

DEPARTMENT OF

**COMPUTER
SCIENCE AND
ENGINEERING
(AI & ML, IOT),
R & AI**

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY HYDERABAD

B.TECH. II YEAR

ROBOTICS AND ARTIFICIAL INTELLIGENCE

III SEMESTER

R25

Course Code	Title of the Course	L	T	P/D	CH	C
25BS1MT204	Probability and Statistical Analysis	2	1	0	3	3
25PC1RA201	Object Oriented Programming Through C++	3	0	0	3	3
25PC1RA202	Sensors and Actuators for Robotics	3	0	0	3	3
25PC1RA205	Fundamentals of Robotics and Artificial Intelligence	3	0	0	3	3
25HS1MG201	Innovation and Entrepreneurship	2	0	0	2	2
25BS2MT211	Computational Mathematics Laboratory	0	0	2	2	1
25PC2RA201	Object Oriented Programming Through C++ Laboratory	0	0	2	2	1
25PC2RA202	Sensors and Actuators Laboratory	0	0	2	2	1
25PC2RA211	Software Systems for Robotics and AI Laboratory	0	0	2	2	1
25ES2CY101	Python Programming Laboratory	0	0	2	2	1
25SD5CS201	Data Visualization Laboratory	0	0	2	2	1
25MN6HS102	Environmental Science	2	0	0	2	0
Total		15	1	12	28	20

IV SEMESTER

R25

Course Code	Title of the Course	L	T	P/D	CH	C
25PC1RA203	Robot Operating System	2	0	0	2	2
25PC1AM201	Machine Learning	3	0	0	3	3
25PC1RA204	Kinematics and Dynamics of Robots	3	0	0	3	3
25PC1EC202	Signals and Systems	3	0	0	3	3
25PC1CS205	Design and Analysis of Algorithms	3	0	0	3	3
25PW4RA201	Design Thinking	1	0	2	3	2
25PC2RA203	Robot Operating System Laboratory	0	0	2	2	1
25PC2AM201	Machine Learning Laboratory	0	0	2	2	1
25PC2RA204	Kinematics and Dynamics of Robots Laboratory	0	0	2	2	1
25SD5CS202	Web Application Development	0	0	2	2	1
25VA7EN201	Gender Sensitization	1	0	0	1	1
Total		16	0	10	26	21

L – Lecture T – Tutorial P – Practical D – Drawing
 C – Credits SE – Sessional Examination CA – Class Assessment
 SEE – Semester End Examination D-D – Day to Day Evaluation
 CP – Course Project PE – Practical Examination

CH – Contact Hours/Week
 ELA – Experiential Learning Assessment
 LR – Lab Record

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. III Semester

(25BS1MT204) PROBABILITY AND STATISTICAL ANALYSIS

TEACHING SCHEME		
L	T/P	C
2	1	3

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE PRE-REQUISITES: First Year Mathematics courses

COURSE OBJECTIVES:

- To learn the theory of Random Variable, and probability distributions of single random variables
- To understand the concept of sampling distribution
- To know the various methods to test the hypothesis for large and small samples
- To find the linear and polynomial relationship between the given data set
- To understand the concepts of Multivariable linear Regression model

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Apply the concepts of Random variable and distributions to some case studies

CO-2: Evaluate statistical parameters of continuous probability distributions and sampling distributions

CO-3: Apply the knowledge to test the hypotheses for large and small samples

CO-4: Use and fit a linear and polynomial regression model to the data and use it for prediction

CO-5: Model the linear relationship between the explanatory (independent) Variables and response(dependent)variables

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	1	2	1	-	-	-	-	-	-	2	3	2
CO-2	3	2	1	2	1	-	-	-	-	-	-	2	3	2
CO-3	3	2	1	2	1	-	-	-	-	-	-	2	3	3
CO-4	3	3	3	2	2	-	-	-	-	-	-	2	2	3
CO-5	3	3	2	2	2	-	-	-	-	-	-	2	3	3

UNIT-I:

Random Variables and Probability Distributions: Concept of a Random Variable – Discrete Probability Distributions – Continuous Probability Distributions – Mean of a Random Variable – Variance of a Random Variable

Discrete Probability Distributions: Binomial Distribution – Poisson distribution

UNIT-II:

Continuous Distributions and sampling: Uniform Continuous Distribution – Normal Distribution – Areas under the Normal Curve – Applications of the Normal Distribution

Sampling Distributions: Population, sample, parameter, statistic, Types of Sampling – Sampling Distributions with and without replacement – Sampling Distribution of Means – Central Limit Theorem.

Introduction – Statistical Inference – Classical Methods of Estimation – Single Sample: Estimating the mean – Standard error of a point estimate - Estimating a proportion.

UNIT-III:

Tests of Hypothesis:

Statistical Hypothesis: General Concepts (Types of hypothesis, Type-I & II errors, Critical region, One-tailed and two-tailed) – Testing a Statistical Hypothesis-small (t-test) and large samples. Single sample: Tests concerning a single mean. Two samples: Tests on two mean. One sample: Test on a single proportion. Two samples: Tests on two proportions. F-test for equality of variances..

UNIT-IV:

Applied Statistics: Correlation and Regression: Curve fitting by the method of least squares, fitting of straight lines, second degree parabolas and more general curves, Correlation coefficient, Rank correlation

UNIT-V:

Multivariate Analysis: Multivariate Analysis: Multivariate Normal Distribution Function, Properties of Multivariate Normal Distribution, Conditional Distribution and its relation to regression model, Linear Regression Analysis-Step up, Step down (Forward/Backward), Subset and Best Model (Step-wise), Tenfold Validation, One -Off technique, (Leave-Out), Multi Collinearity, Variation Inflation Factors, Principal Component Analysis (PCA) based regression analysis.

TEXT BOOKS:

1. Probability & Statistics for Engineers & Scientists, Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, 9th Edition, Pearson Publishers
2. Fundamentals of Mathematical Statistics, S. C. Gupta and V. K. Kapoor, Khanna publications
3. Higher Engineering Mathematics, B. S. Grewal, Khanna Publishers, 36th Edition, 2010

REFERENCES:

1. Fundamentals of Probability and Statistics for Engineers, T. T. Soong, John Wiley & Sons Ltd., 2004
2. Probability and Statistics for Engineers and Scientists, Sheldon M. Ross, Academic Press

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. III Semester

(25PC1RA201) OBJECT ORIENTED PROGRAMMING THROUGH C++

TEACHING SCHEME		
L	T/P	C
3	0	3

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE PRE-REQUISITES: Programming for Problem Solving

COURSE OBJECTIVES:

- To understand C++ fundamentals, control structures, functions, pointers, and memory management
- To apply object-oriented concepts using classes, objects, constructors, and data abstraction
- To implement inheritance and polymorphism using virtual functions and abstract classes
- To perform file handling and formatted input/output operations in C++
- To develop reliable programs using exception handling mechanisms and practical use of multithreading in C++

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Understand fundamental C++ programming constructs to develop structured programs

CO-2: Explain object-oriented concepts and differentiate OOP from procedural programming

CO-3: Analyze inheritance models and implement polymorphism using virtual functions

CO-4: Use C++ stream and file I/O features to perform formatted and unformatted data operations

CO-5: Apply exception-handling and multithreading mechanisms to develop thread-safe C++ programs that use concurrency mechanisms

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	1	-	2	-	-	-	-	-	1	3	2	-
CO-2	3	2	3	-	2	-	-	-	-	-	1	3	3	-
CO-3	2	3	3	2	2	-	-	-	-	-	1	2	3	-
CO-4	2	2	2	2	3	-	-	-	-	-	1	2	2	-
CO-5	2	3	2	2	2	-	-	-	-	-	2	2	2	-

UNIT-I:

C++ Basics: Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References. Flow control statement- if, switch, while, for, do-while, break, continue, goto statements. Functions – Defining the Function, Scope of variables, Parameter passing, Default arguments, inline functions, Recursive functions, Pointers to functions.

Dynamic memory allocation and de-allocation operators - new and delete, Preprocessor directives.

UNIT-II:

Object-Oriented Thinking: Different paradigms for problem solving, need for OOP paradigm, differences between OOP and Procedure oriented programming, Overview of OOP concepts- Abstraction, Encapsulation, Inheritance and Polymorphism.

C++ Classes and Data Abstraction: Class definition, Class structure, Objects creation, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Dynamic creation and destruction of objects, Data abstraction, ADT and information hiding.

UNIT-III:

Inheritance: Defining the Derived class, Different forms of inheritance, Access Modifiers, Access the base class members, Inheritance with constructors, Destructors, Virtual base class.

Virtual Functions and Polymorphism: Static and Dynamic binding, virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes, Implications of polymorphic use of classes, Virtual destructors.

UNIT-IV:

C++ I/O: I/O using C++ functions, Stream classes hierarchy, Stream I/O, File streams and String streams, Overloading operators, Error handling during file operations, Formatted I/O.

UNIT-V:

Exception Handling: Benefits of exception handling, Standard Exception hierarchy, Exception handling mechanism, Exception objects, Exception specifications, Stack unwinding, rethrowing an exception, catching all exceptions, Creating custom exceptions.

Multithreading: Basics of concurrency in robotics, Process vs Thread, C++ thread library, Creating and managing threads, Shared data and synchronization, Race conditions, Condition variables for sensor-control coordination, Producer-Consumer pattern for sensor pipelines, Atomic operations, Thread safety – Deadlocks and prevention.

TEXT BOOKS:

1. The Complete Reference C++, Herbert Schildt, 4th Edition, Tata McGraw-Hill
2. Problem solving with C++: The Object of Programming, Walter Savitch, 4th Edition, Pearson Education
3. C++ Concurrency in Action, Anthony Williams, 2nd Edition, Manning Publications

REFERENCES:

1. The C++ Programming Language, B. Stroutstrup, 3rd Edition, Pearson Education
2. OOP in C++, T. Gaddis, J. Walters and G. Muganda, 3rd Edition, Wiley Dream Tech Press
3. Object Oriented Programming in C++, R. Lafore, 3rd Edition, Galigotia Publications

ONLINE RESOURCES:

1. <https://www.w3schools.com/cpp>
2. <https://www.geeksforgeeks.org/cpp>

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. III Semester

(25PC1RA202) SENSORS AND ACTUATORS FOR ROBOTICS

TEACHING SCHEME		
L	T/P	C
3	0	3

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE PRE-REQUISITES: Basic Electrical and Electronics Engineering

COURSE OBJECTIVES:

- To understand the physics and signal conditioning of internal state (proprioceptive) sensors
- To learn the technical parameters of robotic vision systems, including shutter types and lens characteristics
- To analyze exteroceptive sensors like LiDAR and Force-Torque sensors for environmental interaction
- To evaluate the performance of electric actuators and their associated power electronic drives
- To design high-precision robotic transmission systems using specialized geared mechanisms

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Analyze performance metrics of IMUs and Encoders and Implement error compensation techniques

CO-2: Configure camera parameters (CSI/USB, shutter type, FOV) for specific robotic perception tasks

CO-3: Interpret 3D point cloud data and tactile feedback for robot navigation and grasping

CO-4: Select appropriate electric motors (BLDC/Stepper/Servo) based on torque-speed requirements

CO-5: Design a low-backlash robotic joint using harmonic or cycloidal transmission systems

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	3	1	1	3	1	-	-	-	-	-	-	3
CO-2	3	2	3	1	2	3	1	-	-	-	-	-	-	3
CO-3	3	2	3	3	2	3	1	-	-	-	-	-	-	3
CO-4	3	2	3	1	1	3	1	-	-	-	-	-	-	3
CO-5	3	3	3	3	3	3	1	-	2	-	-	-	-	3

UNIT-I:

Proprioceptive Sensors (Internal State)

Introduction to Metrology: Sensitivity, Resolution, Linearity, Hysteresis, and Bandwidth.

Position and Velocity: Optical Encoders (Incremental vs. Absolute, Quadrature Encoding, Gray Code), Hall Effect Sensors. Inertial Sensing: MEMS Accelerometers,

Gyroscopes (Coriolis Effect), Magnetometers, 6-DOF and 9-DOF Inertial Measurement Units (IMU). Sensor Noise: Bias, Drift, and Noise Modeling.

UNIT-II:

Robotic Vision and Imaging Systems

Camera Hardware: CMOS vs. CCD architectures. Interfaces: CSI (Camera Serial Interface) for embedded systems (Jetson/Pi) vs. USB Webcams. Shutter Mechanisms: Rolling Shutter (artifacts and Jello effect) vs. Global Shutter (high-speed synchronization).

Lens Parameters: Focal length, Aperture (f -number), Field of View (FOV), Radial and Tangential Distortion. Depth Cameras: Stereo Vision (disparity), Structured Light, and Time-of-Flight (ToF) principles.

