ownership. 3. To acquire knowledge of the registration process for startups. 4. To understand cost estimation and funding setup for startups. 5. To develop Go-to-Market (GTM) strategies for business ventures.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome					Bloom's Level	
CO1	APPLY the concepts of the technology commercialization for starting a new venture.					2	
CO2	REGISTER invention to protect the invention through IPR.					3	
CO3	DISCUSS the registration process with requisite market knowledge, skills and right attitude.					3	
CO4	EXPLORE the costing sheet by identifying the available funding resources.					4	
CO5	IMPLEMENT go-to-Market strategy for a business venture.					4	
Course Content							
Unit I	Introduction to Technology commercialization & Startup ecosystem				06 Hrs		CO1
Introduction, Need and importance of commercialization, role of startup in technology and commercialization, challenges in technology commercialization, support systems for startups, future trends in technology, Design Thinking.							
Exemplars / Practical Applications and usage: Electric car, Pharma company, Joint venture, Agriculture.							



Unit 2	IPR & Legal compliance	6 Hrs.	CO2
Invention and innovation: need, benefits, intellectual property rights protection, patent drafting, patent, copyright, trademark design, design registration procedure of IPR filing, legal policies, IT act, GST and income tax, companies act, labor law, environmental protection act.			
Exemplars / practical applications and usage: Google search algorithm, Pepsi ingredient			
Unit 3	Registration process & Market research	6 Hrs.	CO3
Registration process: Steps to register startup with startup India, benefits of startup registration, requisites documents & information, AI-assisted product development, startup eligibility criteria. Market research: Development of marketing plan, pricing concepts and pricing strategy, consumer behavior, market intelligence, marketing communication and promotional strategies.			
Exemplars / practical applications and usage: Proprietary firm- Patnaji Ayurveda, Private Limited Company-TCS, Partnership- Khaitan & Co.			
Unit 4	Costing & Funding strategy	6 Hrs.	CO4
One time cost: Need, financial components, business formation and registration, professional services, advertising, infrastructure, technology, recurring cost: rent, salaries, insurance, tax, loan, maintenance, travel and training, types of startup funding, stages of startups and source of funding, steps to startup fund raising, types of investors, investors look for in startups , investors mindset to invest in startups , startup India funding support, startup India investor connect, credit guarantee scheme for startups.			
Exemplars / Practical Applications and usage: Rapido, Blinkit, OYO, Unacademy			
Unit 5	Growth and scaling -Go to market strategy	6 Hrs.	CO5
Growth and scaling: significance, difference, scaling key metrics, identifying target segments and personas, analyzing customer needs and competitive landscape, value propositions, unique selling points (USPs), choosing distribution channels, pricing strategies, marketing, positioning plans, ans off matrix, scaling frameworks, organic vs. inorganic growth strategies, leveraging technology, partnerships for scalability, key Performance indicators for GTM, Feedback loops and agile adaptation			
Exemplars / Practical Applications and usage: Zomato's Expansion Strategy in Tier 2 and 3 Cities, Analyze how Zomato tailored its GTM strategy to penetrate smaller markets, adjusted pricing, and adapted to local preferences			
Textbooks:			
1. Fundamentals of Information Technology Author: Shambhavi Roy, Clinton Daniel, and Manish Agrawal. 2. 8 Steps To Innovation: Going From Jugaad To Excellence, Collins India, 2013. ISBN: 9789350293584 3. National Student and Faculty Startup Policy 2019. Government of India. 4. Pavan Soni, "Design Your Thinking - The Mindsets, Toolsets and Skill Sets For Creative Problem Solving", Penguin Random House India Pvt. Ltd. 2020, ISBN: 9780670094097. 5. Intellectual Property, A primer for academia, Prof. Rupinder Tiwari, Mamta Bharadwaj, Publication, Bureau Panjab University Chandigarh. https://dst.gov.in/sites/default/files/E-			
Reference Books:			



1. Design Thinking for New Business Contexts: Theory and Practice *Yujia Huang & David Hands* — Palgrave Macmillan, 2022
2. Information Technology Author: V.Rajaraman
3. Innovation and Entrepreneurship, Peter F. Drucker, Harper Business; Reprint, 2006, ISBN:9780060851132.
4. The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically
5. Successful Businesses, Crown Currency; Illustrated, 2011. ISBN: 9780307887894.
6. Innovator's DNA, Updated, with a New Preface: Mastering the Five Skills of Disruptive Innovators,
7. Harvard Business Review Press; Revised, 2019. ISBN: 1633697207.

MOOCs Links :

1. Intellectual property rights-laws and practices <https://dst.gov.in/sites/default/files/E-BOOK%20IPR.pdf>,
2. Innovation and Start-up Policy https://onlinecourses.swayam2.ac.in/imb20_mg22/preview
3. Innovation, Business Models and Entrepreneurship https://onlinecourses.nptel.ac.in/noc19_mg55/preview
4. Intellectual Property https://onlinecourses.nptel.ac.in/noc22_hs59/preview
5. Innovation Driven Entrepreneurship https://onlinecourses.swayam2.ac.in/ntr24_ed05/preview

YouTube/Video Links:

1. Introduction to Information Technology, <https://www.youtube.com/watch?v=7BfdMKelTj0>,
2. Intellectual Property Rights (IPR), https://www.youtube.com/watch?v=zKwJAVg6_ME
3. Harvard i-lab, <https://www.youtube.com/watch?v=rqi-n0hA4uo>
4. Digital Marketing Specialist, <https://www.youtube.com/watch?v=F4YuptMRMBY>
5. Growth for Startups, <https://www.youtube.com/watch?v=6lY9CYIY4pQ>
6. Startup Roadmap <https://www.youtube.com/watch?v=zwQ8TNkcYzc>

