Scheme of Evaluation
B. Tech. VIII Semester *(CSD)*

(for batch admitted in academic session 2022-23)

S. No.	Subject Code	Category Code	Subject Name	Maximum Marks Allotted									Total Marks	Contact Hours per week			Total Credits	Mode of Teaching	Mode of Exam.	Duration of Exam.
				Theory Slot				Practical Slot			MOOCs									
				End Term Evaluation		Continuous Evaluation			Continuous Evaluation					L	T	P				
				End Sem. Exam	$Proficiency in subject /course	Mid Sem. Exam.	Quiz/ Assignment	End Sem. Exam.	Lab Work & Sessional	Skill Based Mini Project	Assignment	Exam								
1.	DE	DE	Departmental Elective*(DE-5)	-	-	-	-	-		-	25	75	100	3	-	-	3	Online	MCQ	3 Hrs
2.	OC	OC	Open Category (OC-3)	-	-	-	-	-		-	25	75	100	3		-	3	Online	MCQ	3 Hrs
3.	2290801	DLC	Internship/Project Project/Innovation & start-up	-	-	-	-	250	150	-	-	-	400	-	-	18	9	Blended	SO	
4.	2290802	-	Professional Development#	-	-	-	-	50	-	-	-	-	50	-	-	4	2			
Total				-	-	-	-	300	150	-	50	150	650	06	-	22	17	-		
	Additional Courses for obtaining Honours or minor Specialization by desirous students			Permitted to opt for maximum two additional courses for the award of Honours or Minor specialization																

$Proficiency in course/subject – includes the weightage towards ability/ skill/ competency /knowledge level /expertise attained etc. in that particular course/subject

Natural Sciences & Skills: Engineering Physics / Engineering Chemistry / Environmental Engineering / Language

Credits of Natural Sciences & Skills will be added in the VI Semester

MCQ: Multiple Choice Question **AO:** Assignment + Oral **PP:** Pen Paper **SO:** Submission + Oral

Mode of Teaching						Mode of Examination					Total Credits
Theory				Lab	NEC/professional development	Theory			Lab	SIP/SLP/NEC/ professional development	
Offline	Online	Blended		Offline	Interactive	PP	AO	MCQ	SO	SO	
		Offline	Online								
-	06			09	02			06	09	02	17
-	35			53	12			35	53	12	Credits %

DE-5*		
S. No.	Subject Code	Subject Name
1.	2290811	GPU Architectures and Programming
2.	2290812	Introduction To Soft Computing
3.	2290813	Selected Topics in Algorithms

OC-3*		
S. No.	Subject Code	Subject Name
1.		Affective Computing
2.		Natural Language Processing
3		Compiler Design

List of courses to be opted for Honors or Minor specialization in VIII Semester

Minor Specialization* *(to be opted by students of Other Department)*
Advanced Computer Networks
Data Mining
Foundations of Deep Learning: Concepts and Applications

Honors* ***(to be opted by students of Parent Department)***	
Track	**Course**
Information Security	Cryptography and Network Security
	Foundations of Cyber Physical Systems
IoT	**VLSI Physical Design**
	Introduction to Information Retrieval
High Performance Computing	Edge Computing
	Embedded System Design with ARM

*** Course run through SWAYAM/NPTEL/MOOC Learning Based Platform**

Scheme of Evaluation
B. Tech. VI Semester *(CSD)*

(for batch admitted in academic session 2023-24)

S. No.	Subject Code	Category Code	Subject Name	Maximum Marks Allotted									Total Marks	Contact Hours per week			Total Credits	Mode of Teaching	Mode of Exam.	Duration of Exam.
				Theory Slot				Practical Slot			MOOCs									
				End Term Evaluation		Continuous Evaluation		End Sem. Exam.	Continuous Evaluation		Assignment	Exam		L	T	P				
				End Sem. Exam	$Proficiency in subject /course	Mid Sem. Exam.	Quiz/ Assignment		Lab Work & Sessional	Skill Based Mini Project										
1.	3290601	DC	IOT System Design	50	10	20	20	40	30	30	-	-	200	3	-	2	4	Blended	PP	2 Hrs
2.	3290602	DC	Artificial Intelligence & Machine Learning	50	10	20	20	40	30	30	-	-	200	3	-	2	4	Blended	PP	2 Hrs
3.	DE	DE	Departmental Elective* (DE-1)	-	-	-	-	-	-	-	25	75	100	3	-	-	3	Blended	MCQ	3 Hrs
4.	OC	OC	Open Category (OC-1)	50	10	20	20	-	-	-	-	-	100	3	-	-	3	Blended	PP	2 Hrs
5.	3290603	DLC	Minor Project-II	-	-	-	-	40	60	-	-	-	100	-	-	6	3	Offline	SO	
6.	200XXX	CLC	Novel Engaging Course (Informal Learning)	-	-	-	-	50	-	-	-	-	50	-	-	2	1	SO	SO	-
8.		NSS	Natural Science & Skills	200	40	80	80	120	40	40	-	-	600	1	-	2	2*	-	-	-
Total				**350**	**70**	**140**	**140**	**290**	**160**	**100**	**25**	**75**	**1350**	**13**	**-**	**14**	**20**			
10.	1000007$$	MAC	Intellectual Property Rights (IPR)	50	10	20	20	-	-				100	2	-	-	Grade	Online	MCQ	1.5 Hrs

$Proficiency in course/subject – includes the weightage towards ability/ skill/ competency /knowledge level /expertise attained etc. in that particular course/subject

Natural Sciences & Skills: Engineering Physics / Engineering Chemistry / Environmental Engineering / Language

Credits of Natural Sciences & Skills will be added in the VI Semester

MCQ: Multiple Choice Question **AO:** Assignment + Oral **PP:** Pen Paper **SO:** Submission + Oral

Mode of Teaching						Mode of Examination					Total Credits
Theory				Lab	NEC	Theory			Lab	SIP/SLP/NEC	
Offline	Online	Blended		Offline	Interactive	PP	AO	MCQ	SO	SO	
		Offline	Online								
	3	9+2*	-	5	1	11	-	03+2*	03	01	20
	15	45+10*		25	5	55		15+10*	15	5	Credits %

DE-1*		
S. No.	Subject Code	Subject Name
1.	3290611	Reinforcement Learning
2.	3290612	Digital Design with Verilog

OC-1		
S. No.	Subject Code	Subject Name
1.		Database Management System
2.		Operating Systems

List of courses to be opted for Honors or Minor specialization in VI Semester

Minor Specialization* *(to be opted by students of Other Department)*
Programming in Modern C++
Introduction To Internet of Things

Honors* ***(to be opted by students of Parent Department)***	
Track1	**Course**
Information Security	Foundations of Cryptography
	Secure Computation: Part I
IoT	Introduction To Industry 4.0 And Industrial Internet Of Things
	Wireless Ad Hoc and Sensor Networks
High Performance Computing	Parallel Computer Architecture
	Advanced Computer Architecture

***Course run through SWAYAM/NPTEL/MOOC Learning Based Platform**

Department of Computer Science and Engineering

IoT System Design

3290601

COURSE OBJECTIVES

.