UNIT-III:

Exteroceptive Perception and Force Sensing

Active Ranging: Ultrasonic Sensors, Infrared (IR) Triangulation. LiDAR: 2D vs. 3D LiDAR, Spinning vs. Solid-State LiDAR, Point Cloud generation.

Force and Tactile: Strain Gauges, Wheatstone Bridge, Load Cells, Multi-axis Force-Torque (F/T) sensors. Tactile arrays for robotic skin and grasping.

UNIT-IV:

Electric Actuators and Drives

Motor Physics: DC Motors (Back EMF, Torque-Speed characteristics).

Brushless DC (BLDC) Motors: Electronic commutation and Hall sensor feedback.

Stepper Motors: Unipolar/Bipolar, Micro-stepping.

Servo Motors: PWM control and PID feedback loops. Drive Electronics: H-Bridge architecture, MOSFET/IGBT selection, and Gate Drivers.

UNIT-V:

Geared Mechanisms and Transmission

Principles of Mechanical Advantage: Gear Ratios, Torque Multiplication, and Efficiency.

Gear Geometry: Spur, Helical, Bevel, and Worm gears. Precision Robotics Gearboxes: Planetary Gears (Stages and planet carrier), Harmonic Drives (Wave generator, Flexspline, Circular spline mechanics), Cycloidal Drives (Zero/Low backlash concepts).

Linear Transmission: Lead screws, Ball screws, and Timing belts.

TEXTBOOKS:

1. Sensors and Actuators: Engineering System Instrumentation, Clarence W. de Silva, 2nd Edition, CRC Press
2. Computer Vision: Algorithms and Applications, Richard Szeliski, 2nd Edition, Springer
3. Introduction to Autonomous Mobile Robots, Roland Siegwart, Illah Nourbakhsh, and Davide Scaramuzza, 2nd Edition, MIT Press

REFERENCES:

1. Introduction to Robotics: Mechanics and Control, John J. Craig, 4th Edition, Pearson
2. Introduction to Robotics: Analysis, Control, Applications, Saeed B. Niku, 3rd Edition, Wiley

ONLINE RESOURCES:

1. https://onlinecourses.nptel.ac.in/noc25_ee76/preview
2. https://onlinecourses.nptel.ac.in/noc24_me55/preview

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. III Semester

(25PC1RA205) FUNDAMENTALS OF ROBOTICS AND ARTIFICIAL INTELLIGENCE

TEACHING SCHEME		
L	T/P	C
3	0	3

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE OBJECTIVES:

- To define the scope, history, and classifications of robotic systems to build a foundational understanding
- To identify the essential electro mechanical components and subsystems required to build a robot
- To understand the fundamental concepts of Artificial Intelligence and the design of Intelligent Agents
- To differentiate between various Machine Learning paradigms and their specific applications in automation
- To illustrate how AI algorithms are integrated with control systems for autonomous task execution

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Summarize the history, laws, and mechanical classifications of robots

CO-2: Deconstruct a robot into its subsystems (sensors, actuators, control) to understand how it is built

CO-3: Apply the concept of Rational Agents to model simple robotic behaviors

CO-4: Distinguish between various ML paradigms (Supervised, Unsupervised, RL) and their specific applications

CO-5: Assess the role of AI in moving from simple automation to autonomous robotic task execution

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	1	1	2	-	-	-	-	-	-	3	2	2
CO-2	3	3	2	2	2	-	-	-	-	-	-	3	3	2
CO-3	3	3	2	3	2	-	-	-	-	-	-	3	3	3
CO-4	3	3	3	2	3	2	1	1	2	2	1	3	3	3
CO-5	2	2	1	1	2	3	3	3	2	2	2	2	2	3

UNIT-I:

Introduction to Robotics: Definitions of Robotics, Automation vs. Autonomy, History of Robotics, Asimov's Three Laws, Classification by Geometry (Cartesian, Cylindrical, Spherical, Articulated, SCARA), Classification by Mobility (Mobile, Wheeled, Legged, Aerial, Fixed Manipulators), Current trends (Cobots, Service Robots, Industrial Robots).

UNIT-II:

The Anatomy of a Robot: The Robot Subsystems (Sense-Think-Act paradigm), Mechanical Structure, Links, Joints (Revolute vs. Prismatic), Degrees of Freedom (DOF), Power Sources (Hydraulic, Pneumatic, Electric drives), End Effectors (Mechanical

Grippers, Vacuum, Magnetic, Tools), Workspace Analysis (Reach and Envelope qualitative understanding).

UNIT-III:

Foundations of Artificial Intelligence: Definitions of AI (Acting Humanly vs. Acting Rationally), The Turing Test, Chinese Room Argument, Intelligent Agents, Agent & Environment, PEAS Description (Performance, Environment, Actuators, Sensors), Environment Types (Fully Observable vs. Partial, Deterministic vs. Stochastic), Types of Agents (Simple Reflex, Model-Based, Goal-Based, Utility-Based).

UNIT-IV:

Introduction to Machine Learning: The AI Hierarchy (Relationship between AI, ML, and DL), Types of Machine Learning, Supervised Learning (Classification, Regression), Unsupervised Learning (Clustering), Reinforcement Learning (Reward/Penalty concepts), Introduction to Neural Networks, The Perceptron model, Weights, Biases, Activation Functions.

UNIT-V:

Robot Control and Task Execution: Control Basics, Open-Loop vs. Closed-Loop Control Systems, Feedback concepts, Task Planning, Motion Planning basics, Obstacle Avoidance (Bug Algorithms, Potential Fields), AI in Control (Computer Vision for Perception, Logic for Decision Making), Case Study (Architecture of an Autonomous Vacuum Cleaner).

TEXTBOOKS:

1. Introduction to Robotics: Analysis, Control, Applications, Saeed B. Niku, 3rd Edition, Wiley
2. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, 4th Edition, Pearson

REFERENCES:

1. Foundations of Robotics: A Multidisciplinary Approach with Python and ROS, Damith Herath and David St-Onge, Springer
2. Modern Robotics: Mechanics, Planning, and Control, Kevin M. Lynch and Frank C. Park, Cambridge University Press

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. III Semester

(25HS1MG201) INNOVATION AND ENTREPRENEURSHIP

TEACHING SCHEME		
L	T/P	C
2	0	2

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE OBJECTIVES:

- To familiarize on the basic concepts of innovation, entrepreneurship and its importance
- To identify and analyze the process of problem-opportunity identification, market segmentation, and idea generation techniques
- To initiate prototype development and understand minimum viable product
- To develop initial Business and financial planning and Go-to-Market strategies
- To impart knowledge on establishing startups, venture pitching and IPR

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Understand the entrepreneurship and the entrepreneurial process and its significance in economic development

CO-2: Assess the problem from an industry perspective and generate solutions using the design thinking principles

CO-3: Assess market competition, estimate market size, and develop a prototype

CO-4: Analyze Business and financial planning models and Go-to-Market strategies

CO-5: Able to build a start-up, register IP and identify funding opportunities

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	-	-	-	1	-	-	1	1		-	-	1
CO-2	2	3	2	1	-	1	-	-	2	1		-	1	-
CO-3	1	2	3	2	-	1	-	-	3	2	1	-	1	-
CO-4	1	1	3	3	-	1	-	-	2	3	1	-	1	-
CO-5	1	1	-	-	2	2	-	-	2	2	1	-	-	-

UNIT-I:

Fundamentals of Innovation and Entrepreneurship

Innovation: Introduction, need for innovation, Features, Types of innovations, innovations in manufacturing and service sectors, fostering a culture of innovation, planning for innovation.

Entrepreneurship: Introduction, types of entrepreneurship attributes, mindset of entrepreneurial and intrapreneurial leadership, Role of entrepreneurs in economic development. Women Entrepreneurship, Importance of on-Campus Startups. Understanding to build an entrepreneurial mindset, attributes, and networks in individuals while on campus.

Core Teaching Tool: Simulation, Game, Industry Case Studies (Personalized for students – 16 industries to choose from), Venture Activity.

UNIT-II:

Problem and Customer Identification: Identification of gap, problem, analyzing the problem from an industry perspective, real-world problems, market and customer segmentation, validation of customer problem fit, Iterating problem-customer fit, Competition and Industry trends mapping, and assessing initial opportunity, Porter's Five Force Model. Idea generation, Ideation techniques: Brainstorming, Brain writing, Round robin, and SCAMPER, Design thinking principles, Mapping of solution to problem.

Core Teaching Tool: Several types of activities, including: Class, game, Gen AI, 'Get out of the Building', and Venture Activity.

UNIT-III:

Opportunity Assessment and Prototype Development: Identify and map global competitors, review industry trends, and understand market sizing: TAM, SAM, and SOM. Assessing scope and potential scale for the opportunity. Understanding prototyping and Minimum Viable Product (MVP). Developing a prototype: Testing and validation.

Core Teaching Tool: Venture Activity, no-code Innovation tools, Class activity

UNIT-IV:

Business & Financial Models: Introduction to Business Model and types, Lean Canvas Approach: 9-block lean canvas model, building a lean canvas for your startup. Business planning: components of Business plan- Sales plan, People plan, and financial plan. Financial Planning: Types of costs, preparing a financial plan for profitability using a financial template, understanding the basics of Unit economics, Economies of Scale and analyzing financial performance. Go-To-Market (GTM) approach – Selecting the Right Channel, creating digital presence, and building a customer acquisition strategy.

Core Teaching Tool: Founder Case Studies – Sama and Securely Share; Class activity and discussions; Venture Activities.

UNIT-V:

Startups and IPR: Startup requirements, building founding team members and mentors, pitch preparation, start-up registration process, funding opportunities and schemes, institutional support to entrepreneurs, startup lifecycle, documentation, legal aspects in startup, venture pitching readiness, National Innovation Startup Policy (NISP) and its features.

Patents, Designs, Patentability, Procedure for grants of patents. Indian Scenario of Patenting, International Scenario: International cooperation on Intellectual Property. Patent Rights: Scope of Patent Rights. Copyright, trademark, and GI. Licensing and transfer of technology.

Core Teaching Tool: Expert talks; Cases; Class activity and discussions; Venture Activities.

TEXT BOOKS:

1. Innovation and Entrepreneurship, John R Bessant, Joe Tidd, 4th Edition, Wiley, Latest Edition
2. The Startup Launch Book - A Practical Guide for Launching Customer Centric Ventures, Ajay Batra, Wiley, 2020
3. Entrepreneurship Development and Small Business Enterprises, Poornima M Charantimath, 3rd Edition, Pearson, 2018

REFERENCES:

1. Entrepreneurship Simplified: From Idea to IPO, Ashok Soota and S. R. Gopalan, Penguin Portfolio, 2021
2. How to Be a Founder: How Entrepreneurs Can Identify, Fund and Launch Their Best Ideas, Alice Bentinck and Matt Clifford, Bloomsbury Business, 2024
3. Scaling Up: How a Few Companies Make It... and Why the Rest Don't, Verne Harnish, Sanage Publishing, 2024
4. Funding Your Startup and Other Nightmares, Dhruv Nath and Sushanto Mitra, Penguin Portfolio, 2020
5. NISP -Brochure inside pages - startup_policy_2019.pdf

B. Tech. III Semester

(25BS2MT211) COMPUTATIONAL MATHEMATICS LABORATORY

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE PRE-REQUISITES: Matrices, Iterative Methods and Ordinary Differential Equations

COURSE OBJECTIVES:

- To solve problems of Eigen values and Eigen Vectors using MATLAB
- To solution of algebraic and transcendental equations using MATLAB
- To solve problems of linear system of equations
- To solve the first-order ODEs and higher order linear differential equations with constant coefficients

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Develop the code to find the Eigen values and Eigen Vectors using MATLAB

CO-2: Develop the code find solution of algebraic and transcendental equations and linear system of equations using MATLAB

CO-3: Write the code to solve problems of first-order ODEs higher order linear differential equations with constant coefficients

CO-4: Fit the curve and visualize all solutions graphically through programmes

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	3	2	2	3	-	-	-	2	-	-	3	2	2
CO-2	3	3	2	2	3	-	-	-	2	-	-	3	2	2
CO-3	3	3	2	2	3	-	-	-	2	-	-	3	2	2
CO-4	3	3	2	2	3	-	-	-	2	-	-	3	3	2

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

WEEK 1-3:

Eigen Values and Eigen Vectors:

Programs:

- Finding real and complex Eigen values
- Finding Eigen vectors
- SVD and PCA

WEEK 4-5:

Linear System of Equations: Jacobi's iteration method and Gauss-Seidel iteration method

Programs:

- Solution of given system of linear equations using Jacobi's method
- Solution of given system of linear equations using Gauss-Seidel method

WEEK 6-7:

Solution of Algebraic and Transcendental Equations: Bisection method, Newton Raphson Method

Programs:

- Root of a given equation using Bisection method.
- Root of a given equation Newton Raphson Method.