CO-PO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	3	1	2	2	1	2	2	3	2	1	2	3
CO2	1	2	2	—	1	2	3	1	2	1	2	—	1	3
CO3	1	2	2	1	1	2	2	2	3	2	2	—	2	3
CO4	2	3	3	2	2	3	1	2	2	3	2	1	2	3
CO5	2	3	3	2	3	3	2	3	3	3	2	2	3	3
Avg.	2	3	3	2	2	2	2	2	2	2	2	1	2	3

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

**Unit wise Weightage:**

Exam	Unit1	Unit2	Unit3	Unit4	Unit5
CCE					
CT1	10	10			
CCE2			10	10	
CCE3					10
ESE					
Total	10	10	10	10	10

Assessment Table:

BT Level	CCE			ESE				
	CCE1	CCE2	CCE3					
CO				CO1	CO2	CO3	CO4	CO5
L1								
L2	√	√						
L3	√	√	√					
L4	√	√	√					
L5								
L6								

This course serves as a prerequisite for / maps with the following future courses:

1. Digital Marketing
2. Business Analytics
3. Project Management
4. Entrepreneurship Development

Job Mapping:

Job opportunities that one can get after learning this course: entrepreneur/startup founder, innovation manager, product developer, business development executive, marketing executive, startup consultant, Business Analyst and project manager in technology and business organizations.



Program:		S. Y. B. Tech			Semester:		IV	
Course:		Database management Systems Lab			Course Code:		R25-AI-PCC-219	
Teaching Scheme				Examination Scheme				
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR	
-	02	-	01	-	-	25	25	
Course Description: This laboratory course provides hands-on experience in database design, SQL programming, PL/SQL and database administration concepts. Students will design databases using ER modeling, implement relational schemas, write and optimize SQL queries, and perform transaction management using real-world datasets. The course emphasizes practical exposure to data integrity, normalization, concurrency control and NoSQL basics, bridging the theoretical DBMS concepts with industry-relevant database applications. Course Relevance: The DBMS Lab develops practical data modeling, query formulation and database handling skills essential for modern software, data, and AI-driven applications. It prepares learners for entry-level roles such as database developer, data analyst and backend engineer, and provides a strong foundation for advanced studies in data science, big data analytics, cloud databases and intelligent data processing systems.								
Prerequisite: 1. Fundamentals of data structures 2. Familiarity with SQL environment setup on Windows/Linux/Mac platforms								
Bridge Content: Fundamental database concepts, relational models, basic SQL commands								
Course Objectives: The course aims to: 1. To understand basics of database manipulation skills and implement a database schema for a given problem domain. 2. To design and implement PL/SQL programs. 3. To explain designing of database, creating relational database and analysis of table design. 4. To study the concepts and techniques relating to MongoDB and its implementations. 5. To learn to design database connectivity for implementation of a project.								
Course Outcomes: After learning the course, students will be able to								
CO	Course Outcome						Bloom's Level	
CO1	DEMONSTRATE the use of SQL DDL, DML, and TCL commands for creating, manipulating, and controlling relational databases.						3	
CO2	EXECUTE SQL queries using joins, views, sequences, and indexes to retrieve data efficiently and improve query performance.						3	
CO3	DESIGN and implement PL/SQL programs, including stored procedures, stored functions, cursors, and packages to solve database-oriented problems.						4	
CO4	EXECUTE MongoDB (NoSQL) queries to retrieve, manipulate, and manage unstructured and semi-structured data.						3	



CO5	IMPLEMENT database-driven solutions by adopting emerging database technologies and real-world application requirements, considering performance and societal impact.	4
Guidelines for Instructor's Manual		
The instructor's manual/Lab Manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword), curriculum of course, conduction and Assessment guidelines, topics under consideration concept, objectives, outcomes, set of typical applications/assignments/guidelines, references.		
Guidelines for Student's Laboratory Journal		
The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and handwritten write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software and Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory Concept in brief, algorithm, flowchart, test cases, Test Data Set(if applicable)), conclusion. Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Students programs maintained on cloud or college server by Laboratory In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory for accreditation purpose.		
Guidelines for Laboratory/Term Work Assessment		
Continuous assessment of laboratory work should be done based on overall performance and Laboratory assignments performance of student. Each Laboratory assignment assessment should be assigned grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each Laboratory assignment assessment include timely completion performance, innovation, efficient codes, punctuality and neatness.		
Guidelines for Laboratory Conduction		
The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy needs to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute them among batches of students. It is appreciated if the assignments are based on real world problems/applications. Use of open source software is to be encouraged. In addition to these, instructors may assign one real life application in the form of a mini-project based on the concepts learned. Instructors may also set one assignment or mini-project that is suitable to respective branch beyond the scope of the syllabus.		
Guidelines for Practical Examination		



Both examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

Experiment No. 1	SQL Queries: (DDL) - To understand and apply SQL Data Definition Language (DDL) commands for designing and managing database structures such as base tables and views.	2 Hrs.	CO1
-------------------------	---	---------------	------------

Objective: To learn how to create and manage database objects using SQL DDL commands.

Associated Tasks:

1. Identify the requirements of a suitable database application.
2. Design the database schema including base tables and views.
3. Create database objects using the CREATE command.
4. Modify database structures using ALTER command.
5. Remove database objects using DROP command.
6. Change object names using the RENAME command.
7. Remove all records from tables using TRUNCATE command.
8. Execute and verify at least 10 SQL DDL queries.

Exemplars and Utility:

Exemplars:

A railway department wants to store train and passenger information for ticket booking.

- i. Create a table Trains with attributes:
Train_ID, Train_Name, Source, Destination, Total_Seats.
- ii. Create a table Passengers with attributes:
Passenger_ID, Name, Age, Gender, Phone.
- iii. Create a table Bookings with attributes:
Booking_ID, Passenger_ID, Train_ID, Seat_Number, Travel_Date.
- iv. Define Passenger_ID and Train_ID as Foreign Keys.

Utility:

1. Enables effective design and maintenance of relational database structures for real-world applications.
2. Builds a strong foundation for advanced database programming and administration tasks.