- To introduce the fundamentals of internet of Things.
- To understand the technologies, system architecture, and communication architecture that propelled the growth of IoT Systems.
- To develop IoT infrastructure for popular applications.

Unit-1 **Introduction to Internet of Things (IoT):** Vision, Definition, IoT architecture: Layers and protocols, technology behind IoT, Sources of the IoT, M2M Communication, IoT Examples. Sensing, Actuation. IoT communication models: Device-to-device, device-to-cloud, device-to-gateway, Challenges and opportunities in IoT

Unit-2 **Hardware for IoT:** Sensors, Digital sensors, actuators, radio frequency identification (RFID) technology, microcontrollers, and single-board computers, **Embedded Platforms for IoT:** Embedded computing basics, Overview of IOT supported Hardware platforms such as Raspberry pi, Jetson nano, Beagle Bone, and Intel Galileo boards. Interfacing and programming using GPIO; introduction to Python-based GPIO programming for minicomputers.

Unit-3 **IoT Protocols and Arduino Programming:** Wi-Fi, Bluetooth, CoAP, LPWAN protocol. Sensor Networks: Sensor deployment & Node discovery, **Introduction to Arduino Programming:** Arduino Platform Boards Anatomy, Arduino IDE, coding, using emulator, using libraries, additions in Arduino, programming the Arduino for IoT, Integration of Sensors and Actuators with Arduino.

Unit-4: **IoT Data Management and Analytics**: Data generation and collection in IoT systems, IoT data storage, Cloud-based and local storage, Data preprocessing and analysis, Basics of analytics for IoT data, IoT dashboards, Visualization and interpretation of data, Role of machine learning and AI in IoT

Unit-5: **Challenges in IoT Design challenges:** IoT applications: Smart homes, smart cities, healthcare, agriculture, Smart Metering, City Automation, Automotive Applications, home automation, smart cards, and industry, IoT and automation: Role in industrial IoT (IIoT), Emerging technologies: AIoT, Blockchain for IoT, and 5G integration IoT, Development Challenges, Security Challenges.

REFERENCE BOOKS: -

1. Olivier Hersent, David Boswarthick, Omar Elloumi "The Internet of Things key applications and protocols", willey.
2. Jeeva Jose, Internet of Things, Khanna Publishing House.
3. Michael Miller "The Internet of Things" by Pearson.
4. Raj Kamal "INTERNET OF THINGS", McGraw-Hill, 1ST Edition, 2016.
5. Arshdeep Bahga, Vijay Madisetti "Internet of Things (A hands on approach)" 1ST edition, VPI publications,2014.
6. Adrian McEwen, Hakin Cassimally "Designing the Internet of Things" Wiley India.

COURSE OUTCOMES:

After completion of this course, the students would be able to:

1. Understand the basic concepts, principles and challenges in IoT.
2. Describe the functioning of hardware devices and sensors used for IoT.
3. Analyze network communication aspects and protocols used in IoT.
4. Apply IoT for developing real life applications using Arduino programming.
5. Develop IoT infrastructure for popular applications

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	2	3	2	1	1	1	1	1	1	3	3	3
CO2	2	2	2	2	2	1	1	1	1	1	1	3	3	3
CO3	3	3	3	3	2	2	1	1	1	1	1	3	3	3
CO4	3	3	2	3	2	1	2	1	1	1	1	3	3	3
CO5	3	3	3	3	2	1	1	1	1	1	2	3	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

IoT System Design

2290601(OLD)

COURSE OBJECTIVES

.

- To introduce the fundamentals of internet of Things.
- To understand the technologies, system architecture, and communication architecture that propelled the growth of IoT Systems.
- To develop IoT infrastructure for popular applications.

Unit-1 **Introduction to Internet of Things (IoT):** Vision, Definition, IoT architecture: Layers and protocols, technology behind IoT, Sources of the IoT, M2M Communication, IoT Examples. Sensing, Actuation. IoT communication models: Device-to-device, device-to-cloud, device-to-gateway, Challenges and opportunities in IoT

Unit-2 **Hardware for IoT:** Sensors, Digital sensors, actuators, radio frequency identification (RFID) technology, microcontrollers, and single-board computers, **Embedded Platforms for IoT:** Embedded computing basics, Overview of IOT supported Hardware platforms such as Arduino, Raspberry pi, Jetson nano Beagle Bone, and Intel Galileo boards.

Unit-3 **IoT Protocols and Arduino Programming:** Wi-Fi, Bluetooth, CoAP, LPWAN protocol. Sensor Networks: Sensor deployment & Node discovery, **Introduction to Arduino Programming:** Arduino Platform Boards Anatomy, Arduino IDE, coding, using emulator, using libraries, additions in Arduino, programming the Arduino for IoT, Integration of Sensors and Actuators with Arduino.

Unit-4: **IoT Data Management and Analytics**: Data generation and collection in IoT systems, IoT data storage, Cloud-based and local storage, Data preprocessing and analysis, Basics of analytics for IoT data, IoT dashboards, Visualization and interpretation of data, Role of machine learning and AI in IoT

Unit-5: **Challenges in IoT Design challenges:** IoT applications: Smart homes, smart cities, healthcare, agriculture, Smart Metering, City Automation, Automotive Applications, home automation, smart cards, and industry, IoT and automation: Role in industrial IoT (IIoT), Emerging technologies: AIoT, Blockchain for IoT, and 5G integration IoT, Development Challenges, Security Challenges.

REFERENCE BOOKS: -

7. Olivier Hersent, David Boswarthick, Omar Elloumi "The Internet of Things key applications and protocols", willey.
8. Jeeva Jose, Internet of Things, Khanna Publishing House.
9. Michael Miller "The Internet of Things" by Pearson.
10. Raj Kamal "INTERNET OF THINGS", McGraw-Hill, 1ST Edition, 2016.
11. Arshdeep Bahga, Vijay Madisetti "Internet of Things (A hands on approach)" 1ST edition, VPI publications,2014.
12. Adrian McEwen, Hakin Cassimally "Designing the Internet of Things" Wiley India.

COURSE OUTCOMES:

After completion of this course, the students would be able to:

6. Understand the basic concepts, principles and challenges in IoT.
7. Describe the functioning of hardware devices and sensors used for IoT.
8. Analyze network communication aspects and protocols used in IoT.
9. Apply IoT for developing real life applications using Arduino programming.
10. Develop IoT infrastructure for popular applications

Department of Computer Science and Engineering

Artificial Intelligence & Machine Learning

3290602

COURSE OBJECTIVES:

1. To provide the fundamental knowledge of Artificial Intelligence and Machine Learning.
2. To understand the basic areas of AI & ML including problem solving, knowledge representation, reasoning, models, Loss functions.
3. To apply machine learning and optimization techniques to make predictions

Unit – I:

Introducing Artificial Intelligence: Introduction to AI - Intelligent Agents, Problem-Solving Agents, Searching for Solutions - Breadth-first search, Depth-first search, Hill-climbing search, simulated annealing search, Local Search in Continuous Spaces. Games - Optimal Decisions in Games, Water-Jug problem, Travelling salesman problem, Alpha–Beta Pruning.