WEEK 8-11:**First and Higher Order ODEs with Constant Coefficients**

Exact and non-exact equations, Applications: exponential growth/decay, Newton's law of cooling.

Programs:

- Solving exact and non-exact equations
- Solving exponential growth/decay and Newton's law of cooling problems
- Solving homogeneous ODEs
- Solving non-homogeneous ODEs

WEEK 12-13:**Curve Fitting**

Linear and Non-linear curve fitting

- Fitting a straight line
- Fitting non-linear curves

WEEK 14:

Lab Internal Exam

TEXT BOOKS:

1. The Fundamentals of MATLAB: First Programs, Kenneth A. Lambert, Cengage Learnings, 2011
2. Think MATLAB, Allen B. Downey, 1st Edition, Orielly
3. Introduction to MATLAB Programming, William Mitchell, Povel Solin, Martin Novak et al., NCLab Public Computing, 2012
4. Introduction to MATLAB Programming, Jacob Fredslund, 2007

REFERENCES:

1. An Introduction to MATLAB, John C. Lusth, The University of Alabama, 2011
2. Introduction to MATLAB, Dave Kuhlman, 2008

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. III Semester

(25PC2RA201) OBJECT ORIENTED PROGRAMMING THROUGH C++ LABORATORY

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE PRE-REQUISITES: Programming for Problem Solving

COURSE OBJECTIVES:

- To develop basic C++ programming skills using functions, arrays, and dynamic memory allocation
- To apply object-oriented concepts such as classes, constructors, overloading, and inheritance in C++ programs
- To implement polymorphism using virtual functions and abstract classes
- To use pointers, files, and stream operations for effective data handling
- To design reliable C++ programs using exception-handling and multithreading mechanisms

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Implement C++ programs using basic programming constructs, functions, arrays, and dynamic memory allocation

CO-2: Develop programs using object-oriented concepts such as classes, constructors, function and operator overloading

CO-3: Apply inheritance and runtime polymorphism using virtual functions and abstract classes

CO-4: Perform file handling and stream-based input/output operations using pointers

CO-5: Design robust C++ programs using exception-handling and multithreading mechanisms

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	3	1	-	2	-	-	-	-	-	1	2	-	-
CO-2	3	3	3	-	2	-	-	-	-	-	1	2	-	-
CO-3	2	2	3	2	2	-	-	-	-	-	1	2	-	-
CO-4	2	3	2	2	3	-	-	-	-	-	2	2	-	-
CO-5	2	3	2	2	2	-	-	-	-	-	2	2	-	-

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

WEEK 1:

- a) Write a C++ program to declare Structure. Initialize and display contents of member variables.
- b) Write a C++ Program to display Names, Roll No., and grades of 3 students who have appeared in the examination. Declare the class of name, Roll No. and grade. Create an array of class objects. Read and display the contents of the array.

- c) Write a C++ program to define a class employee having members Emp-id, Emp-name, basic salary and functions accept() and display(). Calculate DA=25% of basic salary, HRA=800, I-tax=15% of basic salary. Display the pay slip using appropriate output format. 1) To accept the data 2) To display the data

WEEK 2:

- a) Write a C++ Program to create object and release them using destructor.
b) Write a C++ program to demonstrate a parameterized constructor that accepts values from the user and initializes class data members.
c) Write a C++ Program to Demonstrate Function Overloading by Using Same Function Name with Different Data Types.

WEEK 3:

- a) Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.
b) Write a C++ program to Count Number of Objects Using Static Data Members and Static Member Functions.
c) Write a C++ program to demonstrate the use of the this pointer.

WEEK 4:

- a) Write a C++ program to allocate memory using new operator.
b) Write a C++ program to Illustrate the Use of a Friend Function and Friend Class.
c) Write a C++ program to Illustrate the Use of a Destructors.

WEEK 5:

- a) Write a C++ program to Demonstrate the Concept of Multilevel Inheritance.
b) Write a C++ program to Illustrate Multiple Inheritance.
c) Write a C++ program to use pointer for both base and derived classes and call the member function. Use Virtual keyword.

WEEK 6:

- a) Write a C++ program to Demonstrate the Use of Virtual Function.
b) Write a C++ program to Implement Run Time Polymorphism Using Base Class reference.
c) Write a C++ program to Illustrate the Concept of Abstract Class.

WEEK 7:

- a) Write a C++ program using a class to read a text file name from the user and display its contents on the console. Handle the case where the file does not exist.
b) Write a C++ program using a class that accepts two input file names from the user and creates a new output file containing the combined contents of both files. Handle the case where the files does not exist.
c) Write a C++ program to simulate a simple banking system using console I/O that allows deposit, withdrawal, and balance enquiry.

WEEK 8:

- a) Create a C++ class named Complex to represent complex numbers in the form $a+bi$. Write a C++ program that uses operator overloading to perform addition and multiplication of two complex numbers.
b) Create a C++ class named Vector to represent a mathematical vector. Write a C++ program that uses unary operator overloading to perform operations such as negation on the vector.

- c) Write a C++ program to overload the << operator using a friend function to display the components of a 2D vector.

WEEK 9:

- a) Write a C++ program to implement a class template for a stack that supports push and pop operations for different data types.
- b) Write a C++ program to implement a class template Employee that stores employee ID, name, and salary using different data types and displays the details.
- c) Write a C++ program using function templates to compute the sum, average, and maximum of elements in an array of any data type.

WEEK 10:

- a) Write a C++ program using a class to handle division by zero and array index out of bounds exceptions.
- b) Write a C++ program using a class to demonstrate multiple catch blocks for handling different runtime exceptions.
- c) Write a C++ program using a class to illustrate rethrowing of an exception from a member function to the main function.

WEEK 11:

- a) Write a C++ program to create and run a thread using std::thread.
- b) Write a C++ program to execute multiple threads concurrently.
- c) Write a C++ program to pass arguments to a thread function.

WEEK 12:

- a) Write a C++ program to enable inter-thread communication using condition_variable.
- b) Write a C++ program to solve the producer-consumer problem using threads.

TEXT BOOKS:

1. The Complete Reference C++, Herbert Schildt, 4th Edition, Tata McGraw-Hill
2. Problem solving with C++: The Object of Programming, Walter Savitch, 4th Edition, Pearson Education

REFERENCES:

1. The C++ Programming Language, B. Stroutstrup, 3rd Edition, Pearson Education
2. OOP in C++, T. Gaddis, J. Walters and G. Muganda, 3rd Edition, Wiley Dream Tech Press
3. Object Oriented Programming in C++, R. Lafore, 3rd Edition, Galigotia Publications

ONLINE RESOURCES:

1. <https://www.w3schools.com/cpp>
2. <https://www.geeksforgeeks.org/cpp>

B. Tech. III Semester

(25PC2RA202) SENSORS AND ACTUATORS LABORATORY

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE OBJECTIVES:

- To process Proprioceptive and Exteroceptive sensor data and analyze Sensor properties
- To analyze the hardware differences between Rolling and Global shutter camera systems
- To implement computer vision algorithms for object tracking and depth estimation
- To control electric actuators (DC, Stepper, BLDC) and implement a closed-loop perception-action system

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Characterize sensor noise and drift in IMUs and calibrate encoders for odometry

CO-2: Deploy vision hardware configurations to mitigate motion artifacts using Global Shutter technology

CO-3: Utilize LiDAR and Depth cameras to generate environmental maps and distance data

CO-4: Design an integrated robotic system that uses vision-based feedback to control servo actuated mechanisms

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	3	3	3	3	2	1	2	2	-	-	-	-	3
CO-2	3	3	3	3	3	2	1	2	2	-	-	-	-	3
CO-3	3	3	3	3	3	2	1	2	2	-	-	-	-	3
CO-4	3	3	3	3	3	2	1	2	2	-	-	-	-	3

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

CYCLE I: SENSORS & PERCEPTION

EXP-1. Odometry and Kinematics: Process quadrature encoder data from a live sensor or dataset to estimate the displacement and velocity of a robotic platform.

EXP-2. Inertial Data Analysis: Characterize noise and bias from an Inertial Measurement Unit (IMU) stream to estimate stable orientation via digital filtering.

EXP-3. Ranging and Error Characterization: Analyze Ultrasonic and Infrared distance data (Live or Logged) to model sensor accuracy and identify technology-specific limitations.

EXP-4. Shutter and Motion Artifacts: Evaluate image sequences from Rolling and Global shutter sources to quantify geometric distortion in dynamic scenes.

EXP-5. Geometric Camera Calibration: Utilize a dataset of calibration patterns to determine intrinsic parameters and rectify lens-induced distortions.

EXP-6. Object Detection and Tracking: Implement a vision pipeline to extract the spatial coordinates of a target from a live camera feed or a recorded video stream.

EXP-7. Volumetric Depth Interpretation: Process 3D point clouds or depth maps from a vision sensor to extract spatial geometry and distance metrics.

EXP-8. Planar Environment Mapping: Analyze 2D LiDAR scan data (Live or ROS bag) to generate polar plots and identify obstacles in a 2D plane.

CYCLE II: ACTUATORS & CONTROL

EXP-9. DC Motor Characterization: Control a DC motor via an H-Bridge and analyze current consumption under varying load conditions.

EXP-10. Stepper Motor Precision & Resonance: Interface a stepper motor with a micro-stepping driver to analyze motion smoothness and vibration.

EXP-11. Brushless DC (BLDC) Speed Control: Interface a BLDC motor with an electronic controller to map control signals to angular velocity.

EXP-12. Integrated Visual Servoing: Design a closed-loop system where vision feedback controls an actuated mechanism for target tracking.

TEXTBOOKS:

1. Sensors and Actuators: Engineering System Instrumentation, Clarence W. de Silva, CRC Press
2. Computer Vision: Algorithms and Applications, Richard Szeliski, Springer

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. III Semester

(25PC2RA211) SOFTWARE SYSTEMS FOR ROBOTICS AND AI LABORATORY

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE OBJECTIVES:

- To master the Linux Command Line Interface (CLI) for file management, process control, and automation
- To utilize Version Control Systems (Git) for collaboration and implement basic CI/CD pipelines
- To configure development environments using Virtualization (Conda) and Containerization (Docker)
- To implement Inter-Process Communication (IPC) using Network Sockets to understand distributed systems
- To leverage AI-assisted coding tools (GitHub Copilot) to accelerate development and debugging

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Demonstrate proficiency in Linux shell scripting, file permissions, and remote server management via SSH

CO-2: Manage code repositories using Git, resolving merge conflicts, and utilizing GitHub Actions for automation

CO-3: Construct isolated Python environments and Docker containers to ensure software reproducibility

CO-4: Develop client-server applications using TCP/IP sockets to exchange data between distinct processes

CO-5: Build and deploy a complete software module (e.g., a Web Dashboard) using AI- assisted programming workflows

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	2	1	-	-	3	-	-	-	1	-	2	-	2	1
CO-2	2	1	2	-	3	-	-	3	2	2	2	-	2	1
CO-3	2	1	2	-	3	-	-	-	1	-	2	-	3	2
CO-4	3	2	2	-	2	-	-	1	1	-	1	2	2	1
CO-5	2	1	3	-	3	-	-	1	2	1	3	1	2	3

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

The student is expected to perform the following 12 compulsory experiments.

Part A: The Linux Ecosystem & Version Control

EXP-1. The Shell & CLI Mastery:

Navigate the Linux File System via Terminal. Master commands: 'ls', 'cd', 'grep', 'find', 'cat', 'chmod', 'chown'. Implement "Piping" (|) and Redirection (>, >>) to chain commands (e.g., search logs for errors and save to a file).

EXP-2. Shell Scripting for Automation:

Write a Bash script to automate a tedious task (e.g., "Backup all '.py' files to a timestamped folder"). Use Variables, Loops ('for'/'while'), and Conditionals ('if-else').

EXP-3. Remote Access with SSH:

Configure 'ssh' to connect to a remote server (or peer machine). Generate SSH Keys ('ssh-keygen') for password-less login. Use 'scp' (Secure Copy) or 'rsync' to transfer large datasets between machines.

EXP-4. Git & GitHub Fundamentals:

Initialize a repo, stage files, commit, and push to GitHub. Demonstrate the workflow of 'git clone', 'git pull', and 'git log'. Create a '.gitignore' file to exclude binary/env files.

EXP-5. Collaboration & Merge Conflicts:

Simulate a team workflow. Create a "Feature Branch," make changes, and open a Pull Request (PR). Intentionally create a "Merge Conflict" with a peer and resolve it manually in the editor.