Experiment No. 2	SQL Queries: (DML) - To understand and apply SQL Data Manipulation Language (DML) commands for retrieving, inserting, updating, and deleting data from base tables and views.	2 Hrs.	CO1
-------------------------	--	---------------	------------



Objective: To learn how to insert, retrieve, update, and delete data using SQL DML commands.

Associated Tasks:

1. Study the structure of base tables and views in a suitable database application.
2. Insert data into tables using the INSERT command.
3. Retrieve required data using the SELECT command.
4. Modify existing records using the UPDATE command.
5. Remove records using the DELETE command.
6. Execute and verify at least 10 SQL DML queries demonstrating different operations.

Exemplars and Utility:

Exemplars: A railway reservation system maintains information about trains, passengers, and ticket bookings. The database administrator needs to manipulate the data for booking, updating, retrieving, and cancelling tickets.

- i. Trains
(Train_ID, Train_Name, Source, Destination, Total_Seats)
- ii. Passengers
(Passenger_ID, Passenger_Name, Age, Gender, Phone)
- iii. Bookings
(Booking_ID, Passenger_ID, Train_ID, Seat_Number, Travel_Date)

Utility:

1. Develops practical skills in handling and manipulating data stored in relational databases.
2. Forms the basis for real-world database application development and data-driven decision making.

Experiment No. 3

Index, Sequence and View - To apply SQL commands for creating and using sequences, indexes, and views, and to retrieve meaningful information from an employee database.

2 Hrs.

CO2

Objective: To learn how to use sequences, indexes, and views for efficient database management.

Associated Tasks:

1. Create a sequence to automatically generate unique employee numbers for the *empno* column.
2. Create an index on the *county* attribute to improve query performance.
3. Execute a query using *zipcode = 071* and verify whether the, recording the observation.
4. Create a view for employees having a salary less than 50,000 and residing in *Mumbai*.
5. Retrieve the count of employees residing in *Mumbai*.
6. Calculate the average salary of employees from the created view.
7. Display employee names residing on the same street based on the created view.

Exemplars and Utility:

Exemplars:

A hospital management system maintains records of doctors, patients, and appointment bookings. The system uses sequences to automatically generate appointment IDs, indexes to speed up patient searches, and views to display useful information for hospital staff and patients. Create a sequence to generate booking IDs.

- i. Create a sequence to generate appointment IDs.
- ii. Insert an appointment record using the sequence.
- iii. Create an index to improve the speed of patient searches.



iv. Create a view combining patient and doctor information.

Utility: Enhances understanding of database performance optimization and data abstraction using indexes and views.

Experiment No. 4	SQL Joins - To understand and apply INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN in SQL for retrieving related data from multiple tables.	2 Hrs.	CO2
-------------------------	--	---------------	------------

Objective: To learn how to retrieve related data from multiple tables using SQL join operations.

Associated Tasks :

1. Design and study the structure of Customers and Orders tables.
2. Retrieve customer orders using INNER JOIN to display only customers who have placed orders.
3. Retrieve all customers using LEFT JOIN, including those who have not placed any orders.
4. Retrieve all orders using RIGHT JOIN, including orders placed by customers not present in the Customers table.
5. Retrieve a complete list of customers and orders using FULL OUTER JOIN, including unmatched records from both tables.

Exemplars and Utility:

Exemplars:

A library database stores information in multiple tables:

- Members (Member_ID, Member_Name, Phone)
- Books (Book_ID, Book_Title, Author, Publisher)
- Borrowings (Borrowing_ID, Member_ID, Book_ID, Issue_Date, Return_Date)

Based on this database, write SQL queries to perform the following operations:

- i. Retrieve member names and book titles for all confirmed borrowings.
- ii. Retrieve member names and book titles (if borrowed).
- iii. Retrieve book titles with member names (if available).
- iv. Retrieve member and book details including unmatched records.

Utility:

1. Strengthens understanding of relational data retrieval across multiple tables.
2. Provides practical knowledge applicable to real-world database-driven applications such as order management systems.

Experiment No. 5	PL/SQL blocks - To design and execute PL/SQL blocks for inserting borrower details and calculating book return fines using control structures in a library management database.	2 Hrs.	CO3
-------------------------	--	---------------	------------

Objective: To learn how to write and execute PL/SQL blocks for database programming.

Associated Tasks:

1. Create the Borrower and Fine tables with appropriate attributes.
2. Write a PL/SQL block to accept user input and insert records into the Borrower table.
3. Calculate the number of days between Date of Issue and current date.
4. Apply conditional logic (IF-ELSE) to compute fine based on the given rules.
5. Update the Status of the book from *Issued (I)* to *Returned (R)* after submission.
6. Insert fine details into the Fine table when applicable.

Exemplars and Utility:

Exemplars: When a guest cancels a hotel room reservation, the system calculates a cancellation charge



based on the time remaining before the scheduled check-in date.

Cancellation Rules (Example)

- More than 48 hours before check-in → ₹50 charge
- Between 24 and 48 hours before check-in → ₹100 charge
- Less than 24 hours before check-in → ₹200 charge

Write a PL/SQL block to calculate the cancellation charge when a guest cancels a hotel room reservation based on the time remaining before the check-in date.

Utility: Provides practical experience in PL/SQL programming for real-world applications such as library management systems, enhancing skills in data validation, conditional processing, and transaction handling.

Experiment No.6	Triggers and Cursors - To implement PL/SQL triggers and cursors for audit logging and efficient data processing in an Employee Management System.	2 Hrs.	CO3
------------------------	--	---------------	------------

Objective: To learn how to implement triggers and cursors for automating database operations.

Associated Tasks:

1. Design and create Employee and Employee_Audit tables.
2. Develop a trigger on the Employee table to log INSERT, UPDATE, and DELETE operations into the Employee_Audit table.
3. Verify trigger execution by performing data modification operations.
4. Declare and use a cursor to iterate through employee records.
5. Process salary increments using the cursor based on defined conditions (e.g., 10% increment for IT department employees).