Unit – II:

Knowledge Representation in AI: Need for Knowledge representation. Types of Knowledge. Knowledge and Intelligence, AI Knowledge cycle. Various approaches to Knowledge Representation. Requirements of Knowledge Representation System. Intelligent agent.

Constraint Satisfaction: Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Knowledge-Based Agents, Logic Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses.

Unit – III:

Introduction to Machine Learning: Learning, Traditional vs Machine Learning, Types of ML, Classification and Regression model, Challenges faced by ML, Steps of developing an ML model, Bias and Variance, Regularization, Testing and validating, K cross validation, Hyper parameter tuning, Model Selection.

Model optimization and Evaluation: Confusion matrix, Recall, accuracy, precision, Model optimization, Cost/Loss Function, Derivative of cost function and non-derivative cost function, Gradient descent.

Unit – IV:

Supervised Machine Learning Algorithm with python: Supervised Machine Learning Algorithms, k-Nearest Neighbors, Linear Regression, Logistic Regression, Support Vector Machine, Decision Trees,Ensemble learner, Random Forests.

Unit –V:

Unsupervised Machine Learning with python: The Curse of Dimensionality, Projection, Clustering, K-Means, Limits of K-Means, Clustering for Image Segmentation, Clustering for Preprocessing, Clustering for Semi-Supervised. Explainable AI (XAI) or AI, Ethical AI, AutoML, Capstone Project and Case Studies: Build an intelligent agent or a knowledge-based chatbot

---

RECOMMENDED BOOKS:

1. Artificial Intelligence: A Modern Approach by Stuart J. Russell and Peter Norvig, Prentice Hall.
2. Artificial Intelligence: Elaine Rich, Kevin Knight, Mc-Graw Hill.
3. Pattern Recognition and Machine Learning, Christopher M. Bishop
4. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow by Aurélien Géron
5. Introduction to Machine Learning with Python by Andreas C. Müller & O'reilly

---

COURSE OUTCOMES: After completing the course, the student will be able to:

CO1: Define basic concepts of Artificial Intelligence & Machine Learning.

CO2: Illustrate various techniques for knowledge representation and processing.

CO3: Apply various model optimization and tuning approaches.

CO4:Develop a model using supervised/unsupervised machine learning algorithms for classification/prediction/clustering

CO5: Evaluate performance of machine learning algorithms on various data sets of a domain.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	3	2	1	1	-	1	1	2	3	3	1
CO2	3	3	3	1	2	1	-	-	1	1	3	3	3	1
CO3	3	3	3	2	3	1	1	-	1	1	2	3	3	2
CO4	3	3	2	2	3	1	-	-	1	1	2	3	3	3
CO5	2	2	3	3	3	1	-	-	2	3	3	3	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

Department of Computer Science and Engineering

Artificial Intelligence & Machine Learning

290603(OLD)

COURSE OBJECTIVES:

1. To provide the fundamental knowledge of Artificial Intelligence and Machine Learning.
2. To understand the basic areas of AI & ML including problem solving, knowledge representation, reasoning, models, Loss functions.
3. To apply machine learning and optimization techniques to make predictions

Unit – I:

Introducing Artificial Intelligence: Introduction to AI - Intelligent Agents, Problem-Solving Agents, Searching for Solutions - Breadth-first search, Depth-first search, Hill-climbing search, simulated annealing search, Local Search in Continuous Spaces. Games - Optimal Decisions in Games, Water-Jug problem, Travelling salesman problem, Alpha–Beta Pruning.

Unit – II:

Knowledge Representation in AI: Need for Knowledge representation. Types of Knowledge. Knowledge and Intelligence, AI Knowledge cycle. Various approaches to Knowledge Representation. Requirements of Knowledge Representation System. Intelligent agent.

Constraint Satisfaction: Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Knowledge-Based Agents, Logic Propositional Logic, Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses

Unit – III:

Introduction to Machine Learning: Learning, Traditional vs Machine Learning, Types of ML, Classification and Regression model, Challenges faced by ML, Steps of developing an ML model, Bias and Variance, Regularization, Testing and validating, K cross validation, Hyper parameter tuning, Model Selection.

Model optimization and Evaluation: Confusion matrix, Recall, accuracy, precision, Model optimization, Cost/Loss Function, Derivative of cost function and non-derivative cost function, Gradient descent, Mini-batch Gradient, Descent (sckit-learn), Stochastic Gradient descent (sckit-learn), Momentum (sckit-learn)

Unit – IV:

Supervised Machine Learning Algorithm with python: Supervised Machine Learning Algorithms, k-Nearest Neighbors, Linear Regression, Logistic Regression, Log Loss, Support Vector Machine, Hinge Loss, Kernel Trick, polynomial Kernel, Decision Trees, Gini impurity, Ensemble learner, Random Forests.

Unit –V:

Unsupervised Machine Learning with python: The Curse of Dimensionality, Projection, Manifold Learning Principal component analysis, Clustering , K-Means, Limits of K-Means, Clustering for Image Segmentation, Clustering for Preprocessing, Clustering for Semi-Supervised

---

RECOMMENDED BOOKS:

1. Artificial Intelligence: A Modern Approach by Stuart J. Russell and Peter Norvig, Prentice Hall.
2. Artificial Intelligence: Elaine Rich, Kevin Knight, Mc-Graw Hill.
3. Pattern Recognition and Machine Learning, Christopher M. Bishop
4. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow by Aurélien Géron
5. Introduction to Machine Learning with Python by Andreas C. Müller & O'reilly

---

COURSE OUTCOMES: After completing the course, the student will be able to:

CO1: Define basic concepts of Artificial Intelligence & Machine Learning.

CO2: Illustrate various techniques for knowledge representation and processing.

CO3: Apply various model optimization and tuning approaches.

CO4: Develop a model using supervised/unsupervised machine learning algorithms for classification/prediction/clustering

CO5: Evaluate performance of machine learning algorithms on various data sets of a domain.

IoT System Design

3290601

List of Experiments

1. Implementation of LED Blinking with arduino/ Rasberry Pi.
2. Implementation of Temperature and Humidity Monitoring.
3. Interfacing between sensors and actuator.
4. Develop an intrusion detection system through motion sensors.
5. Develop an IoT-based Fire Detection System.
6. Develop an IoT-based Air Quality Monitoring.
7. Implementation of IoT-based Home Automation System
8. Implementation of MQTT-based IoT Communication. .
9. Implementing a mechanism for IoT Data Logging to Cloud.
10. Implement a mechanism to control the state of an LED using a Bluetooth-enabled microcontroller.
11. Implementation of RFID-based Access Control System.