EXP-6. Introduction to CI/CD (GitHub Actions):

Create a simple GitHub Actions workflow ('.yaml') that automatically runs a Python script (e.g., "Hello World" or a basic Unit Test) whenever code is pushed to the repository.

Part B: Deployment, IPC, and AI Tools

EXP-7. Environment Management (Conda/Venv):

Create specific Python environments for conflicting projects (e.g., "Project A: Python 3.7 + TensorFlow", "Project B: Python 3.10 + PyTorch"). Export the environment to 'requirements.txt' or 'environment.yml' and replicate it on a different machine.

EXP-8. Build Systems (Intro to CMake):

Write a 'CMakeLists.txt' to compile a simple C++ "Hello World" project. Understand the concepts of 'add_executable' and linking libraries (essential for ROS).

EXP-9. Containerization with Docker:

Write a 'Dockerfile' to containerize a Python script. Build the image, run the container, and verify it works isolated from the host OS.

EXP-10. Socket Programming (TCP/IP):

Write a Python "Server" script that listens on a port and a "Client" script that sends messages. Task: Create a "Chat Application" or a "Remote Controller" where the client sends specific keys ('w', 'a', 's', 'd') and the server prints the movement command.

EXP-11. Serial Communication (UART):

(Hardware or Virtual) Read data from a serial port (e.g., Arduino connected via USB or Virtual Serial Port). Use the 'pyserial' library to parse incoming CSV sensor data (e.g., "Temperature, Humidity").

EXP-12. AI-Assisted Full Stack Development:

Challenge: Use GitHub Copilot / Cursor AI to build a Streamlit/Gradio Web Dashboard that visualizes a CSV dataset. Constraint: The student should write minimal code manually; they must use Prompt Engineering to guide the AI to generate the plotting logic and UI layout.

TEXT BOOK:

1. The Missing Semester of Your CS Education, Anish Athalye, Jose Javier Gonzalez Ortiz, and Jon Gjengset

REFERENCES:

1. Scott Chacon and Ben Straub, Pro Git, Apress
2. Official Documentation, Docker, GitHub Actions, and CMake

B. Tech. III Semester

(25ES2CY101) PYTHON PROGRAMMING LABORATORY

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE OBJECTIVES:

- To install and run the Python interpreter & IDE's
- To learn control structures
- To understand advanced Data Types, Functions in Python
- To handle Exceptions, Files and OOps concepts in Python
- To understand Python Advanced modules like numpy

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Apply comprehensions, different Decision-Making statements and Functions

CO-2: Implement various data types like lists, tuples, strings

CO-3: Use different File handling operations and Maps

CO-4: Apply Object oriented programming in Python types

CO-5: Use Numpy in developing mathematical applications

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	3	2	2	2	-	-	-	-	1	2	3	2	3
CO-2	3	2	3	2	2	-	-	-	-	2	2	2	1	2
CO-3	3	3	3	3	2	-	2	-	-	3	2	2	3	1
CO-4	3	3	2	3	3	-	-	-	-	3	3	2	2	2
CO-5	3	2	3	3	3	-	-	2	-	3	3	2	3	2

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

WEEK 01:

- Install Python and configure Interpreter & IDE like Jupyter, google colab.
- Write and execute your first Python program to display "Hello, Python!".
- Demonstrate interactive mode and script mode.
- Use command-line arguments to run Python scripts.
- Write a program to find the largest element among three Numbers.
- Write a program to print the sum of all the even numbers in the range 1 - 50 and print the even sum.
- Write a Program to display all prime numbers within an interval of given X1 & X2.

WEEK 02:

- Program to demonstrate Python's built-in data types.

- b) Type conversion using int(), float(), str(), etc.
- c) Calculate Simple Interest and Area of a Circle.
- d) Program to swap two numbers without using a third variable.

WEEK 03:

- a) Write a program to print the following patterns using loop:

```
*
**
***
****
```

- b) Write a program to print multiplication table of a given number X1 to range X2.
- c) Grade calculation using if-elif-else.
- d) Determine the type of a triangle (equilateral, isosceles, scalene).

WEEK 04:

- a) Write a program to find the length of the string without using any library functions.
- b) Write a program to check if two strings are anagrams or not.
- c) Write a program to check if the substring is present in a given string or not.
- d) Write a program to perform the given operations on a list:
 - i. add ii. insert iii. slicing
- e) Write a program to perform any 5 built-in functions by taking any list.
- f) Write a program to get a list of the even numbers from a given list of numbers. (use only comprehensions)

WEEK 05:

- a) Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenated tuples.
- b) Write a program to return the top 'n' most frequently occurring chars and their respective position.
- c) Write a program on counts. e. g. aaaaabbbbccccc, it should return ex:


```
[(a 5) (b 4) (c 4)]
```
- d) Write a program to count the number of vowels in a string.
- e) Write a program that displays which letters are present in both strings.
- f) Write a program to sort given list of strings in the order of their vowel counts.

WEEK 06:

- a) Write a program to check if a given key exists in a dictionary or not.
- b) Write a program to add a new key-value pair to an existing dictionary.
- c) Write a program to sum all the items in a given dictionary.
- d) Demonstrate functions with different argument types: positional, keyword, default, variable-length.
- e) Program to show local vs global variable behavior.
- f) Recursive function to calculate factorial.
- g) Recursive Fibonacci sequence generator.

WEEK 07:

- a) Write a program to sort words in a file and put them in another file. The output file should have only lower-case words, so any upper case words from source must be lowered.
- b) Write a program to find the most frequent words in a text. (read from a text file).

WEEK 08:

Lab internal -1

WEEK 09:

- a) Program using try, except, else, and finally.
- b) Raise and handle custom exceptions.
- c) Read and write data to a text file.
- d) Count the number of words and lines in a text file.

WEEK 10:

- a) Write a Python class named Person with attributes name, age, weight (kgs), height (ft) and takes them through the constructor and exposes a method get_bmi_result() which returns one of "underweight", "healthy", "obese".
- b) Write a Python class named Circle constructed by a radius and two methods which will compute the area and the perimeter of a circle.

WEEK 11:

- a) Accept two lists, one list represents temperatures in Fahrenheit and another list represents temperatures in Celsius. Perform map operations Fahrenheit-Celsius and Celsius- Fahrenheit using lambda
- b) Create a Fibonacci sequence that contains 'N' terms, and filter only even terms using lambda
- c) Write a program to find the number of rows in a text file using Generator and yield.
- d) Find Sum of Squares of 1 to n numbers using Generator Expressions.

WEEK 12:

Let's take a password as a combination of alphanumeric characters along with special characters, and check whether the password is valid or not with the help of a few conditions.

Conditions for a valid password are:

- Should have at least one number.
- Should have at least one uppercase and one lowercase character.
- Should have at least one special symbol.
- Should be between 6 to 20 characters long.

Input : Gvpce12# Output : Password is valid.

Input : asd123 Output : Invalid Password !!

WEEK 13:

- a) Write a Python Program to demonstrate numpy arrays creation using array () function.
- b) Write a python program to demonstrate use of ndim, shape, size, dtype.
- c) Python program to demonstrate basic slicing, integer and Boolean indexing.
- d) Write a python program to find min, max, sum, cumulative sum of array.
- e) Create two single dimensional NumPy arrays, one is height, and another is weight, calculate BMI (weight/height**2) and keep all BMI values in another NumPy Array. Calculate mean, median, and standard deviation of BMI values.

WEEK 14:

Lab internal 2

TEXT BOOKS:

1. Python with Machine Learning, A. Krishna Mohan, T. Murali Mohan & Karunakar, 1st Edition, S. Chand Publications, 2019
2. Introduction to Programming using Python, Y. Daniel Liang, 1st Edition, Pearson Publications, 2017

REFERENCES:

1. Python Programming A Modular Approach, Sheetal Taneja, 1st Edition, Pearson Publications, 2017
2. Effective Python: 59 Specific Ways to Write Better Python, I/C, Brett Slatkin (C), 1st Edition, Pearson Publications, 2015
3. Programming and Problem Solving with Python, Ashok Namdev Kamathane and Amit Ashok Kamathane, 1st Edition, McGraw-Hill Education, 2017

B. Tech. III Semester

(25SD5CS201) DATA VISUALIZATION LABORATORY

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE OBJECTIVES:

- To install and run the Rstudio for data analysis
- To understand principles and techniques for data visualization
- To visualize the data that can improve comprehension, communication, and decision making
- To implement various tools and methods for easy interpretation of data

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Understand the importance of data visualization in analytics

CO-2: Gain knowledge in the principles of data visualization

CO-3: Apply the principles of data visualization on toy data sets using R

CO-4: Analyze data towards decision making using visualization

CO-5: Identify appropriate/suitable visualization for particular requirements imposed by the data type and analytics algorithms

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	2	2	1	-	1	-	-	-	1	-	2	2	1	1
CO-2	2	2	2	1	2	-	-	-	1	-	2	3	1	1
CO-3	2	3	2	2	3	-	-	-	1	-	2	3	2	2
CO-4	2	3	2	2	2	1	-	-	2	1	2	3	2	2
CO-5	2	3	3	1	3	-	-	-	2	-	2	3	2	2

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

WEEK 1:

Basics

Introduction to basic components of R programming, overview of visualization, data types, basics of plotting graphs, different types of graphs in analytics

WEEK 2:

Importance of Visualizations

Principles of communicating data, Principles of encoding data to make visualizations, Importance of color in visualizations

WEEK 3:**Plots Using Basic R**

Plots with one categorical, continuous variable

Functions in R for plotting, plots with one categorical variable, plots with one continuous variable, plots with one categorical and one continuous variable

Plots with 2 Continuous Variables

Plots with two continuous variables, controlling various aesthetics of the graph.

WEEK 4:**ggplot2 in R**

Group manipulation and data reshaping in R, understanding the philosophy of ggplot2, bar plot, piechart, histogram, box plot, scatter plot and regression plots

WEEK 5:**ggplot2 in R**

Controlling aesthetics like color, size, legend and facets.

WEEK 6:**Data Visualization**

Importing the data, Dimensions and measures, color code for various types of variables

WEEK 7:**Working with Sheets**

Understanding the worksheet, row and column shelves, showme card, filter and pages shelf

WEEK 8:**Calculations**

Working with different measures, creating new calculated fields, Quick table calculations, parameters and groups

WEEK 9:**Graphs for Analytics**

Calculate Proportions and percentages, Comparing current to historical and Actual to Target

WEEK 10:**Normal Distribution Variation**

Calculate Mean and Median–Normal Distribution Variation and Uncertainty

WEEK 11:**Reporting and Visualizing Variation**

Reporting and Visualizing variation, Control Charts Multiple Quantities –Scatter Plots, Stacked bars, Regressions and trend lines

WEEK 12:**Depicting Changes Over Time**

Depicting changes over time, Line Chart, Dual Axis Line Chart, scatter plot

WEEK 13:**Reporting**

Introduction to dashboard, use of filters in dashboard, Imbedding pictures, Insert live webpage, story

CASE STUDY:

1. Student Performance Analytics Dashboard
2. Sales Performance Analysis for Retail Store
3. Hospital Patient Data Visualization
4. Weather Trend and Climate Analysis
5. Employee Performance and HR Analytics
6. Website Traffic & User Behavior Analysis

WEEK 14:

Internal Exam

TEXT BOOKS:

1. Microsoft Power BI, Marco Russo
2. R for everyone, Jarad P Lander
3. Statistics: An Introduction using R, Michael J. Crawley

REFERENCE:

1. Think Python, Allen Downey, Green Tea Press

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. III Semester

(25MN6HS102) ENVIRONMENTAL SCIENCE

TEACHING SCHEME		
L	T/P	C
2	0	0

EVALUATION SCHEME		
SE-I	SE-II	TOTAL
50	50	100

COURSE PRE-REQUISITES: Basic knowledge on environmental issues

COURSE OBJECTIVES:

- To recognize the principles of environment
- To identify the sources and causes of Environmental Pollution
- To analyze the causes of climate change and identify climate change mitigation and adaptation technologies
- To list out the benefits in creating circular economy and sustainability
- To emphasize the Environmental conventions, policies and regulations

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Gain knowledge about principles of Environment and allied problems

CO-2: Identify the causes and suggest remedial measures for Environmental Pollution

CO-3: Appraise the mitigation measures and adaptation technologies related to climate change

CO-4: Familiarize with the importance of circular economy towards sustainability

CO-5: Relate and implement environmental policies and regulations

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	2	2	1	-	1	2	2	1	1	1	2	-	-	-
CO-2	2	2	1	-	1	2	2	1	1	1	2	-	-	-
CO-3	2	2	1	-	1	2	2	1	1	1	2	-	-	-
CO-4	2	2	1	-	1	2	2	1	1	1	2	-	-	-
CO-5	-	1	1	-	-	2	1	1	1	1	-	-	-	-

MODULE 1:

Introduction to Environmental Science: Importance and Scope of Environmental Science. Overview of Environment & its components. Ecosystem function and uses. Biodiversity – Types and Values.