Exemplars and Utility:

Exemplars: A university database maintains records of courses, students, and enrollments. Create a trigger to increase the available seats of a course when a student drops it (deleted). Also, write a PL/SQL cursor to display enrollment details with student names and course titles.

Utility: Provides practical experience in maintaining data integrity and developing advanced PL/SQL skills applicable to real-world enterprise database applications.

Experiment No.7	NoSQL Database Introduction – MongoDB Aggregation and Indexing - To work with MongoDB for NoSQL data storage and apply aggregation and indexing techniques for efficient data analysis and performance optimization.	2 Hrs.	CO4
------------------------	---	---------------	------------

Objective: To learn how to perform data analysis and improve query performance using MongoDB aggregation and indexing.

Associated Tasks:

1. Create a MongoDB database and collection for user data.
2. Perform CRUD operations (Create, Read, Update, Delete) on the collection.
3. Write aggregation queries to compute statistics such as totals, counts, and averages.
4. Create indexes on suitable fields and observe query performance improvements.
5. Analyze the impact of indexing on data retrieval efficiency.

Exemplars and Utility:

Exemplar: An online shopping system uses MongoDB to store order details. Write aggregation queries to calculate total sales per category and average spending per customer. Create appropriate indexes to improve query performance and analyze the results using explain().

100
Aeela

R. Ganesh



Utility: Builds practical skills in handling NoSQL databases, enabling efficient data analysis and performance tuning for modern, large-scale applications.			
Experiment No. 8	Database Connectivity - To implement database connectivity between a front-end application and a MySQL/Oracle database using a programming language such as Python, Java, or PHP, and perform database navigation operations.	2 Hrs.	CO5
Objective: To learn how to connect a frontend application with a database and perform CRUD operations. Associated Tasks: 1. Design a simple database schema in MySQL/Oracle. 2. Establish database connectivity using appropriate drivers or libraries (e.g., JDBC, Python DB-API, PHP MySQLi). 3. Develop a menu-driven program for database operations. 4. Implement database navigation operations such as add, delete, update, and view records. Test the application and verify successful interaction between the front end and the database. Exemplars and Utility: Exemplar: Development of a menu-driven application that performs CRUD operations through database connectivity using a front-end programming language. Utility: Provides hands-on experience in integrating applications with databases, preparing students for real-world software development involving backend data management.			
Experiment No. 9	Unstructured Data Processing with MongoDB: A Social Media Case Study - To process and analyze unstructured social media data using a NoSQL database (MongoDB) for efficient storage and retrieval.	2 Hrs.	CO5
Objective: To learn how to store, process, and analyze unstructured data using MongoDB. Associated Tasks: 1. Import a dataset containing social media posts into MongoDB. 2. Design an appropriate document structure for storing unstructured data. 3. Write MongoDB queries to filter, search, and analyze social media content. 4. Identify top trending hashtags using aggregation and sorting operations. 5. Interpret the query results for data-driven insights. Exemplars and Utility: Exemplars: Unstructured Data Processing with MongoDB: An E-Commerce Product Review Case Study Utility: Enhances skills in handling large-scale unstructured data and applying NoSQL techniques for real-world data analytics and social media applications.			
Experiment No. 10	Mini Project (Group of 2 to 3 students can develop mini project) - To design and develop a database-driven mini project by integrating relational databases, SQL, PL/SQL, NoSQL, and frontend-backend connectivity for solving real-world problems.	2 Hrs.	CO5



Objective: To apply database concepts by developing a complete database-driven application.

Associated Tasks:

1. Identify a real-world problem and define project requirements.
2. Design the relational database schema and/or NoSQL data model.
3. Implement SQL queries, PL/SQL modules, and transaction management.
4. Integrate the database with a frontend and backend using an appropriate programming language.
5. Test, demonstrate, and document the working mini project.

Mini Project Topic

1. Smart City Data Management System
2. E-Learning Analytics Platform
3. Fake News Detection Data Platform
4. Smart Agriculture Data System
5. Disaster Management & Alert System
6. Student Course Registration System
7. Library Management System
8. Employee Management System
9. Hospital Appointment & Patient Record System
10. Online Shopping / Order Management System
11. Attendance Management System
12. Banking Management System
13. Placement Management System
14. Placement Management System
15. Social Media Data Analytics System (NoSQL)
16. Complaint / Grievance Management System

Text Books:

1. Elmasri, R. and Navathe, S. B., Fundamentals of Database Systems, 7th Edition, Pearson Education, 2016.
2. Silberschatz, A., Korth, H. F., and Sudarshan, S., Database System Concepts, 6th Edition, McGraw-Hill Education, 2019.
3. Ramakrishnan, R. and Gehrke, J., Database Management Systems, 3rd Edition, McGraw-Hill Education, 2014.
4. Connolly T., Begg C., "Database Systems", 4th Edition, Pearson Education, 2002, ISBN 8178088614.
5. "MongoDB: The Definitive Guide" by Kristina Chodorow, O Reilly Publications 4.
6. Pramod J. Sadalage Martin Fowler, "NoSQL Distilled", Addison Wesley, ISBN- 10:0321826620

Reference Books:

1. Date, C. J., An Introduction to Database Systems, 8th Edition, Pearson Education.
2. Connolly, T. and Begg, C., Database Systems: A Practical Approach to Design, Implementation and Management, 6th Edition, Pearson Education.
3. Sadalage, P. J. and Fowler, M., NoSQL Distilled, Pearson Education.
4. C J Date, "An Introduction to Database Systems", Addison-Wesley, ISBN: 0201144719 2.
5. S.K.Singh, "Database Systems: Concepts, Design and Application", Pearson Education, ISBN 978-81-317-6092-5
6. Kristina Chodorow, Michael Dierolf, "MongoDB: The Definitive Guide", O Reilly Publications, ISBN: 978-1-449-34468-9 4