COURSE OUTCOMES:

After completion of this course, the students would be able to:

1. **Demonstrate** the ability to interface sensors and actuators with microcontrollers
2. **Develop** IoT applications incorporating real-time data acquisition, cloud connectivity, and communication protocols
3. **Analyze** sensor-generated data for monitoring, automation, and decision-making in applications

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	3	2	1	1	1	1	1	1	3	3	3
CO2	3	2	3	3	3	1	1	1	1	1	1	3	3	3
CO3	3	3	3	3	3	2	1	1	1	1	1	3	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

IoT System Design

3290601

List of Skill Based Mini Projects

1. Smart Home Automation System
2. Weather monitoring system
3. Smart Plant Watering System
4. Health Monitoring Wearable
5. Trash Can Monitoring System
6. Parking Space Availability System
7. Smart Irrigation System
8. Home Security System
9. Energy Consumption Monitoring
10. Air Quality Monitoring
11. Smart Traffic Management System
12. IoT-Based Fire Detection System
13. IoT-Based Attendance System
14. Smart Lighting System
15. Smart Retail Shelf Monitoring
16. Smart Greenhouse
17. IoT-Based Crop Monitoring
18. IoT-Based Object Tracking
19. Water Quality Monitoring in Rivers and Lakes
20. Solar Panel Monitoring System
21. IoT-Based Green Building System

Artificial Intelligence & Machine Learning

3290602

List Of Experiment

1. Implement Depth first search for water jug problem
2. Write a Program to find the solution for travelling salesman Problem
3. Write a program to implement 8 puzzle problem
4. Write a program to implement Hill Climbing Algorithm
5. Write a program to implement A* Algorithm and AO* Algorithm
6. Perform exploratory data analysis and visualization after importing a .CSV file.
   - Handle missing data by detecting and dropping/ filling missing values.
   - Transform data using different methods.
   - Detect and filter outliers.
   - Perform Vectorized String operations on Pandas Series.
   - Visualize data using Line Plots, Bar Plots, Histograms, Density Plots and Scatter Plots.
7. Implement Various Regression algorithm for House Price Prediction (USA housing Dataset) and compare there accuracy using scikitlearn
   - Linear Regression
   - Polynomial Regression
   - Support Vector machine
8. Implement Regularized Regression for house price prediction and evaluate there accuracy using sckitlearn.
   - Ridge Regression
   - Lasso Regression
9. Implement Various Classification algorithm for iris data set and evaluate there performance.
   - Navie Bayes Classifier
   - Logistic Regression
   - Support vector Machine
   - Decision tree
10. Implement Various ensemble on housing and iris dataset and evaluate there performance
    - Voting classifier
    - Random Forest (Bagging and pasting)
11. Implement principle component analysis on any choosen dataset
12. Implement various clustering algorithm on choosen dataset
    - K-Mean
    - DBSCAN

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	1	**3**			**1**	**1**	**1**				
CO2	**3**	3	2	2	**3**			**1**	**1**	**1**				
CO3	2	2	**3**	2	**3**			**1**	**2**	**1**			2	
CO4	**3**	**3**	3	**3**	**3**	1	2	**1**	2	**2**		1		2

1 - Slightly; 2 - Moderately; 3 – Substantially

Artificial Intelligence & Machine Learning

3290602

List of Skill Based Mini Project

Artificial Learning Project

1. Build a bot which provides all the information related to students in college.
2. Build a virtual assistant for Wikipedia using Wolfram Alpha and Python.
3. Build a Banking Bot
4. Online Assignment Plagiarism Checker

Supervised learning projects

5. Implement a regressor for any Medical disease diagnosis.
6. Implement a Cervical Cancer Risk Classifier
7. Regression model for Video Game Sales Prediction
8. Calories Burnt Prediction using Machine Learning
9. Vehicle Count Prediction From Sensor Data
10. Regression model for predicting if song will be popular
11. Regression model for Customer Behavior Analysis
12. Regression model to predict health insurance cost
13. Titanic Survival Prediction

Unsupervised Learning Projects

14. Spam and not Spam Classifier
15. Spotify Music Recommendation System
16. Online Payment Fraud Detection using Machine Learning in Python
17. Customer Segmentation using Unsupervised Machine Learning in Python
18. Target Customer segmentation.
19. Topic Modeling for Twitter Customer Reviews
20. Bank-Note Authentication

माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर (म.प्र.), भारत
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA
Deemed to be University
(Declared under Distinct Category by Ministry of Education, Government of India)
NAAC ACCREDITED WITH A++ GRADE

Scheme of Evaluation
B. Tech. II Semester *(Computer Science and Design)*

(for batch admitted in academic session 2025-26)

S. No.	Course Code	Category Code	Course Name	Maximum Marks Allotted						Total Marks	Contact Hours per week			Total Credits	Mode of Learning	Mode of Major Exam.	Duration of Major Exam.
				Theory Block				Practical Block									
				Continuous Evaluation			Major Evaluation	Continuous Evaluation	Major Evaluation		L	T	P				
				Minor Evaluation I	Minor Evaluation II	Quiz/ Assignment		Lab Work & Sessional									
1.	29251201	DC	Computer Graphics & Animation	25	25	20	30	-	-	100	3	-	-	3	Face to Face	MCQ	2 Hrs
2.	29251202	DC	Object oriented Programming & Methodology	25	25	20	30	-	-	100	2	1	-	3	Face to Face	MCQ	2 Hrs
3.	29251203	DC	Computer System and Organization	25	25	20	30	-	-	100	2	1	-	3	Face to Face	MCQ	2 Hrs
4.	29251204	DC	Operating Systems	25	25	20	30	-	-	100	2	1	-	3	Face to Face	MCQ	2 Hrs
5.	29251205	ESC	**Basic Electrical & Electronics Engineering**	25	25	20	30	-	-	100	2	-	-	2	Face to Face	MCQ	2 Hrs
6.	29251206	DLC	Computer Graphics & Animation Lab	-	-	-	-	70	30	100	-	-	2	1	Experimental	AO	-
7.	29251207	DLC	Object oriented Programming & Methodology Lab	-	-	-	-	70	30	100	-	-	2	1	Experimental	AO	-
8.	29251208	DLC	**Electrical & Electronics Engineering Lab**	-	-	-	-	70	30	100	-	-	2	1	Experimental	AO	-
9.	29251209	SP	**Semester Proficiency$**	-	-	-	-	50	-	50	-	-	2	1	Face to Face	SO	-
10.	29251210	PBL	**Micro Project-II#**	-	-	-	-	70	30	100	-	-	2	1	Experiential	SO	-
11.	NECXXXXX	NEC	**Novel Engaging Course (Activity Based Learning)**	-	-	-	-	50	-	50	-	1	-	1	Interactive	SO	-
12.	SIP1XXXX	SIP	**Skill Internship Program(Soft Skill)**	-	-	-	-	60	-	60	-	-	-	2**	Experiential	SO	-
Total				**125**	**125**	**100**	**150**	**440**	**120**	**1060**	**11**	**04**	**10**	**22**	-	-	-
13.	29251211	MAC	Sustainability & Environmental Science	-	-	-	-	100	-	100	-	2	-	GRADE	Blended	SO	-
14.	29251212	MWS	Mandatory Workshop on Career Planning & Goal Setting at Department Level											GRADE	Interactive	MCQ	-

Summer Semester of six-eight week duration will be conducted for makeup of I & II semester examination.