MODULE 2:

Synergy With Environment: Relationship between environmental science & society – Influence of Industry & infrastructure on environment (Specific case studies). Sources, Impacts and prevention of air, water and soil pollution.

MODULE 3:

Climate Change: Science behind climate change – factors responsible for climate change, Role of greenhouse gases – Global temperature rise & its impact on environment & human health. Carbon footprint – Briefing on Paris agreement, Identify

key sectors for low carbon footprint. Climate change mitigation & adaptation strategies.

MODULE 4:

Moving Towards Sustainability: EIA and EMP - importance, Sustainable agriculture – Importance of Organic farming and hydroponics. Role of AI & IOT for efficient management of environmental issues – Health, air, water, and soil. Sustainable living practices – Principles and methods (Linear Economy vs circular economy). Green Buildings.

MODULE 5:

Policies and Regulations: International conventions: Montreal Protocol, COP15, Kyoto protocol (Salient Features). National Environmental Policies – Salient features of EPA act, Air act, Water Act, Forest and Wildlife act. National Green Tribunal Act (2010)- Features and Significance. Extended producer responsibility.innovative approaches to waste management: Plastic, tyre and batteries.

TEXT BOOKS:

1. Living in the Environment, G. Tyler Miller/ Scott E. Spoolman, 20th Edition, Cengage Learning
2. Environmental Studies, Rajagopalan, Oxford University Press, 2005
3. Introduction to Climate Change, Andreas Schmittner, Oregon State University, 2018. <https://open.oregonstate.edu/climatechange/>

REFERENCES:

1. Green Development: Environment and Sustainability in a Developing World, Bill Adams, 4th Edition, Routledge Publishers, 2021
2. Fixing Climate, Robert Kunzig & Wallace S. Broecker, Profile Books Publisher, 2009
3. Plastic Waste and Recycling-Environmental Impact, Societal Issues, Prevention and Solutions, 1st Edition, Edited by Trevor Letcher, Academic Press, 2020
4. Waste Management Practices: Municipal, Hazardous, and Industrial by John Pichtel, CRC Press, 2020
5. Environmental Science: A Global Concern, William P. Cunningham, Mary Ann Cunningham, McGraw Hill Education, 2022

ONLINE RESOURCE:

1. https://www.google.co.in/books/edition/Waste_Management_Practices/pG3SBQAAQBAJ?hl=en&gbpv=1&pg=PA1&printsec=frontcover

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. IV Semester

(25PC1RA203) ROBOT OPERATING SYSTEM

TEACHING SCHEME		
L	T/P	C
2	0	2

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE OBJECTIVES:

- To understand the distributed computational graph architecture of the Robot Operating System (ROS)
- To master the various inter-process communication mechanisms (Topics, Services, Actions) provided by ROS
- To utilize ROS tools for debugging, visualization (RViz), and data recording (Rosbag)
- To learn model robot geometry and physical properties using the Unified Robot Description Format (URDF)
- To implement system integration by composing complex robotic applications using launch files and parameter servers

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Explain the role of the ROS Master, Nodes, and the file system structure in a robotic software stack

CO-2: Develop modular software nodes that communicate using asynchronous topics and synchronous services

CO-3: Analyze the suitability of Action Lib for long-running robotic tasks compared to Services

CO-4: Construct a functional robot model (URDF) and visualize its coordinate frames using the TF library

CO-5: Utilize standard ROS tools (rviz, rqt, rosbag) to debug and validate the behavior of a robotic system

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	1	1	3	1	1	1	-	-	2	3	2	2
CO-2	3	3	2	2	3	1	1	1	-	-	2	3	3	2
CO-3	3	3	3	3	3	2	1	1	2	1	2	3	3	3
CO-4	3	3	3	3	3	2	2	2	2	2	3	3	3	3
CO-5	2	2	2	2	2	3	3	3	3	2	3	2	3	3

UNIT-I:

ROS Architecture & Ecosystem: Introduction to Middleware, The ROS Graph, The Master Node, Nodes, Peer-to-Peer Communication, Parameter Server, ROS File system, Packages, Stacks, CMakeLists.txt, package.xml, The Catkin Build System, Workspaces, Environment Variables, ROS Distributions.

UNIT-II:

Synchronous & Asynchronous Communication: Publish-Subscribe Model, Topics, Message Types (.msg), Standard Messages, Custom Message Generation, Subscribers, Publishers, Service-Client Model, Services (.srv), Request-Response, Custom Service Generation, Command Line Tools (rostopic, rosservice).

UNIT-III:

Advanced Communication & Coordination: Action Lib Architecture, Goal-Feedback-Result, Action Definition Files (.action), Action Servers, Action Clients, Launch Files, XML Syntax for Launch, Remapping Names, Namespaces, Grouping Nodes, Introduction to rosparam, Dynamic Reconfigure.

UNIT-IV:

Transforms, Modelling & Visualization: Unified Robot Description Format (URDF), XML Macros (Xacro), Links, Joints, Visual vs Collision Geometry, The TF2 Library, Coordinate Frames, The TF Tree, Static Transform Publisher, Broadcasting and Listening to Transforms, RViz Visualization, Markers

UNIT-V:

System Integration & Tools: ROS Bags (Recording and Playback), Filtering Bag files, Debugging Tools, RQT Graph, RQT Plot, RQT Console, Logging Levels (INFO, WARN, ERROR), Introduction to Gazebo Simulation Interface, Sensor Plugins, Configuration of Navigation Stack (cost maps and move base integration overview).

TEXT BOOKS:

1. Programming Robots with ROS: A Practical Introduction to the Robot Operating System, Quigley M., Gerkey B., & Smart W. D., O'Reilly Media, 2015

REFERENCE:

1. Mastering ROS for Robotics Programming, Joseph .L, Packt Publishing, 2018
2. A Gentle Introduction to ROS, O'Kane J. M., Available Online, 2014

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. IV Semester

(25PC1AM201) MACHINE LEARNING

TEACHING SCHEME		
L	T/P	C
3	0	3

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE OBJECTIVES:

- To introduce the basic concepts and techniques of Machine Learning
- To understand Feature engineering and various probability-based learning techniques
- To have a thorough understanding of the supervised learning techniques
- To learn the concepts of unsupervised learning techniques
- To understand the need for neural networks, deep learning, and reinforcement learning in solving real-time problems

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Demonstrate the foundational concepts and principles of machine learning

CO-2: Apply the feature engineering and probability techniques in machine learning

CO-3: Apply supervised learning algorithms, including regression and classification, to real-world problems

CO-4: Interpret the outcomes of unsupervised learning algorithms and convert them into actionable insights for practical applications

CO-5: Apply neural networks, deep learning, and reinforcement learning techniques to real-world case studies for effective problem-solving

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	1	1	2	-	-	-	-	-	-	3	2	1
CO-2	3	3	2	2	3	1	-	1	-	-	1	3	3	2
CO-3	3	3	3	2	3	2	1	1	-	2	-	3	3	2
CO-4	2	3	2	3	3	2	1	1	2	2	1	3	3	2
CO-5	3	3	3	2	3	2	2	1	2	2	1	3	3	3

UNIT-I:

Introduction to Machine Learning: Types of Human learning, machine learning process, Well-posed learning problem, Types of machine learning and comparison, applications of machine learning. Model Preparation, Evaluation and feature engineering: Machine learning activities, Types of data in machine learning, dataset understanding, plotting and exploration, checking data quality, remediation, data pre-processing, selecting a model, predictive and descriptive models, supervised learning model training, cross-validation and boot strapping, lazy vs eager learner, interpreting the model- underfitting, overfitting, bias-variance trade-off. Parameter for evaluating the performance of classification, regression, and clustering models. Improving the performance of a model.

UNIT-II:

Feature Engineering: Feature transformation - feature construction, feature extraction by PCA, SVD, LDA. Feature subset selection – feature relevancy and redundancy measures. Feature selection process and approaches. Review of Probability concepts: joint probability, conditional probability, Bayes' rule, Common discrete and continuous distributions, dealing with multiple random variables, and the central limit theorem. Bayes classifier, Multi-class Classification, Naïve Bayes classifier, Bayesian belief network.

UNIT-III:

Supervised Learning - Introduction to supervised learning, Regression: Introduction of regression, Regression algorithms: Simple linear regression, Multiple linear regression, Polynomial regression model, Logistic regression, Maximum likelihood estimation. Classification: Classification model and learning steps, Classification algorithms: Naïve Bayes classifier, Distance measures, k-Nearest Neighbor (kNN), Decision tree, Support vector machines, Kernel trick, Random Forest.

UNIT-IV:

Unsupervised Learning Introduction to unsupervised learning, Unsupervised vs supervised learning, Application of unsupervised learning, Clustering and its types, Partitioning method: k-Means and K-Medoids, Hierarchical clustering, Density-based methods – DBSCAN

UNIT-V:

Artificial Neural Network: Biological neuron, Artificial neuron, Activation functions, neural network architecture, perceptron, learning process in ANN, Back propagation. Introduction to deep learning, overview of reinforcement learning, Representation learning, Evolutionary learning. Case-study of ML applications: Image recognition, Email spam filtering, Online fraud detection.

TEXT BOOKS:

1. Machine Learning, Saikat Dutt, S. Chjandramouli, Das, 1st Edition, Pearson
2. Machine Learning, M N Murty, Anathanarayana V S, 1st Edition, University Press
3. Machine Learning, Tom M Mitchell, 1st Edition, McGraw-Hill Education, 2013

REFERENCES:

1. Machine Learning – An Algorithmic Perspective, Stephen Marsland, 2nd Edition Chapman and Hall/CRC
2. Machine Learning and Pattern Recognition Series, 2014

ONLINE RESOURCES:

1. <https://www.coursera.org/learn/machine-learning>
2. <https://nptel.ac.in/courses/106/106/106106202/>

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. IV Semester

(25PC1RA204) KINEMATICS AND DYNAMICS OF ROBOTS

TEACHING SCHEME		
L	T/P	C
3	0	3

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE OBJECTIVES:

- To establish a strong foundation in single-body mechanics, including 3D vector algebra and rigid body spatial descriptions
- To formulate Forward Kinematics for serial manipulators using the classic Denavit-Hartenberg (DH) convention
- To develop analytical and algebraic solutions for Inverse Kinematics in various robotic configurations
- To analyze differential motion and robot velocities using the Manipulator Jacobian and singularity analysis
- To apply Newtonian and Lagrangian mechanics to derive the dynamic equations of motion and perform static force analysis

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Represent rigid body position and orientation in 3D space using Rotation Matrices and Homogeneous Transformations

CO-2: Compute Forward Kinematics for multi-link open-chain manipulators using D-H parameters

CO-3: Derive Inverse Kinematic solutions for 3-DOF and 6-DOF planar and spatial robotic arms

CO-4: Interpret the Robot Jacobian to identify singularities and relate joint velocities to end-effector motion

CO-5: Formulate dynamic equations of motion using Euler-Lagrange mechanics and perform static force/moment transformations

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	1	2	2	-	-	-	1	1	1	3	1	1
CO-2	3	2	2	2	3	-	-	-	1	1	1	3	1	2
CO-3	3	3	2	3	3	-	-	-	2	1	1	3	2	2
CO-4	3	3	3	2	3	-	-	-	2	1	1	2	2	3
CO-5	3	3	3	2	3	-	-	-	2	1	1	1	3	3

UNIT-I:

Rigid Body Foundations and Spatial Analysis

Position Analysis: Robots as Mechanisms, Matrix Representation of a point and vector in space, Representation of a frame at the origin and relative to a fixed reference frame, Representation of a rigid body (Pose).

Transformations: Homogeneous Transformation Matrices (HTM), Representation of pure translation and pure rotations, Inverse of transformation matrices.

UNIT-II:

Forward Kinematics (D-H Convention): Introduction to Forward Kinematics, Forward Kinematic Equations for Position and Orientation,

Denavit-Hartenberg (D-H) Representation: Link and Joint descriptions, D-H parameter table (a , α , d , θ), Assigning coordinate frames, Deriving the Forward Kinematic equations for common robots (Articulated, SCARA, Cylindrical).

UNIT-III:

Inverse Kinematics: The Inverse Kinematic Solution of Robots, Workspace analysis,

Analytical Solutions: Algebraic vs. Geometric approaches, Inverse Kinematic programming of robots, Solvability, Degeneracy and Dexterity, The fundamental problem with D-H representation in Inverse Kinematics.