**MOOCs Links and additional reading material:**

1. What is NoSQL? NoSQL Databases Explained
<https://www.mongodb.com/resources/basics/databases/nosql-explained%20>
2. Data Modeling in Azure Cosmos DB for NoSQL
<https://learn.microsoft.com/en-us/azure/cosmos-db/nosql/modeling-data>
3. Data Base Management System (NPTEL, IIT Kharagpur)
https://onlinecourses.nptel.ac.in/noc22_cs91/preview
4. Database Management System (DBMS) – NPTEL Video
<http://www.nptelvideos.com/lecture.php?id=6518>
5. Database System Concepts (Sixth Edition)
<https://www.db-book.com/db6/index.html>

CO-PO-PSO Mapping

CO	P01	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	2	1	3	-	-	-	-	-	1	3	2	1
CO2	3	3	2	1	3	-	-	-	-	-	1	3	3	1
CO3	3	2	3	1	3	-	-	1	-	-	1	3	3	1
CO4	2	2	2	1	3	-	-	1	-	-	2	3	2	1
CO5	2	2	3	2	3	1	1	-	-	-	3	3	2	2
Avg.	3	2	3	1	3	1	1	1	-	-	2	3	3	1

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Assessment Pattern

BT Level	CCE based on ARCV									
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9	Exp. 10
CO	CO1	CO1	CO2	CO2	CO3	CO3	CO4	CO4	CO5	CO5
L1										
L2	√	√	√	√	√	√	√	√	√	√
L3	√	√	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√	√	√
L5										
L6										

This course serves as a prerequisite for / maps with the following future courses:

1. Big Data Analytics
2. Data Warehousing and Data Mining
3. Machine Learning
4. Cloud Computing
5. Data Science and Analytics

Job Mapping:

Job opportunities that one can get after learning this course: Database Developer, Database Developer, Database Administrator (DBA), Data Analyst, Data Analyst.



Program:	S. Y. B. Tech			Semester:		IV	
Course:	Embedded System & Internet of Things Lab			Course Code:		R25-AI-MDM-220	
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
-	2	-	01	-	-	25	-
Course Description: This lab provides hands-on experience in designing and implementing embedded systems integrated with IoT technologies. Students work with microcontrollers, sensors, and communication protocols to develop real-time, data-driven applications relevant to AI & Data Science.							
Course Relevance: This course equips students with practical skills to build smart, connected systems essential for AI & Data Science applications. It bridges hardware, software, and data analytics, preparing learners for careers in IoT, automation, smart devices, and intelligent systems development.							
Prerequisite: Computer Fundamentals, Basic Digital Logic							
Bridge Content: The bridge content is designed to help students connect their previous knowledge with the concepts required for Embedded Systems and Internet of Things (IoT). It provides a brief introduction to fundamental topics that support understanding of the laboratory experiments.							
Course Objectives: 1. To understand the fundamentals of Embedded Systems and Internet of Things (IoT) architecture and components. 2. To study various embedded hardware platforms, sensors, and actuators used in IoT systems. 3. To develop the ability to design and implement small embedded IoT applications using microcontrollers or single board computers. 4. To learn communication protocols and networking concepts used in IoT systems. 5. To implement real-time IoT applications based on sensor data collection and device control.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome					Bloom's Level	
CO1	UNDERSTAND the architecture and working of embedded systems and IoT devices .					2	
CO2	IDENTIFY and interface sensors, actuators, and embedded platforms for IoT applications.					2	
CO3	DEVELOP simple IoT systems using embedded programming and hardware platforms.					3	
CO4	APPLY communication protocols for data transmission between IoT devices and cloud systems.					4	
CO5	DESIGN IoT-based applications for real-world problems.					5	
Guidelines for Lab Conduction							



A set of 10 practical assignments is prescribed for the laboratory course. Students must perform all 10 practical assignments distributed as follows Any 3 practicals using Arduino, Any 3 practicals using NodeMCU (ESP8266/ESP32), Any 4 practicals using Raspberry Pi

- Operating System recommended: - Microsoft Windows 10/11 (64-bit)
- Programming tools recommended: - Arduino IDE, Raspberry Pi OS, NodeMCU

Guidelines for Student's Laboratory Journal

The laboratory journal is an essential academic record that documents the experiments performed, observations made, and knowledge gained during the laboratory course. It helps students develop technical documentation skills, analytical thinking, and systematic problem-solving abilities. The journal should include the experiment title, objectives, theory, hardware and software requirements, circuit diagram, algorithm, program code, observations, results, and conclusion for each practical. Students are encouraged to maintain digital copies of source code and outputs through cloud storage or the institutional server to promote Green Computing practices and reduce paper usage.

Guidelines for Laboratory/Term Work Assessment

Continuous assessment of laboratory work should be carried out based on the student's overall performance throughout the semester and the successful completion of laboratory experiments in Embedded Systems and IoT (ESIoT). Each experiment should be evaluated using suitable assessment parameters with appropriate weightage. The suggested parameters include timely completion of experiments, correctness of hardware interfacing and circuit implementation, program execution, innovation in design, troubleshooting ability, understanding of embedded and IoT concepts, documentation quality, punctuality, laboratory discipline, and performance during viva-voce

Guidelines for Practical Examination

The practical examination for Embedded Systems and IoT (ESIoT) should be conducted jointly by the internal and external examiners. The examiners should prepare the practical problem statements together, ensuring appropriate coverage of the course outcomes and laboratory experiments. Assessment should primarily focus on the correct implementation of hardware interfacing, embedded programming, IoT communication, circuit design, debugging, and demonstration of the developed application. Supplementary viva-voce questions may be asked to evaluate the student's conceptual understanding, problem-solving ability, and knowledge of embedded systems and IoT technologies. The evaluation process should be fair, transparent, and objective, encouraging students to demonstrate their practical skills, innovation, and confidence.