$Semester Proficiency– includes the weightage towards ability/ skill/ competency /knowledge level /expertise attained etc. in the semester courses

MCQ: Multiple Choice Question **AO:** Assignment + Oral **PP:** Pen Paper **SO:** Submission + Oral **OB:** Open Book

** These credits will be transferred from Skill Internship Program (Soft Skill).

Micro Project-II will be presented and evaluated through an interdisciplinary project evaluation committee.

HSMC	BSC	ESC	DC	DE	SPC	OC	DLC	NEC	SP	SIP	SLP	PDC	PBL	MAC	MWS
0	0	1	4	0	0	0	4	1	1	0	0	0	1	1	1

Mode of Learning								Mode of Examination								Total Credits
Theory		Lab				NEC	SIP	Theory				Lab		NEC	SIP	
Face to Face	Online	Face to Face	Blended	Experiential	Experimental	Interactive	Experiential	PP	AO	MCQ	OB	SO	AO	SO	SO	
14		1	-	1	3	1	2			14		2	3	1	2	22
63.5		4.5		4.5	14	4.5	9			63.5		9	14	4.5	9	Credits %

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER GRAPHICS & ANIMATION
29251201

COURSE OBJECTIVES

- To provide an introduction to the theory and practice of computer graphics.
- To give a good exposure related to Computer Graphics algorithms and to design various graphics primitives.
- To enhance the proficiency in programming skills related to animation and graphics object Design

Unit-I

Introduction to Computer Graphics: Interactive Computer Graphics, Application of Computer Graphics, Random and Raster Scan Displays, Storage Tube Graphics Display, Calligraphic Refresh Graphics Display, Flat Panel Display, Refreshing, Flickering, Interlacing, Resolution, Bit Depth, Aspect Ratio etc, Societal and legal implications of CG, Real-world applications of CG in social problem-solving, sustainable use of graphics hardware

Unit-II

Scan Conversion Technique: Image representation, Line drawing: DDA, Bresenham's Algorithm. Circle Drawing: General Method, Mid-Point, DDA, Bresenham's Circle Generation Algorithm, And Ellipse Generation Algorithm, Curves: Parametric Function, BezierMethod, B-Spline Method.

Unit-III

2D & 3D Transformations: Translation, Rotation, Scaling, Reflection, Shearing, Inverse Transformation, Composite Transformation, World Coordinate System, Viewing Transformation, Representation of 3D object on Screen, Parallel and Perspective Projections.

Unit-IV

Clipping: Point clipping, Line Clipping, Simple Visibility Line Clipping Algorithm, Cohen Sutherland Line Clipping Algorithm, etc., Polygon Clipping, Convex and Concave Polygon, Sutherland Hodgeman Polygon Clipping Algorithm etc., Area Filling, Hidden Surface Elimination: Z- Buffer algorithm and Painter's Algorithm, Performance evaluation and comparison of graphics algorithms; Color Models: RGB, YIQ, CMY, HSV

Unit-V

Computer Animation: Modern animation basics and uses; 2D/3D types; Animation principles; AI-assisted animation; technical documentation and presentation of computer graphics workflows, Graphics project workflow and asset pipeline management **Image Manipulation and Storage:** Modern DIP basics; Digitization; Histogram and enhancement; AI-based image manipulation; Real-time GPU processing; Ethical issues in digital image manipulation and AI-generated graphics

RECOMMENDED BOOKS

- Computer Graphics, Donald Hearn and M.P. Becker, PHI Publication.
- Computer Graphics principle and Practice, FoleyVandam, Feiner, Hughes.
- Principles of Computers Graphics, Rogers, TMH.
- Computer Graphics, Sinha and Udai, TMH.
- Digital Image Processing, Gonzalez.
- Principle of multimedia RanjanParekh,TMH.

COURSE OUTCOMES

After completion of the course students will be able to:

CO1. Explain interactive Computer Graphics, various display devices and explore applications of computer graphics.

CO2. Illustrate various line generations, circle generation, curve generation and shape Generation algorithms.

CO3. Apply various 2-Dimensional and 3-Dimensional transformations and projections on Images.

CO4. Classify methods of image clipping and various algorithms for Line and Polygon clipping.

CO5. Analysis various color models, shading methods, animation and Digital Image Processing.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1		1	2	2	1		1		1	2	3
CO2	3	3	1	1	1							1	3	1
CO3	3	3	3	2	2				1		1	2	3	2
CO4	2	3	2	3	2							1	2	1
CO5	2	2	2	2	3	1	1	2	1	2	2	3	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

OBJECT ORIENTED PROGRAMMING & METHODOLOGY
29251202

COURSE OBJECTIVES

- To study about the concept of object oriented programming.
- To create C++ programs that leverage the object oriented features of the C++ Language.
- To apply object oriented or non-object oriented techniques to solve bigger computing problems.

Unit-I

Introduction to C++ and Object Oriented Concepts: Basics of C++, Tokens, I/O Statements, Structure of Program, Operators and Expressions, Flow of Control, Arrays, Structures, Functions and its type, Programming Techniques: Unstructured & Structured Programming, Object Oriented Paradigm, Features of Oops, Comparison with Procedural Oriented Programming & Object Oriented Programming.

Unit-II

Classes & Objects: Creating of Objects, Characteristics of Object, Static Data Member, Static Member Function, Array of Objects, Object as Arguments, Inline Function, Default Arguments, Friend Function, and Recursion. Introduction, Types of Constructors.

Unit-III

Polymorphism: Introduction, Type of Polymorphism: Compile Time Polymorphism & Run Time Polymorphism, Function Overloading, Operator Overloading: Binary Operators, Arithmetic Assignment Operators, Unary Operators, Rules for Operator Overloading, Pitfalls of Operator Overloading.

Unit-IV

Inheritance: Types of Inheritance, Virtual Base Classes, Abstract Classes, Constructors in Derived Classes, Nesting of Classes, Overriding Member Function. Pointers Overview, Pointers to Objects, This Pointer, Pointers to Derived Classes, Virtual Functions & Pure Virtual Function, Association, Type of Association, Aggregation, File Handling Concepts

Unit-V

Emerging Oops concept: Standard Template Library (STL) / Collections Framework: Utilizing pre-built data structures, applying associated algorithms and iterators for efficient data manipulation, Multithreading, Interface-Driven Design.

RECOMMENDED BOOKS

- C++ How to Program, H M Deitel and P J Deitel, Prentice Hall..
- Programming with C++, D Ravichandran, T.M.H.
- Computing Concepts with C++ Essentials, Horstmann, John Wiley.
- The Complete Reference in C++, Herbert Schildt, TMH.
- Object-Oriented Programming in C++, E Balagurusam

COURSE OUTCOMES

After completion of the course, students would be able to:

CO1. Describe various fundamental of object-oriented design for programming practices.