UNIT-IV:

Differential Motions and Jacobians

Introduction to Differential Relationships, The Jacobian Matrix: Differential motions of a frame versus a robot, Calculation of the Jacobian for serial manipulators, Interpretation of differential changes, Singularity Analysis: Decoupling singularities, The Inverse Jacobian, Relationship between the Jacobian and the Differential Operator.

UNIT-V:

Dynamic Analysis and Force Analysis

Lagrangian Mechanics: Short overview of energy-based dynamics, Kinetic and Potential energy formulations, Effective moments of inertia, Dynamic equations for multiple-DOF robots.

Force Analysis: Static force analysis of robots, Transformation of forces and moments between coordinate frames, Relationship between joint torques and end-effector forces.

TEXT BOOK:

1. Introduction to Robotics: Analysis, Control, Applications, Saeed B. Niku, 3rd Edition, Wiley

REFERENCES:

1. Introduction to Robotics, K. Saha, McGraw-Hill Education. (For Lab implementation and Robo Analyzer)
2. Robot Modeling and Control, Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, Wiley (For advanced control and Euler-Lagrange depth)

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. IV Semester

(25PC1EC202) SIGNALS AND SYSTEMS

TEACHING SCHEME		
L	T/P	C
3	0	3

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE PRE-REQUISITES: Ordinary Differential Equations, Laplace Transforms and Vector Calculus

COURSE OBJECTIVES:

- To understand various fundamental characteristics of signals and systems
- To represent the signal in frequency domain using transform techniques
- To analyse, design continuous and discrete systems
- To study the operations of convolution, correlation and the effects of sampling
- To understand Laplace and Z-transform properties for the analysis of signals and systems

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Represent, classify, and manipulate continuous-time and discrete-time signals using standard operations

CO-2: Apply Fourier series and transforms to determine the frequency-domain characteristics of signals and systems

CO-3: Implement Laplace and Z-Transforms for the analysis of continuous and discrete signals, and for computing inverse transforms

CO-4: Analyze the performance of LTI systems using convolution and correlation

CO-5: Apply the sampling theorem to discrete-time signal processing, reconstruct signals from samples, and evaluate aliasing effects

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	3	-	-	-	-	-	-	-	-	-	2	1	1
CO-2	3	3	2	-	-	-	-	-	-	-	-	2	1	2
CO-3	3	3	2	-	-	-	-	-	-	-	-	2	2	2
CO-4	3	3	2	2	2	-	-	-	-	-	-	2	1	2
CO-5	3	3	2	2	2	-	-	-	-	-	-	2	3	2

UNIT-I:

Representation of Signals: Continuous time and Discrete Time signals, Classification of Signals – Periodic and aperiodic, even and odd, energy and power signals, deterministic and random signals, complex exponential and sinusoidal signals. Concepts of Impulse function, Unit step function, Signum function. Various operations on Signals.

Signal Analysis: Analogy between vectors and signals, Orthogonal signal space, Signal approximation using orthogonal functions, Mean square error, Closed or complete set of orthogonal functions, Orthogonality in complex functions

UNIT-II:

Fourier Series: Representation of Fourier series, Continuous time periodic signals, properties of Fourier series, Dirichlet's conditions, Trigonometric Fourier series and Exponential Fourier series, Complex Fourier spectrum, Gibb's Phenomenon

Fourier Transforms: Deriving Fourier transform from Fourier series, Fourier transform of arbitrary signals, Fourier transform of standard signals, Fourier transform of periodic signals, properties of Fourier transforms.

UNIT-III:

Laplace Transforms: Concept of region of convergence (ROC) for Laplace transforms. Properties of ROC. Relation between Laplace Transforms and Fourier transform of a signal. Introduction to Hilbert Transform.

Signal Transmission through Linear Systems: Classification of Continuous time and discrete time Systems, impulse response, Response of a linear system, Transfer function of a LTI system. Filter characteristics of linear systems. Distortion less transmission through a system, Signal bandwidth, system bandwidth, Ideal LPF, HPF and BPF characteristics, Causality and Paley -Wiener criterion for physical realization, relationship between bandwidth and rise time.

UNIT-IV:

Convolution of Signals: Concept of convolution in time domain and frequency domain, Graphical representation of convolution, Properties of Convolution

Correlation of Signals: Concepts of correlation, properties of correlation. Relation between convolution and correlation, Detection of periodic signals in the presence of noise by correlation, Extraction of signal from noise by filtering.

UNIT-V:

Sampling Theorem: Representation of continuous time signals by its samples - Sampling theorem – Reconstruction of a Signal from its samples, aliasing – discrete time processing of continuous time signals, sampling of band pass signals.

Z –Transforms: Basic principles of z-transform, region of convergence, properties of ROC, Properties of z-transform, Poles and Zeros. Inverse z-transform using Contour integration, Residue Theorem, Power Series expansion and Partial fraction expansion. Distinction among Fourier transform, Laplace Transform and Z - Transforms.

TEXT BOOKS:

1. Signals, Systems and Communications, B. P. Lathi, BS Publications, 2009
2. Signals and Systems, Alan V. Oppenheim, Alan S. Willsky and S. Hamid Nawab, 2nd Edition, PHI, 2000

REFERENCES:

1. Signals and Systems, Simon Haykin and Barry Van Veen, 2nd Edition, John Wiley, 1998
2. Signals, Systems and Transforms, C. L. Philips, J. M. Parr and Eve A. Riskin, 3rd Edition, PE, 2004
3. Fundamentals of Signals & Systems, Michael Roberts, 2nd Edition, Tata McGraw-Hill, 2010
4. Signals and Systems, H. P. Hsu, R. Ranjan, Scham's Outlines, Tate McGraw-Hill, 2006
5. Signals and Systems, A. Anand Kumar, 2nd Edition, PHI, 2012

ONLINE RESOURCES:

1. <https://nptel.ac.in/courses/108104100>
2. <https://nptel.ac.in/courses/108106163>

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. IV Semester

(25PC1CS205) DESIGN AND ANALYSIS OF ALGORITHMS

TEACHING SCHEME		
L	T/P	C
3	0	3

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE PRE-REQUISITES: Programming, Data Structures, Discrete Mathematics

COURSE OBJECTIVES:

- To reinforce algorithms analysis methods
- To analyse running time of an algorithm
- To understand different algorithm design strategies
- To apply algorithms design techniques to solve computational problems

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Explain fundamental algorithmic concepts such as asymptotic notations, recurrence relations, and performance measures of algorithms

CO-2: Apply algorithm design techniques—divide & conquer, greedy, dynamic programming, backtracking, and branch & bound—to solve computational problems

CO-3: Analyze the time complexity of algorithms and compare different solutions using asymptotic bounds

CO-4: Develop graph algorithms to solve real world problems and analyze time complexity

CO-5: Identify computational complexity classes P, NP, NP-Hard and NP complete problems and formulate solutions using standard approaches

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	-	-	1	-	-	-	-	-	1	2	1	2
CO-2	3	3	2	1	2	-	-	-	-	-	1	3	2	3
CO-3	3	3	-	2	1	-	-	-	-	-	2	2	1	3
CO-4	3	3	2	2	2	1	-	-	-	-	2	3	2	3
CO-5	2	3	1	2	1	-	-	-	-	-	3	2	1	3

UNIT-I:

Algorithm Analysis: Characteristics of Algorithm. Analysis of Algorithm, Best, Average and Worst-Case behavior, Asymptotic Notations– Order of Growth, Big oh, Omega, Theta notations and Little oh, Little omega notations, Analysis of Recursive Algorithms through Recurrence Relations, Substitution Method, Recursion Tree Method and Masters' Theorem.

UNIT-II:

Divide and Conquer: General method, Applications- Binary search, Quick sort, Merge sort, Strassen's matrix multiplication.

Greedy Method: General method, Applications- Job sequencing with deadlines, Fractional knapsack problem, Minimum cost spanning trees (Prims and Kruskal), Single source shortest path problem (Dijkstra), Huffman Codes.

UNIT-III:

Dynamic Programming: General method, Principle of optimality, Applications- Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem (Floyd Warshall), Travelling sales person problem, Reliability design. Longest Common Subsequence problem (LCS).

Pattern Matching Algorithms: Brute force, KMP.

UNIT-IV:

Backtracking Method: Applications- N-Queen problem, Sum of subsets problem, Graph coloring, Hamiltonian cycles.

Branch and Bound Method: Applications - Travelling sales person problem, 0/1 knapsack problem- LC Branch and Bound solution, FIFO Branch and Bound solution

UNIT-V:

Graph Algorithms: Depth First Search (DFS) and Breadth First Search (BFS), Topological sorting of DAGs, Connected, Bi Connected and strongly connected components, finding articulation points, Network Flow Algorithm (Ford Fulkerson Algorithm).

Tractable and Intractable Problems: P, NP, NP-complete and NP-hard. Cook's theorem, Reductions: 3-SAT, CLIQUE.

TEXT BOOKS:

1. Fundamental of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, 2nd Edition, Galgotia Publications
2. Introduction to Algorithms, Thomas H. Cormen, Charles E. Lieserson, Ronald L. Rivest and Clifford Stein, 4th Edition, MIT Press/McGraw-Hill

REFERENCES:

1. Algorithm Design, Jon Kleinberg and Eva Tardos, 1st Edition, Pearson
2. Algorithm Design: Foundations, Analysis, and Internet Examples, Michael T. Goodrich and Roberto Tamassia, 2nd Edition, Wiley
3. Introduction to the Design and Analysis of Algorithms, Anany Levitin, 3rd Edition, Pearson Publications
4. The Design and Analysis of Computer Algorithms, A. Aho, J. Hopcroft and J. Ullman

ONLINE RESOURCES:

1. <http://nptel.ac.in/courses/106101060/>
2. <https://www.cmi.ac.in/~madhavan/teaching.html>
3. <https://www.ics.uci.edu/~eppstein/161/960312.html>
4. <https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/>

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. IV Semester

(25PW4RA201) DESIGN THINKING

TEACHING SCHEME		
L	T/P	C
1	2	2

EVALUATION SCHEME		
CIE	SEE	TOTAL
40	60	100

COURSE OBJECTIVES:

- To instill a sense of significance towards applying creativity to product and service design
- To teach a systematic approach to identifying and defining a problem before brainstorming for a solution
- To inculcate core design principles and applied creativity to develop innovative strategies that better connect engineers and technologies with their end users
- To build a mindset leading to flow of creative ideas, validating those ideas and prioritizing the best ones among them
- To motivate students to apply design thinking while implementing projects focusing on local, regional or global societal problems

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Demonstrate the understanding of design principles from a technology perspective

CO-2: Validate problem statements through user empathisation with societal, cultural, global and Environmental consciousness

CO-3: Use specific and relevant ideation and brainstorming techniques to find innovative solutions

CO-4: Prototype a solution to address user challenges

CO-5: Investigate the cultural, emotional, environmental, technological and business factors relevant to developing new product or service design concept

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	-	-	2	-	-	2	-	-	3	3	1	2	-	-
CO-2	-	-	3	2	1	3	-	-	3	2	1	1	-	-
CO-3	-	-	2	-	-	2	-	-	2	2	3	1	-	-
CO-4	-	-	3	-	2	-	-	-	2	3	1	2	3	3
CO-5	-	-	-	3	-	3	1	-	1	-	1	1	1	1

UNIT-I:

Design Overview: Need of Design in Engineering, Various perspectives of design, Case study of Design thinking at Apple, IBM, IDEO and GE Healthcare with lessons learned on Good and Bad Design, Human-Centred Design (HCD) - Desirability, Feasibility, viability. Problem Identification - 17 SDGs as the core thematic problem areas.

Design Process: Introduction to various Design models, The Design Double Diamond model: Discover-Define-Develop-Deliver and Stanford's d. school design thinking model: Empathy-Define-Ideate-Prototype-Test.

UNIT-II:

User Empathy: Identifying Stakeholders and Defining User Personas, Data collection- User Interviews and conducting surveys, Empathy Tools: What/Why/How, Five Whys, Empathy Maps, AEIOU, Customer Journey Maps (CJM).

Need Analysis: Types of Users, making sense of observations and insights, Types of Needs-Primary, secondary, latent needs, Value Proposition to the Users, Identifying Addressable Needs and Touch points, Structuring Need Statements.

UNIT-III:

Ideation Process: Introduction to creativity and closed-world solutions, Idea generation techniques: Brainstorming, Mind Maps, SCAMPER, Systematic Inventive Thinking methods (Subtraction, Multiplication, Division, Task Unification, Attribute Dependency), 3-box thinking, and Box-3 ideation.