Experiment No. 1	Flash the operating System onto the device into a stable functional state by porting desktop environment with necessary packages	2Hrs.	CO1
-------------------------	---	--------------	------------



Objective: To install and configure an operating system on an embedded device like Raspberry Pi. To set up a stable desktop environment with necessary software packages. To gain practical experience in system setup and configuration.

Associated Tasks:

1. Download the required OS image (e.g., Raspberry Pi OS / Linux OS).
2. Use flashing tools (like Balena Etcher / Raspberry Pi Imager) to write the OS onto an SD card.
3. Insert SD card into device and boot the system.
4. Perform initial configuration (language, network, user credentials).
5. Install desktop environment (if not pre-installed).
6. Install necessary packages (browser, editors, utilities).
7. Update and upgrade the system packages.
8. Verify system stability and functionality.

Exemplars & Utility:

Successfully booting the device, installing a desktop environment, and running applications demonstrates practical OS setup and system functionality. This experiment builds hands-on skills in OS installation, system configuration, and package management. It is useful for IoT, embedded systems, and developing real-world projects with strong troubleshooting ability.

Experiment No. 2	Programs to perform arithmetic, logical, branching, and loop operations by a microcontroller.	2Hrs.	CO1
-------------------------	--	--------------	------------

Objective: To understand basic programming of a microcontroller. To perform arithmetic, logical, branching, and looping operations. To develop simple embedded programs for decision-making and control.

Associated Tasks:

1. Write programs for arithmetic operations (addition, subtraction, multiplication, division).
2. Implement logical operations (AND, OR, NOT, XOR).
3. Develop branching programs using conditions (if, if-else, switch).
4. Implement loop operations (for loop, while loop).
5. Compile and upload the program to the microcontroller (e.g., Arduino).
6. Test outputs using LEDs/serial monitor.
7. Debug and verify correct execution.

Exemplars and Utility:

Execution of arithmetic, logical, and control programs on a microcontroller demonstrates core programming concepts.

Helps in developing real-time embedded applications involving decision-making and automation. Builds a strong foundation for IoT, robotics, and embedded system design while improving problem-solving and debugging skills.

Experiment No. 3	Create a program that illuminates the green LED if the counter is less than 100, illuminates the yellow LED if the counter is between 101 and 200 and illuminates the red LED if the counter is greater than 200	2Hrs.	CO2
-------------------------	---	--------------	------------

Objective: To implement conditional programming logic using microcontrollers and LEDs.

Associated Tasks:

1. Write a program to increment a counter variable.
2. Use conditional statements to control LEDs.
3. Connect green, yellow, and red LEDs to the microcontroller.
4. Observe LED behavior based on counter values



Exemplars: Green LED glows when counter < 100, yellow LED when counter is between 101–200, and red LED when counter > 200. Demonstrates how logical conditions are used in embedded systems and IoT applications			
Experiment No. 4	Write a program to show the temperature and shows a graph of the recent measurements	2Hrs.	CO3
Objective: To visualize sensor data using graphical representation. Associated Tasks: 1. Read temperature values from the sensor. 2. Send the data to the computer using serial communication. 3. Use software tools to generate graphs of temperature readings. 4. Analyze the trend of temperature changes Exemplars and Utility: Exemplars & Utility A graph showing temperature variations over time is displayed on the computer screen. Helps students understand data visualization and analysis in IoT systems. Send temperature/humidity data to Thing Speak or Blynk cloud dashboard.			
Experiment No. 5	Interfacing DHT11 sensor with Arduino/Node MCU/Raspberry Pi.	2Hrs.	CO4
Objective: To interface the DHT11 sensor with a microcontroller and measure temperature and humidity. To display sensor data and gain practical knowledge of real-time data acquisition in embedded systems. Associated Tasks: 1. Study the working of the DHT11 sensor. 2. Connect the DHT11 sensor to Arduino / NodeMCU / Raspberry Pi (VCC, GND, Data pin). 3. Install required libraries (e.g., DHT library). 4. Write and upload code to read sensor data. 5. Display temperature and humidity values on Serial Monitor / LCD. 6. Verify output readings and troubleshoot errors. 7. Modify program for continuous data monitoring. Exemplars & Utility: Successful interfacing and real-time display of temperature and humidity demonstrate sensor integration and data acquisition. Useful for applications like weather monitoring, smart homes, and IoT-based environmental systems. Enhances skills in embedded programming, sensor handling, and real-time system development. (Google Colab / TinyML / Sensor Data Analytics) Export sensor data CSV file. Analyze graphs in Google Colab. Perform basic data analytics like average, max, min temperature.			
Experiment No. 6	Interface the Servo Motor rotates 180 degrees Using Arduino /Node MCU/Raspberry Pi.	2Hrs.	CO3
Objective: To interface a servo motor with a microcontroller and control its rotation. To achieve precise 0° to 180° movement using programming. To understand real-time control of actuators in embedded systems. Associated Tasks: 1. Study the working principle of a servo motor. 2. Connect servo motor to Arduino / Node MCU / Raspberry Pi (VCC, GND, Signal pin). 3. Use PWM (Pulse Width Modulation) for controlling servo position. 4. Write and upload program to control angle (0° to 180°).			



5. Test servo movement at different angles.
6. Verify smooth rotation and troubleshoot issues.
7. Modify code for continuous or step-wise movement.

Exemplars and Utility:

Controlling a servo motor to rotate between 0° and 180° demonstrates precise actuator control using PWM signals.

This is useful in robotics, automation, and IoT applications like robotic arms and smart systems. It enhances skills in embedded programming, hardware interfacing, and real-time control systems.

Experiment No. 7	Write a program to display "Hello world" on display of the receiving microcontroller using RS232.	2Hrs.	CO3
-------------------------	--	--------------	------------

Objective: To understand serial communication using RS232 protocol between microcontrollers. To transmit and receive data and display “Hello World” on the receiving device. To gain practical knowledge of communication interfaces in embedded systems.