CO2. Apply fundamental Object-Oriented Programming principles such as encapsulation, inheritance, and polymorphism in real problem.

CO3. Develop robust and scalable software systems and applications.

CO4. Create modular, maintainable, and extensible code that adheres to industry best practices.

CO5. Evaluate Object-Oriented solutions and make informed design decisions

---

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	3												
CO2			2				3						1	
CO3					2				2			2	2	
CO4					1								1	
CO5					2							1	3	1

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER SYSTEM AND ORGANIZATION
29251203

COURSE OBJECTIVES

- To provide the fundamental knowledge of a computer system and its processing units.
- To provide the details of input & output operations, memory management and performance measurement of the computer system.
- To understand how computer represents and manipulate data.

Unit I

Introduction: Von-Neumann Model, Harvard Architecture, Various Subsystems, CPU, Memory, I/O, System Bus, CPU and Memory Registers, Program Counter, Accumulator, Register Transfer and Micro Operations: Tree-State Bus Buffers, Bus and Memory Transfers, Arithmetic Micro-Operation: Logic Micro-Operation, Shift Micro- Operation.

Unit- II

Computer Arithmetic: Addition and Subtraction with Signed-Magnitude, Multiplication Algorithm, Division Algorithm, Floating-Point Arithmetic Operations.

Central Processing Unit (CPU): General Purpose Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC).Hardwired and Micro programmed Control.

Unit -III

Input-Output Organization: Peripheral Devices, I/O Interface, Asynchronous Data Transfer, Modes of Transfer, Priority Interrupt, DMA (DMA Controller, DMA Transfer), Input-Output Processor (IOP), Data Transfer- Serial/Parallel, Simplex/ Half Duplex/ Full Duplex.

Unit -IV

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, AssociativeMemory, Cache Memory- Organization and Mappings, Replacement policies, Memory Management Hardware. **Introduction to Pipelining:** Concept of pipelining, Performance of pipelined processors, Pipeline speedup & efficiency, Non-pipelined vs pipelined execution, Pipeline Hazards

Unit-V

Introduction to Modern Processors: Intel: i-series, Apple: M-Series, ARM Cortex-A, AMD etc. Tensor Cores, Neural Processing Unit (NPU), AI Accelerators, Warp/Wavefront Execution, CUDA Cores, Matrix Multiplication Engine, Inference Engine, Unified Shader Architecture.

RECOMMENDED BOOKS

- Computer System Architecture, Morris Mano, PHI.
- Microprocessor Architecture, Programming and Applications with the 8085, Gaonkar,
- Computer Organization, Carl Hamacher, THM.
- Computer Architecture and Organization, J P Hayes, Mc-Graw Hills, New Delhi.

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. Explain the fundamental computer architectures and internal CPU operations.
CO2. Evaluate CPU organization, instruction design, addressing modes, and control unit implementation.
CO3. Describe input/output structures and analyze data transfer techniques including interrupts, DMA, and I/O processing.
CO4. Evaluate cache design, replacement strategies, memory management hardware, and pipelined execution performance.
CO5. Analyze modern processor and advanced architectures including AI accelerators, NPUs, Tensor Cores, and GPU execution models.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3			3							3	3	
CO2	3	3					3					2	3	3
CO3	3	2	2		3				2			3	3	3
CO4	3	3		3	2							3	3	3
CO5	3	3		3	3							2	3	3

1 - Slightly; 2 - Moderately; 3 - Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

OPERATING SYSTEMS
29251204

COURSE OBJECTIVES

- Provide basic knowledge of computer operating system structures and functioning.
- Compare several different approaches to memory management, file management and process management.
- Understand various problems related to concurrent operations and their solutions.

Unit- I

Basics of operating systems: Generations, Types, Structure, Services, System Calls, System Boot, System Programs, Protection and Security, User mode vs kernel mode.
Process management: Process Concepts, Process States, Process Control Block, Scheduling-Criteria, Scheduling Algorithms and their Evaluation, Threads, Threading Issues.

Unit-II

Process synchronization: Background, Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors.
Deadlock: System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery form Deadlock.

Unit-III

Memory management: Main Memory, Swapping, Contiguous Memory Allocation, Paging, Structure of Page Table, Segmentation, Virtual Memory, Demand Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.

Unit-IV

Storage management: Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, RAID Structure.
File system interface: File Concept, Access Methods, Directory Structure, File System Structure, Allocation Methods, and Free-Space Management.
System Protection: Goals, Principles, Domain of Protection, Access Matrix, Access Control

Unit-V

Emerging topics: Overview of Real-Time Operating Systems (RTOS), Brief introduction to mobile and cloud operating systems, Virtualization (Hypervisors, Type-1 & Type-2), Containerization Basics (Docker Concept)

RECOMMENDED BOOKS

- Operating System Concepts, Silberschatz, Ninth Edition, Willey Publication.
- Operating Systems, Internals and Design Principles, Stallings, Seventh Edition, Pearson Publication.
- OPERATING SYSTEMS, Dr. Sukomal Pal, AICTE
- Modern Operating Systems, Tanenbaum, Fourth Edition. Pearson Publication.

COURSE OUTCOMES

After the successful completion of this course, the student will be able to:

CO1. **Explain** the fundamental concepts, structure, and services of operating systems.
CO2. **Analyze** the mechanisms of process management, synchronization, and deadlock handling.
CO3. **Apply** memory management techniques to solve allocation and paging problems.
CO4. **Evaluate** the performance of CPU scheduling, disk scheduling, and page replacement algorithms.
CO5. **Design** solutions for issues related to system protection, security, and resource management.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	-	1	1	-	-	-	1	-	2	3	1
CO2	3	3	2	2	1	-	-	-	1	1	-	2	3	2
CO3	3	3	3	2	1	-	-	-	1	1	-	2	3	2
CO4	2	3	2	3	2	1	-	-	1	1	-	2	2	3
CO5	2	3	3	2	2	2	1	1	2	1	1	3	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BASIC ELECTRICAL & ELECTRONICS ENGINEERING
29251205

Course Objectives:

- Impart foundational knowledge in Electrical and Electronics Engineering.
- Enable students to analyze electric circuits, understand electrical machines, and implement digital systems.
- Explore emerging applications in industrial automation, smart grids, and renewable systems.

Unit I

D.C. Circuits Analysis: Voltage and Current Sources: Dependent and independent source. Source conversion. Kirchhoff's Law, Mesh and Nodal analysis. Network theorems: Superposition theorem, Thevenin's theorem & Norton's theorem and their applications.

Unit II

Single-phase AC Circuits: Generation of sinusoidal AC voltage, definitions: Average value, R.M.S. value, Form factor and Peak factor of AC quantity, Concept of Phasor, analysis of R-L, R-C, R-L-C Series and Parallel circuit, Power and importance of Power factor, Resonance in AC circuits.