UNIT-IV:

Building Prototypes & Design for Sustainability: Building Conceptual model of product/service using various prototype methods. Concern for Environment and Sustainability in Design, Case Studies to understand Design for Environment (DFE).

UNIT-V:

Capstone Project (Interdisciplinary): Applying design thinking principles and methods for problem definition, ideation, prototyping, testing, refining and taking the solution to the users, using visual representation tools to indicate problem, User persona, needs, empathisation, ideas and prototype that leads to chosen solution, creating presentation.

TEXT BOOKS:

1. Change by Design, Tim Brown, Harper Business, 2012 (ISBN: 978-0062337382)
2. The Design of Everyday Things, Donald A. Norman, MIT Press, 2013 (ISBN: 978-0262525671)

REFERENCES:

1. The Art of Innovation, Tom Kelly, Jonathan Littman, HarperCollins Business, 2002
2. Design Thinking: Integrating Innovation, Customer Experience, and Brand Value, Thomas Lockwood, Allworth Press, 2009 (ISBN: 978-1581156683)
3. Design Thinking for Start-ups: A Handbook for Readers and Workbook for Practitioners, Jimmy Jain, Notion Press, 2018 (ISBN: 978-1642495034)

ONLINE RESOURCES:

1. https://eict.iitg.ac.in/course_img/76/052014859ef93ebe8c4dd4285fdd5dbe09296b.pdf
2. https://www.ideo.com/collections/design-thinking-courses?srltid=AfmBOoqcLf7_8mqdRd-KMqVCt5J6z6A6ABSbKQNJcvQKFzdq84E1dnjj

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. IV Semester

(25PC2RA203) ROBOT OPERATING SYSTEM LABORATORY

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE OBJECTIVES:

- To learn about the ROS 2 framework's subjects, services, actions, and parameters as well as its architecture, tools, and communication mechanisms
- To using Turtlesim and associated tools, create ROS 2 nodes in Python for real-time robot control, communication, and system integration
- To use URDF/Xacro to model and simulate mobile robots, and incorporate physics-based simulations into Ignition or Gazebo
- To configure Lidar, camera sensors, SLAM tools, and occupancy grid map production to use perception and mapping methodologies
- To create behaviour-tree-based task execution for mobile robots and use the Nav2 stack to implement autonomous navigation

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Set up ROS 2 workspaces, start nodes, and examine component communication using ROS CLI tools

CO-2: To manage and keep an eye on robots, create Python-based ROS 2 nodes for publishers, subscribers, services, and actions

CO-3: Create and model a differential-drive robot with URDF/Xacro, then set it up in Gazebo with the necessary sensors and plugins

CO-4: Create 2D maps with SLAM algorithms and use sensor data (camera, Lidar) for robot localization and visualization

CO-5: Use Behaviour Trees and Nav2 to implement and assess autonomous navigation behaviors in ROS 2 for multi-step robotic tasks

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	1	1	3	1	1	-	-	-	2	3	2	2
CO-2	3	3	2	2	3	1	1	-	-	-	2	3	3	2
CO-3	3	3	3	3	3	2	1	-	-	-	2	3	3	3
CO-4	3	3	3	3	3	2	2	2	2	2	3	3	3	3
CO-5	2	2	2	3	2	3	3	3	3	2	3	2	3	3

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

The student is expected to perform the following 12 compulsory experiments using ROS 2 (Hum- ble/Jazzy) and Python.

CYCLE I: THE ROS 2 FRAMEWORK & COMMUNICATION

EXP-1. Workspace Setup and" Turtlesim":

Install ROS 2 (Desktop Full). Setup the 'colcon' workspace and overlay. Launch the standard 'turtlesim' node. Use CLI tools ('ros2 topic list', 'ros2 node info', 'rqt graph') to inspect the active nodes and connections.

EXP-2. Publishers and Subscribers (Topics):

Write a Python Publisher node that sends velocity commands ('geometry_msgs/Twist') to the Turtlesim. Write a Subscriber node that reads the turtle's pose.

Concept: Asynchronous communication and QoS (Quality of Service) profiles.

EXP-3. Client-Server Communication (Services):

Create a Custom Service Interface ('.srv') (e.g., 'AddTwoInts' or 'ResetTurtle'). Implement a Service Server node to perform the computation and a Client node to request it.

Concept: Synchronous Request-Response pattern.

EXP-4. Long-Running Tasks (Actions):

Implement an Action Server ('.action') that moves a robot for a specific duration or distance, providing periodic "Feedback" to the client before sending the final "Result".

Concept: Non-blocking goals and feedback loops.

EXP-5. Launch Files and Parameters:

Create a Python Launch file ('.launch.py') to start multiple nodes (Publisher, Subscriber, Simulator) simultaneously. Use ROS Parameters to dynamically change node configuration (e.g., changing background color or movement speed) without recompiling.

EXP-6. Coordinate Transforms (TF2):

Write a static broadcaster to define a transform between a "World" frame and a "Robot" frame. Write a listener to compute the relative position of an object in the robot's frame. Visualize the TF tree in Rviz.

CYCLE II: SIMULATION & AUTONOMY

EXP-7. Robot Description (URDF/Xacro):

Create a URDF model of a simple 2-wheeled differential drive robot. Define links (chassis, wheels) and joints (continuous, fixed). Use 'xacro' to clean up the code. Visualize the model in Rviz.

EXP-8. Physics Simulation (Gazebo Classic/Ignition):

Add collision and inertial properties to the URDF. Add the "Differential Drive" plugin. Create a custom '.world' file with obstacles and spawn the robot into this Gazebo environment.

EXP-9. Sensor Simulation (Lidar & Camera):

Attach virtual sensors (LaserScan and RGB Camera) to the robot URDF using Gazebo plugins. Visualize the point cloud and camera feed in Rviz while the robot moves in Gazebo.

EXP-10. Mapping (SLAM Toolbox / Cartographer):

Launch the 'slam toolbox' or 'cartographer' node. Teleoperate the robot (using keyboard) around the Gazebo world to generate a 2D Occupancy Grid Map. Save the map using 'map server'.

EXP-11. Autonomous Navigation (Nav2 Stack):

Configure the Nav2 stack (AMCL for localization, Costmaps, and Planners). Load the saved map. Send a "2D Nav Goal" via Rviz and observe the robot planning a path and avoiding obstacles to reach the target.

EXP-12. Behavior Trees (Nav2 Commander API):

Write a Python script using the 'BasicNavigator' API to sequence complex tasks (e.g., "Go to Room A, wait 5 seconds, then go to Room B"). Monitor the Behavior Tree execution.

TEXT BOOK:

1. A Concise Introduction to Robot Programming with ROS2, Francisco Martín Rico, CRC Press

REFERENCES:

1. Documentation, Official ROS 2 Documentation (docs.ros.org), Open Source Robotics Foundation.
2. Automatic Addison / The Construct, Online ROS 2 Navigation Tutorials

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. IV Semester

(25PC2AM201) MACHINE LEARNING LABORATORY

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE OBJECTIVES:

- To gain an overview of various supervised machine learning techniques
- To understand and evaluate various unsupervised machine learning techniques
- To implement and demonstrate various libraries used in machine learning
- To understand the process of training, building, and evaluating machine learning models
- To apply machine learning techniques to real-world case studies

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Illustrate and implement supervised machine learning algorithms

CO-2: Develop and apply unsupervised machine learning algorithms, such as clustering and dimensionality reduction, to analyse patterns in data

CO-3: Explore the various libraries used in Machine Learning

CO-4: Build, train, and evaluate machine learning models using appropriate performance metrics, cross-validation, and model validation techniques

CO-5: Perform experiments on real-world datasets using machine learning techniques

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	2	2	3	1	-	1	1	1	-	3	2	2
CO-2	3	3	2	3	3	1	1	1	1	1	1	3	3	2
CO-3	2	2	1	1	3	-	-	-	1	1	-	3	2	2
CO-4	3	3	3	3	3	1	-	1	1	1	1	3	3	3
CO-5	3	3	2	2	3	2	1	1	2	2	1	3	3	3

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

WEEK 1: Write a python program to compute Central Tendency Measures: Mean, Median, Mode, Measure of Dispersion: Variance, Standard Deviation

WEEK 2: Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy

WEEK 3: Study of Python Libraries for ML application such as Pandas and Matplotlib

WEEK 4: Write a Python program to implement Simple Linear Regression

WEEK 5: Implementation of Multiple Linear Regression for House Price Prediction using sklearn

WEEK 6: Implementation of Decision tree using sklearn and its parameter tuning

WEEK 7: Implementation of KNN using sklearn

WEEK 8: Implementation of Logistic Regression using sklearn

WEEK 9: Implementation of K-Means Clustering

WEEK 10: Performance analysis of Classification Algorithms on a specific dataset (Mini Project)

WEEK 11: Implementation of real-world case study - Lower Back Pain Symptoms Clustering/ Personalized Cancer Diagnosis.

WEEK 12: Lab Internal Exam

TEXT BOOK:

1. Machine Learning, Tom M. Mitchell, McGraw-Hill

REFERENCE:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

B. Tech. IV Semester

(25PC2RA204) KINEMATICS AND DYNAMICS OF ROBOTS LABORATORY

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE OBJECTIVES:

- To develop spatial intuition through the implementation of rotation matrices and homogeneous transformations
- To master the Denavit-Hartenberg (D-H) convention for modeling serial robotic manipulators
- To solve inverse kinematics problems using both analytical and geometric techniques
- To analyze differential motion, velocities, and singular configurations using the Jacobian matrix
- To execute industrial robotic operations including jogging, simulation-based verification, and vacuum based pick and place

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Utilize homogeneous transformation matrices to represent the position and orientation of rigid bodies in 3D space

CO-2: Construct forward kinematic models for multi-DOF manipulators using the D-H parameter convention

CO-3: Validate kinematic models by comparing theoretical end-effector coordinates with experimental measurements

CO-4: Compute the manipulator Jacobian to relate joint velocities to task-space motion and identify singularities

CO-5: Perform industrial-grade robotic operations including jogging, vacuum suction, and Roboguide-integrated pick and place

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	2	1	2	2	-	-	-	1	1	1	3	1	1
CO-2	3	2	2	2	3	-	-	-	1	1	1	3	1	2
CO-3	3	3	2	3	3	-	-	-	1	1	1	3	2	2
CO-4	2	3	3	2	3	-	-	-	1	1	1	2	2	3
CO-5	2	3	3	2	3	-	-	-	1	1	1	1	3	3

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

The student is expected to perform the following experiments.

PART A: SPATIAL FOUNDATIONS AND KINEMATICS

EXP-1. Spatial Descriptions and Frames:

Implement functions to describe a frame relative to a fixed reference frame and perform frame rotations.

EXP-2. Homogeneous Transformation Matrices (HTM)

Write a program to calculate the final pose of a rigid body after a sequence of translations and rotations.

EXP-3. D-H Parameter Assignment

Construct the D-H parameter table based on the distal coordinate frame convention.

EXP-4. Forward Kinematics Implementation

Develop a script to compute the end-effector pose by implementing the product of successive link transformation matrices derived from the D-H table.

EXP-5. Workspace Visualization

Identify the boundaries of the primary and secondary workspaces of a 6-DOF robot.

EXP-6. Analytical Inverse Kinematics

Implement an algebraic or geometric solution to determine the joint variables required to reach a specific target position (x, y, z) in task space.

EXP-7. Jacobian Matrix Computation

Develop a function to compute the Manipulator Jacobian for a serial chain.

EXP-8. Singularity and Dexterity Analysis

Analyze the determinant of the Jacobian across the robot's workspace.

EXP-9. Static Force and Torque Analysis

Utilize the Jacobian Transpose to map an external force applied at the end-effector back to the required steady-state joint torques.

PART B: INDUSTRIAL APPLICATIONS AND OPERATIONS

EXP-10. Perform jogging operation using 6 axis industrial robot.

EXP-11. Simulation of pick and place operation using Roboguide software and verify it using 6 axis industrial robot.

EXP-12. To perform pick and place operation using vacuum suction.