Associated Tasks:

1. Study the RS232 communication protocol.
2. Connect two microcontrollers using RS232 interface (via MAX232 if required).
3. Configure baud rate and serial communication parameters.
4. Write a transmitter code to send “Hello World”.
5. Write receiver code to read and display the message.
6. Upload programs to both devices.
7. Test communication and verify output on display/serial monitor.
8. Troubleshoot communication errors (baud mismatch, wiring issues).

Exemplars and Utility:

Successful transmission and display of “Hello World” demonstrates reliable serial communication between devices.

This experiment is useful in embedded systems, industrial communication, and IoT applications. It builds a strong foundation Modern Communication Protocols (I2C, SPI, MQTT) in interfacing, data transfer, and communication protocols. Modern Communication Protocols (I2C, SPI, MQTT)

Experiment No.8	Write an application to detect obstacle and notify user using LEDs	2Hrs.	CO4
------------------------	---	--------------	------------

Objective: To design an application for detecting obstacles using a sensor (e.g., ultrasonic/IR). To notify the user through LEDs based on obstacle presence. To understand real-time sensing and indication in embedded systems.

Associated Tasks:

1. Study the working of obstacle detection sensors (Ultrasonic/IR).
2. Interface the sensor with Arduino / Node MCU / Raspberry Pi.
3. Connect LEDs for indication (e.g., red for obstacle, green for clear path).
4. Write a program to read sensor data.
5. Implement logic to detect obstacles based on distance/value.
6. Control LEDs based on detection condition.
7. Test system in real-time environment.
8. Debug and verify proper working.

Exemplars and Utility: Detecting obstacles and indicating through LEDs demonstrates real-time sensing and alert systems. This is useful in robotics, smart vehicles, and automation systems. It enhances skills in sensor interfacing, programming, and embedded system design.



Experiment No.9	Using the light sensor, monitor the light intensity and automatically turn ON/OFF LED	2Hrs.	CO4
Objective: To monitor ambient light intensity using a light sensor (LDR). To automatically control an LED based on light conditions. To understand real-time sensing and automation in embedded systems. Associated Tasks: 1. Study the working of light sensors (LDR). 2. Interface LDR with Arduino / Node MCU / Raspberry Pi using resistor. 3. Connect LED as output indicator. 4. Write a program to read analog light intensity values. 5. Set threshold value for light detection. 6. Control LED (ON/OFF) based on intensity. 7. Test system under different lighting conditions. 8. Debug and verify performance. Exemplars and Utility: Automatic LED control based on light intensity demonstrates sensor-based automation. This is useful in street lights, smart homes, and energy-saving systems. It enhances skills in sensor interfacing, embedded programming, and real-time control. Use Node MCU Wi-Fi module for wireless monitoring.			
Experiment No. 10	Write a program using piezo element and use it to play a tune after someone knocks	2Hrs.	CO3
Objective: To generate sound signals using a piezo element connected to a microcontroller Associated Tasks: 1. Connect a piezo buzzer to Arduino output pins. 2. Write a program to generate sound tones. 3. Create a simple tune sequence. 4. Trigger the sound when a sensor input is detected. Exemplars and Utility: Exemplar: The buzzer plays a tune when someone knocks or presses a button Utility: Demonstrates sound generation and alert systems used in security or notification devices.			
Mini Project Topic			
1. Smart Home Environmental Monitoring System 2. IoT-Based Weather Monitoring Station 3. Smart Obstacle Detection and Alert System 4. Automatic Street Light Control System 5. Smart Servo-Based Automatic Door System 6. IoT-Based Security Alarm System 7. Smart Traffic Signal Control System 8. Embedded Device Health Monitoring Dashboard			
Text Books:			
1. Arshdeep Bahga and Vijay Madisetti, “Internet of Things: A Hands-On Approach”. 2. Massimo Banzi “Getting Started with Arduino”,			
Reference Books:			



1. Zacker, C., “PC Hardware: The Complete Reference”, McGraw-Hill/Osborne. ISBN: 978-0072129405
2. Singh, R., Gehlot, A., & Bhargava, S. C., “Internet of Things with Arduino and Raspberry Pi”, CRC Press, 2020. ISBN: 978-0367418275

MOOCs Links and additional reading material:

1. Internet of Things-https://onlinecourses.nptel.ac.in/noc22_cs53/preview
2. Arduino-<https://projecthub.arduino.cc/>
3. Computer Networks: https://onlinecourses.nptel.ac.in/noc22_cs19/preview
4. Google Colab-<https://colab.research.google.com/notebooks/intro.ipynb>
5. ChatGPT-<https://help.openai.com/en/collections/3742473-chatgpt>
6. Teachable Machine-<https://teachablemachine.withgoogle.com>

CO-PO-PSO Mapping

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

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Assessment Pattern

BT Level	CCE based on ARCV									
	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp. 6	Exp. 7	Exp. 8	Exp. 9	Exp. 10
CO	CO1	CO1	CO2	CO2	CO3	CO3	CO4	CO4	CO5	CO5
L1										
L2										
L3	√	√	√	√	√	√	√	√	√	√
L4	√	√	√	√	√	√	√	√	√	√
L5										
L6										

This course serves as a prerequisite for / maps with the following future courses:

1. Robotics and Automation
2. Machine Learning

Job Mapping:

Job opportunities that one can get after learning this course: Embedded Systems Engineer, IoT Developer / Engineer, Robotics Engineer, IoT Product Developer, Hardware Engineer, Automation Engineer