Unit III

Transformer & Electrical Machines: Magnetic Circuits and Electromagnetism, Transformers: Construction, principle, types, losses & efficiency, OC & SC test DC Machines: Motor and Generator working Principles, Characteristics, Introduction to Induction Motors and Synchronous Machines.

Unit IV

Digital Electronics, Devices & Circuits: Number Systems, Logic Gates and Truth Tables, Diodes, Transistors (BJT), Multiplexers, Demultiplexers.

Unit V

Emerging Trends and Applications: Introduction to Smart Grids, Smart Meters, and Renewable Systems. Types of earthing, Sensors and Basic IoT Applications.

Recommended Books:

1. Basic Electrical and Electronics Engineering, D.P. Kothari and I.J. Nagrath, 2nd Edition, McGraw-Hill Education, 2020.
2. Basic Electrical and Electronics Engineering, S.K. Bhattacharya, 2nd Edition, Pearson Education, 2017.

3. Basic Electrical Engineering, V.N. Mittle and Arvind Mittal, 2nd Edition, McGraw-Hill Education, 2005.
4. Basic Electrical Engineering, A.E. Fitzgerald, David E. Higginbotham, and Arvin Grabel, 5th Edition, McGraw-Hill Education, 1981.
5. Principles of Electrical Engineering and Electronics, V.K. Mehta and Rohit Mehta, Revised Edition, S. Chand Publishing, 2019.

Course Outcomes (COs):

At the end of the course, the student will be able to:

CO1. **Apply** fundamental laws and network theorems to analyze DC circuits
CO2. **Analyze** single-phase series & parallel AC circuits for calculation of power, power factor, and resonance conditions.
CO3. **Explain** the working principles, construction, and operational characteristics of transformers, DC machines, and induction motors.
CO4. **Design** basic digital logic circuits using logic gates, flip-flops, and counters
CO5. **Discuss** the concepts of smart meter, smart grids, earthing, and IoT systems to emerging industrial applications in automation and renewable energy systems.

Course Articulation Matrix

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	*PSO1	*PSO2
CO1	3	3	2	2	1	-	-	-	-	-	-	1		
CO2	3	3	2	2	1	-	-	-	-	-	-	1		
CO3	3	2	3	2	2	1	-	-	-	-	-	2		
CO4	3	3	3	2	1	-	-	1	2	2	-	1		
CO5	3	2	3	2	3	2	2	2	-	1	1	2		

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMPUTER GRAPHICS & ANIMATION LAB
29251206

List of Experiments

1. Installation and Introduction to OpenGL basics, graphic functions, commands for compiling and executing an OpenGL Program.
2. Write a Program to create an output window, to plot a point with given coordinates and other basic demonstrations.
3. Write a Program to implement DDA Line Drawing Algorithm.
4. Write a Program to implement Bresenham's Line Algorithm.
5. Write a Program to implement Mid-Point Circle Algorithm.
6. Performance evaluation of Z-Buffer vs. Painter's Algorithm using runtime measurements.
7. Write a Program to implement following 2D transformations:
   i. Translation of a point, line and polygon.
   ii. Scaling of a line and polygon.
   iii. Rotation of a line and polygon around origin.
8. Write a Program to implement Bezier curve & B-spline.
9. Write a Program to implement Flood Filling Algorithm & Boundary Filling Algorithm using polygon.
10. Write a program to design a short 2D/3D educational or social-awareness animation (e.g., road safety, hand-washing steps, disaster safety, sign-language gesture animation).
11. Write a program to visualize a real-world social dataset (such as pollution levels, rainfall intensity, or traffic density) using animated color gradients and interpret how the visualization supports societal awareness.

COURSE OUTCOMES

After completion of the course students will be able to:

CO1. Develop the ability to implement and manipulate algorithms for drawing geometric shapes, transformations, and color models.
CO2. Implement and apply 2D and 3D transformations (translation, rotation, scaling, reflection, etc.) on graphical objects.

CO3. Implement various animation techniques, such as keyframing, interpolation, and skeletal animation, to create smooth and realistic motion.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	2	3	1	1	1	1	1		1	3	1
CO2	3	2	2	1	3				1	1	1	1	3	1
CO3	2	2	3	1	3	2	1	1	2	3	2	2	2	3

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

OBJECT ORIENTED PROGRAMMING & METHODOLOGY
29251207

List of Experiments

1. Write a program to swap two integers without using third variable. The swapping must be done in a function of a particular class.

2. Write a program that uses a class where the member functions are defined outside a class.

3. Design a class to represent a bank account. Which includes account number, name of the depositor, type of the account, balance amount in the account. Define Methods, to assign initial values, to Deposit an amount, to Withdraw amount after checking balance, to display name and balance.

4. Write a program to find the greater of two given numbers in two different classes using friend function.

5. Create an inheritance hierarchy of Rodent, Mouse, Gerbil, Hamster etc. In the base class provide methods that are common to all Rodents and override these in the derived classes to perform different behaviors, depending on the specific type of Rodent. Create an array of Rodent, fill it with different specific types of Rodents and call your base class methods.

6. Create two classes: Polar and Cartesian, to represent Polar and Cartesian coordinates of a point. Demonstrate how to convert Polar coordinates to Cartesian coordinates by writing the conversion code in source class.

7. Write a program to demonstrate anomaly caused in Multi-path Inheritance. Also, write a program to overcome the anomaly.

8. Create an abstract class Shape which has a field P1=3.14 as final and it has an abstract method Volume. Make two sub-classes 'Cone' and 'Sphere' from this class and they should print their volume

9. Create a class called LIST with two pure virtual function store() and retrieve().To store a value call store and to retrieve call retrieve function. Derive two classes stack and queue from it and override store and retrieve

10. Write a program to demonstrate working of various file handling operations in C++

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. Demonstrate the use of various OOPs concepts with the help of programs.

CO2. Formulate solutions to problems demonstrating usage of control structures, modularity, I/O and other standard language constructs.

CO3. Enumerate Implement appropriate object orient programming concepts for solving real world problems.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1		1											1
CO2		1	2				1						1	
CO3		1			1									2

1 - Slightly; 2 - Moderately; 3 – Substantially

BASIC ELECTRICAL & ELECTRONICS ENGINEERING
29251208

LIST OF EXPERIMENT

1. To verify Kirchhoff's Current Law & Kirchhoff's Voltage Law.
2. To verify Superposition Theorem
3. To determine resistance & inductance of a choke coil.
4. To determine active & reactive power in a single phase A.C circuit.
5. To determine voltage ratio & current ratio of a single phase transformer.
6. To determine the polarity of a single phase transformer.
7. To perform open circuit & short circuit test on a single phase transformer.
8. To study multimeter & measure various electrical quantities
9. To study of constructional details of DC machine.
10. To determine the V-I characteristics of diode in forward bias & reverse bias condition.
11. To determine phase and line quantities in three phase star and delta connection
12. To study of effect of open and short circuits in simple circuits
13. To plot Transistor CB characteristics (Input and Output)
14. To plot Transistor CE characteristics (Input and Output)
15. **Study the output characteristics of a solar PV panel under varying conditions**
16. **Develop a simple IoT system to monitor temperature and humidity using sensors.**