TEXT BOOK:

1. Introduction to Robotics: Analysis, Control, Applications, Saeed B. Niku, 3rd Edition, Wiley

REFERENCE:

1. Introduction to Robotics, S.K. Saha, McGraw Hill Education

B. Tech. IV Semester

(25SD5CS202) WEB APPLICATION DEVELOPMENT

TEACHING SCHEME		
L	T/P	C
0	2	1

EVALUATION SCHEME					
D-D	PE	LR	CP	SEE	TOTAL
10	10	10	10	60	100

COURSE PRE-REQUISITES: Object-Oriented Programming through Java, HTML and CSS Basics, Fundamental understanding of web technologies

COURSE OBJECTIVES:

- To build responsive web pages using HTML5, CSS3, Bootstrap, and modern layout techniques (Flexbox, Grid)
- To implement client-side interactivity and state management using ES6+ JavaScript
- To develop REST APIs and backend services using Node.js and Express.js
- To design and manage NoSQL databases using MongoDB and implement database operations from the backend and build single-page applications (SPAs) with React, including routing, state management, and component lifecycle
- To implement authentication, authorization, and security best practices in full-stack applications and deploy and host full-stack web applications on cloud platforms

COURSE OUTCOMES: After completion of the course, the student should be able to
CO-1: Build responsive, accessible web pages using HTML5, CSS3, Bootstrap, and modern CSS layout techniques and Implement client-side interactivity, form validation, and state management using ES6+ JavaScript features

CO-2: Design and develop REST APIs and server-side logic using Node.js and Express.js frameworks and Design NoSQL database schemas, implement CRUD operations, and manage data persistence using MongoDB

CO-3: Build single-page applications with component-based architecture, routing, and state management using React

CO-4: Implement user authentication, authorization, and security mechanisms (JWT, hashing, CORS) in web applications

CO-5: Deploy, test, and host full-stack applications on cloud platforms with CI/CD pipelines

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	3	3	3	1	3	-	-	1	2	-	2	3	2	1
CO-2	3	3	3	2	3	-	-	1	1	-	2	3	3	2
CO-3	3	3	3	2	3	-	-	2	2	-	2	3	3	2
CO-4	2	3	3	2	3	1	3	1	1	-	2	3	2	2
CO-5	2	3	3	2	3	1	2	2	2	2	3	3	3	3

LIST OF PROGRAMS / EXPERIMENTS / EXERCISES:

WEEK 1: Responsive Multi-Page Website

Build a responsive website (3-4 pages) for your chosen project domain using HTML5, CSS3, Bootstrap, and modern layout techniques. Pages should include: landing page, content pages, contact/form page. Implement CSS Grid, Flexbox, and responsive design principles.

WEEK 2: Client-Side Form Validation & Interactivity

Add JavaScript interactivity to Exercise 1. Implement client-side form validation, dynamic DOM manipulation, event handling, and local storage. Create interactive UI elements (modal dialogs, collapsible sections, animated transitions).

WEEK 3: Learn ES6+ Features & Fetch API

Explore ES6+ features (arrow functions, destructuring, spread operator, async/await). Create a practice application that fetches data from a public API (OpenWeatherMap, JSONPlaceholder, etc.) and displays it dynamically on the webpage. Implement error handling and loading states.

WEEK 4: RESTful API Development

Develop a REST API for your project using Node.js and Express.js. Create endpoints for key entities in your domain (e. g., products, users, posts, students). Implement GET, POST, PUT, DELETE operations. Use Postman to test all endpoints. Document API with Swagger/OpenAPI

WEEK 5: Error Handling, Validation & Middleware

Enhance your API from Exercise 4 with:

- Request validation middleware
- Error handling middleware
- Logging middleware
- Authentication middleware (basic)

Implement comprehensive error responses and HTTP status codes.

WEEK 6: MongoDB Schema Design & CRUD Operations

Design a MongoDB schema for your project entities. Implement CRUD operations using Mongoose. Create seed scripts to populate sample data. Test all operations. Implement data validation and indexing for performance.

WEEK 7: Integrate Backend API with Express & MongoDB

Connect your Express API (Exercise 4-5) with MongoDB database (Exercise 6). Replace mock data with actual database operations. Implement data persistence, retrieval, and updates through the API.

WEEK 8: React Component Architecture & Hooks

Create a React application with component-based architecture. Implement functional components, hooks (useState, useEffect), props, and component composition. Build interactive UI components for your project (forms, cards, lists).

WEEK 9: React Routing & State Management

Implement React Router for multi-page navigation in your React app. Add route parameters and nested routes. Implement state management using Context API or Redux for shared state across components.

WEEK 10: Integrate React Frontend with Backend APIs

Connect your React SPA (single Page Application) to your backend API. Implement data fetching, form submissions, and CRUD operations through API calls. Add loading states and error handling. Implement pagination and filtering where applicable.

WEEK 11: JWT Authentication & Authorization

Implement user authentication using JWT. Create login/signup endpoints in backend. Implement JWT token generation and validation. Add protected routes in React that require authentication. Implement role-based access control.

WEEK 12: Security Best Practices & Input Validation

Implement security measures:

- Password hashing (bcrypt)
- CORS configuration
- Input validation and sanitization
- Prevention of common vulnerabilities (XSS, CSRF, SQL injection)
- Secure environment variable management

WEEK 13: Deployment & DevOps

Deploy your complete full-stack application to a cloud platform (Heroku, Vercel, Firebase, AWS). Implement Docker containerization. Set up CI/CD pipeline (GitHub Actions). Configure environment-specific configurations. Implement monitoring and logging.

WEEK 14: Capstone Project Submission & Presentation

Present your complete full-stack application demonstrating:

- Responsive frontend with intuitive UI/UX
- Functional backend APIs with proper error handling
- Secure authentication and authorization
- Database operations and data persistence
- Deployment and live hosting
- Code quality and documentation

CAPSTONE PROJECT THEMES (Students Choose One)

1. E-commerce Platform: Product catalog, shopping cart, checkout, order management, user authentication
2. Blog/Content Management System: Article creation/editing, commenting, tagging, search functionality
3. Social Network: User profiles, posts/feeds

TEXT BOOKS:

1. Full Stack Web Development with React, Node.js, Express, and MongoDB, Vasan Subramanian, Apress, 2023

2. Learning Full-Stack JavaScript Development: MongoDB, Node and React, Daniel Shiffman, O'Reilly Media, 2020

REFERENCE:

1. The MERN Stack Tutorial: Building Web Applications with MongoDB, Express, React, and Node, Jack Herrington, 2021

B. Tech. IV Semester

(25VA7EN201) GENDER SENSITIZATION

TEACHING SCHEME		
L	T/P	C
1	0	1

EVALUATION SCHEME				
SE	CA	ELA	SEE	TOTAL
30	05	05	60	100

COURSE DESCRIPTION:

This course offers an introduction to Gender Studies, an interdisciplinary field that asks critical questions about the meanings of sex and gender in society. The primary goal of this course is to familiarize students with key issues, questions, and debates in Gender Studies, both historical and contemporary. It draws on multiple disciplines – such as literature, history, economics, psychology, sociology, philosophy, political science, anthropology, and media studies – to examine cultural assumptions about sex, gender, and sexuality. This course integrates analysis of current events through student presentations, aiming to increase awareness of contemporary and historical experiences of women, and of the multiple ways that sex and gender interact with race, class, caste, nationality, and other social identities. This course also seeks to build an understanding and initiate and strengthen programs combating gender-based violence and discrimination. The course also features several exercises and reflective activities designed to examine the concepts of gender, gender-based violence, sexuality, and rights. It will further explore the impact of gender-based violence on education, health and development

ACTIVITIES:

Classes will consist of a combination of activities: dialogue-based lectures, discussions, collaborative learning activities, group work, and in-class assignments.

COURSE OBJECTIVES:

- To sensitize students on issues of gender in contemporary India
- To provide a critical perspective on the socialization of men and women
- To expose the students to debates on the politics and economics of work
- To enable students to reflect critically on gender violence
- To expose students to more egalitarian interactions between men and women

COURSE OUTCOMES: After completion of the course, the student should be able to

CO-1: Understand important issues related to gender in contemporary India

CO-2: Attain a finer grasp of how gender discrimination works in our society and how to counter it

CO-3: Acquire insight into the gendered division of labour and its relation to politics and economics

CO-4: Respond to put an end to gender violence

CO-5: Equipped to work with the other gender, treating them as equals

COURSE ARTICULATION MATRIX:

(Correlation of Course Outcomes with Program Outcomes and Program Specific Outcomes using mapping levels 1 = Slight, 2 = Moderate and 3 = Substantial)

CO	PROGRAM OUTCOMES (PO)											PROGRAM SPECIFIC OUTCOMES (PSO)		
	PO-1	PO-2	PO-3	PO-4	PO-5	PO-6	PO-7	PO-8	PO-9	PO-10	PO-11	PSO-1	PSO-2	PSO-3
CO-1	-	-	-	-	-	2	3	2	-	-	3	-	1	1
CO-2	-	-	-	-	-	2	3	2	-	-	3	-	1	1
CO-3	-	-	-	-	-	2	3	2	-	2	3	-	1	1
CO-4	-	-	-	-	-	2	3	2	-	-	3	-	1	1
CO-5	-	-	-	-	-	2	3	2	-	3	3	-	1	1

MODULE 1: Understanding Gender – Concepts and Myths

Core Focus: Gender vs Sex; Gender as a social construct

Activities:

- Think–Pair–Share: Is gender biological or social?
- Writing: How is gender socially constructed in Indian society?

Roots of Gender Inequality in Society

Core Focus: Patriarchy, culture, socialization

Activities:

- Brainstorming everyday examples of inequality
- Writing: Explain two root causes of gender inequality.

Gender Roles and Stereotypes

Core Focus: Social expectations and bias

Activities:

- Role-reversal discussion
- Writing: Impact of gender stereotypes on youth.

MODULE 2: Gender and Family Division of Labour

Core Focus: Unpaid work and care economy

Activities:

- Household work mapping
- Writing: Do we take women's labour at home for granted?

Gender Inequality in Education

Core Focus: Girl child education and dropout issues

Activities:

- Case discussion
- Writing: Problems leading to girls' school dropout in India.

Gender Inequality in the Workplace

Core Focus: Wage gap and glass ceiling

Activities:

- Debate: Does gender help or hinder women at work?
- Writing: Is there a glass ceiling for women?

Gender Equality and Economic Development

Core Focus: Women's participation and growth

Activities:

- Group discussion on women and GDP
- Writing: What does gender inequality cost an economy?

MODULE 3: Gender-Based Violence

Core Focus: Forms and causes

Activities:

- Scenario analysis
- Writing: Types of gender-based violence.

Sexual Harassment and MeToo Movement

Core Focus: Workplace safety and awareness

Activities:

- Case study discussion
- Writing: Causes and effects of the MeToo Movement.

MODULE 4: Gender Laws in India

Core Focus: Legal protection and enforcement

Activities:

- Law-awareness quiz
- Writing: Is Indian law leaning favourably towards women?

Gender and Media

Core Focus: Television, advertisements, and stereotypes

Activities:

- Media clip analysis
- Writing: How media reinforces gender roles.

Gender Representation in Indian Cinema

Core Focus: Objectification and portrayal

Activities:

- Movie scene critique
- Writing: Representation of men and women in Indian films.

MODULE 5: Women in Politics and Leadership

Core Focus: Power and representation

Activities:

- Discussion on women leaders in India
- Writing: What does gender equality in politics mean?

Gender, Self-Esteem, and Identity

Core Focus: Psychological dimensions

Activities:

- Self-reflection exercise
- Writing: Gender and low self-esteem among youth.

Gender Spectrum and LGBTQ+ Issues

Core Focus: Inclusion and rights

Activities:

- Myth vs Fact activity
- Writing: Is gender a social and cultural construct?

Gender Equality – Achievements and Way Forward

Core Focus: India's progress and future goals

Activities:

- Group presentation: India and Gender Equality
- Writing (Mid Exam): Is true gender equality achievable in India?

TEXT BOOK:

1. Towards a World of Equals: A Bilingual Textbook on Gender, A. Suneetha, Uma Bhrugubanda, Duggirala Vasanta, Rama Melkote, Vasudha Nagaraj, Asma Rasheed, Gogu Shyamala, Deepa Sreenivas and Susie Tharu, Telugu Akademi, Telangana, Government, 2015

REFERENCES:

1. More than One Million Women are Missing, Sen, Amartya, New York Review of Books 37.20 (20 December 1990), Print 'We Were Making History' Life Stories of Women in the Telangana People's Struggle, New Delhi: Kali for Women, 1989
2. By the Numbers: Where Indian Women Work Women's Studies Journal (14 November 2012), Tripti Lahiri, Available online at: <http://blogs.wsj.com/India/real-time/2012/11/14/by-the-numbers-where-india-women-work/>>.
3. I Fought For My Life ...and Won, Abdulali Sohaila, Available online at: <http://www.thealternative.in/lifestyle/i-fought-for-my-lifeand-won-sohaila-abdulali>
4. The Violence of Development: The Politics of Identity, Gender and Social Inequalities in India, K. Kapadia, London: Zed Books, 2002
5. Just Development: Beyond Adjustment with a Human Face, T. Banuri and M. Mahmood, Karachi: Oxford University Press, 1997