Program:	S. Y. B. Tech			Semester:	IV		
Course:	Sustainable Development			Course Code:	R25-AI-VEC-221		
Teaching Scheme				Examination Scheme			
Lecture	Practical	Tutorial	Credit	CCE	ESE	TW	PR
02	-	-	02	50	-	-	-
Course Description: This course explores the balance between the environment, economy, and society through the study of ecosystems, biodiversity, and resource conservation. Students will evaluate global challenges and the role of innovation like renewable energy, green computing, and waste management in achieving the Sustainable Development Goals (SDGs).							
Course Relevance: In a world facing climate change, sustainability expertise is now a core professional requirement. This course prepares students to implement eco-friendly technologies and waste-reduction strategies in urban and industrial settings. By mastering resource lifecycles and biodiversity conservation, students gain the skills to drive "Green Innovation" and ensure organizations meet global environmental standards.							
Prerequisite: 1. Wellness And Lifestyle Education 2. Universal Human Values							
Bridge Content: Basic knowledge of the environmental footprint of digital communication and how online privacy rights intersect with the social equity principles of sustainable development.							
Course Objectives: 1. To introduce the multidisciplinary nature and scope of environmental studies and sustainable development. 2. To understand ecosystem structures, biodiversity, and ecological balance through observation and case studies. 3. To examine the use and impact of natural resources and environmental pollution on sustainability 4. To develop e-waste management, environmental regulations, and the role of technology in sustainable development.							
Course Outcomes: After learning the course, students will be able to							
CO	Course Outcome					Bloom's Level	
CO1	EXPLAIN the relationship between environment, ecosystems, and sustainable development.					2	
CO2	USE knowledge of ecosystems and biodiversity conservation strategies.					3	
CO3	ANALYZE environmental issues such as pollution and natural resource depletion.					4	
CO4	ANALYZE sustainable technologies and waste management practices for environmental protection.					4	
Course Content							
Unit 1	Fundamental Sustainable Development					4 Hrs.	CO1



Concept, meaning, scope, and definitions of sustainable development – Principle of Sustainable Development – The pillars of sustainable development – Approaches to Sustainable Development: Status Quo Approach, Community Capacity Building Approach, Industrial Sector Approach, Integrated Systems Approach, Human Development Approach, and Green Account Approach.			
Exemplars / Practical Applications and usage: Analysis of Green Data Center cooling and renewable energy use compared to Campus Solar Plants.			
Unit 2	Sustainable Development Goals	4 Hrs.	CO2
Nature of Sustainable Development Goals – 2030 Global Agenda for Sustainable Development. Government Policies and their implications for sustainable development in India, Contribution of International Organizations and NGOs, Government Initiatives for Sustainable Development.			
Exemplars / Practical Applications and usage: A study of how United Nations Sustainable Development Goals (SDGs) are implemented through initiatives such as Smart Cities Mission, Swachh Bharat Abhiyan etc.			
Unit 3	Environmental Sustainability	3 Hrs.	CO3
Introduction and components of ecosystem, Structure and functioning of ecosystem. Types of ecosystems: Forest ecosystem, Grassland ecosystem, Desert ecosystem, Aquatic ecosystem. Food chain and food web, Biodiversity and its importance, Biodiversity conservation strategies. Natural resources: renewable and non-renewable, Resource depletion and conservation. Environmental pollution, Environmental protection measures and policies.			
Exemplars / Practical Applications and usage: Using "Nature-Based Solutions," such as restoring upstream forests, to naturally filter a city's water supply instead of building expensive chemical treatment plants.			
Unit 4	Sustainable Practice and Technologies in AI	3 Hrs.	CO4
Introduction to sustainable technologies and the role of Artificial Intelligence (AI) in environmental sustainability. Renewable energy sources including solar, wind, biomass, and hydropower, E-waste management, recycling, and resource recovery technologies, sustainable IT practices, and energy-efficient systems, Smart infrastructure and AI-enabled solutions for sustainable development, environmental monitoring, and resource optimization.			
Exemplars / Practical Applications and usage: A case study on "Urban Mining," where gold, silver, and copper are extracted from discarded electronics to reduce the need for destructive traditional mining.			
Textbooks:			
1. Dr. C. Jeyaprabha & Dr. V. Sribharathy, "Text Book on Environmental Science and Sustainability", Evincepub Publishing, 2024, ISBN: 978-9-35-673980-2. 2. Baldev Raj Nayar, Globalization and Nationalism: The Changing Balance of India's Economic Policy, 1950–2000 (New Delhi: Sage, 2001)			
Reference Books:			



1. Agrawal, A N (1995). Indian Economy: Problems of development and planning. pune: Vishwa Prakashan.
2. Nandhini Kanmani, "Sustainable Development: Theory and Practice", PHI Learning, 2014, ISBN: 978-8-12-034938-4.
3. Bidyut Mohanty (1993) Urbanization in Developing Countries Basic Services and community Participation, Institute of Social Science, Concept Publishing House.

MOOCs Links and additional reading material:

1. The Age of Sustainable Development coursera.org/learn/sustainable-development

CO-PO-PSO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	-	-	-	-	-	2	2	2	1	-	3	-	2	2
CO2	-	-	-	-	-	2	2	2	2	-	3	-	3	2
CO3	-	-	-	-	-	2	2	2	2	-	3	-	3	2
CO4	1	1	1	1	1	2	3	2	2	-	3	1	2	2
Avg.	1	1	1	1	1	2	2.5	2	2	-	3	1	3	2

1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

Unit wise Weightage:

Exam	Unit1	Unit2	Unit3	Unit4
CCE				
CCE1	12	13		
CCE2			12	13
CCE3				
ESE				
Total	12	13	12	13

Assessment Pattern

BT Level	CCE			ESE				
	CT1	CT2		CO1	CO2	CO3	CO4	CO5
CO								
L1	√	√						
L2	√	√						
L3	√	√						
L4	√	√						
L5								
L6								

This course serves as a prerequisite for / maps with the following future courses:

1. Environmental Law and Corporate Governance.
2. Renewable Energy Engineering.
3. Green Computing and Data Centers.

Job Mapping:

Job opportunities that one can get after learning this course: Sustainability Consultant roles, Environmental Green Audit professional training.

Aeela

R. Ganesh