Course Outcomes:

After the completion of the lab, the student will be able to –

CO1 **Demonstrate** the ability to operate lab equipment and instruments relevant to the electrical engineering field

CO2 **Collect** experimental data accurately and effectively in ethical manner

CO3 **Integrate** theoretical knowledge from coursework into practical applications and experiments

CO4 **Communicate** experimental results effectively through oral presentations and written documentation

CO5 **Demonstrate** responsibility and professionalism in the completion of lab tasks and assignments

CO6 **Show** willingness to learn new techniques, tools, or methods to enhance practical engineering skills

Course Articulation Matrix

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	3	3	3	3	2	-	-	-	2		
CO2	2	3	2	3	2	3	2	3	-	2	-	2		
CO3	3	3	3	3	2	2	2	2	2	3	2	3		
CO4	1	2	2	3	-	2	2	3	3	3	2	2		
CO5	-	-	1	2	-	3	3	3	3	3	2	3		
CO6	2	2	2	2	3	3	3	2	3	3	2	3		

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MICRO PROJECT-II
29251210

Course Objective:

- Understand and apply basic programming constructs, file handling, and simple data structures to solve small-scale computational problems.
- Develop practical applications integrating fundamental concepts of algorithms, database operations, and introductory networking.
- Build interactive CLI/GUI/Web interfaces using beginner-friendly tools to enhance user experience and problem-solving.

List of Projects

1. Simple Calculator (with History Log)

Addition, subtraction, multiplication, division and stores past calculations in a file

2. Password Strength Checker

Checks password quality using regex rules.

3. Basic To-Do List (CLI)

Add, delete, view simple tasks (stored in a text file).

4. Student Grade Calculator

Based on marks → grade/percentage.

5. Electricity Bill Calculator

Uses slabs and conditions.

6. Number Guessing Game

Random number + attempts counter.

7. Currency Converter (Hardcoded Rates)

Simple menu-driven program.

8. String Analyzer Tool

Counts vowels, words, characters.

9. Marks/Attendance Percentage Calculator

Takes multiple inputs, shows summary.

10. Mini Banking Menu (Without DB)

Deposit, withdraw, balance using variables.

11. Library Management (File-Based)

Add, search, delete books; store in text file.

12. Contact Book (Array/List Based)

Add, search, update contacts.

13. Shopping List Manager

Simple item list + remove items.

14. Student Record System (File-Based)

Add student info and retrieve records.

15. Simple Inventory Tracker

Products stored in file; stock updates.

16. Railway/Bus Reservation Prototype (Simple)

Shows seats as array, books a seat.

17. Basic Billing System for Shop

Calculates total & prints bill to text file.

18. Hostel Attendance Tracker

List of students → mark present/absent.

19. Quiz System (File-Based Questions)

Reads Q/A from file; calculates score.

20. Employee Payroll Calculator

Basic salary, allowances, net pay.

21. Maze Solver (DFS/BFS optional)

User enters maze layout → find path.

22. Banking System Simulation (File Handling)

Account creation + deposit/withdraw logs.

23. CLI Chat Simulator (Very Simple Sockets or Simulation)

Even without networking: simulate chat between two users (turn-based).

24. Real-Time Voting System (Simple Data Structures)

Add votes → count votes dynamically.

25. Online Quiz System (With Very Simple SQL)

26. Blogging System (File-Based)

Create, edit, delete blog posts.

27. Basic Weather App (Public API Optional)

28. Portfolio Website (HTML/CSS Only)

Very simple personal profile site.

29. Smart Home Simulation Console App

Turn appliances ON/OFF, store status in memory or file.

30. Mini Inventory Store with Bill Generation

Manage items + generate printed bill.

Recommended Books:

- Head First Programming, David Griffiths, Paul Barry, O'Reilly Media.
- Let Us C, Yashavant Kanetkar, BPB Publications.
- Data Structures Using C, Reema Thareja, Oxford University Press.
- Head First HTML and CSS, Elisabeth Robson & Eric Freeman, O'Reilly Media

Course Outcomes:

Upon successful completion, students will be able to:

CO1: Apply fundamental programming concepts, including variables, control structures, functions, arrays, strings, and file handling, to design and implement simple computational applications.

CO2: Use file handling or very simple database operations to store, retrieve, and update information.

CO3: Develop functional CLI/GUI/Web-based mini projects using beginner-friendly tools and demonstrate problem-solving ability through proper documentation, testing, and execution of applications.

Course Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	1	2	2	-	-	1	1	-	1	2	1
CO2	3	2	2	1	3	3	-	-	1	1	2	1	3	2
CO3	2	2	3	1	3	1	-	1	3	3	2	3	3	2

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Sustainability & Environmental Science

29251211

Course Objective:

To equip students with a comprehensive understanding of environmental science, pollution control, sustainability, and global frameworks, enabling them to analyze environmental challenges and contribute to sustainable solutions through informed decision-making and responsible practices.

Unit I:

Introduction to Environmental Science: definition, importance and its components. Ecosystem and its components. Water cycle, carbon cycle, food chain, energy flow in ecosystem. Current state of environment in India and world; Underlying reasons (root causes) of modern environmental degradation (social, psychological, cultural). Introduction to Environmental pollution: air, water, noise, soil, thermal and radioactive.

Unit II:

Environmental Pollution and Management: air, water, noise, soil, thermal and radioactive. Causes, impacts, pollution control techniques and mitigation strategies. Solid waste management: Principles of waste management, different components of waste management system and introduction to management of hazardous waste like e-waste, plastic waste. Global environmental Issues: Climate change, global warming, ozone layer depletion, urban heat island

Unit III:

Environmental policies and laws in India: Environmental Protection Act, Water Act, Air Act. **Overview of global environmental policies and frameworks:** Kyoto protocol, Montreal protocol, COP summits. Introduction to clean development mechanism, carbon credit, carbon trading. Environmental audit.

Unit IV:

Sustainability concepts: definition, importance, pillars of sustainability (economic, environmental, and social). Sustainable development. Overview of UN Sustainable Development Goals (SDGs) and their global relevance. Concept of circular economy, resource efficiency,

energy conservation, green buildings and sustainable manufacturing.

Unit V:

Sustainable Energy solutions: New energy sources: need of new sources, different types of new energy sources, application of hydrogen energy, ocean energy sources, and tidal energy conversion. Concept, origin and power plant of geothermal energy. Renewable energy sources like water, wind etc. Overview of sustainable materials and construction practices. Introduction to sustainable transportation systems and sustainable water infrastructure.

Recommended Books:

1. D. K. Asthana, Meera Asthana, A Text Book of Environmental Studies, S Chand & Co., New Delhi.

2. S. K. Dhameja, Environmental Engineering & Management, S K Kataria & Sons, New Delhi

3. C. S. Rao, Environmental Pollution Control Engineering, C.S. Rao, New Age International Publishers

4. A. K. Gupta, Environmental Sustainability and Green Technologies, PHI Learning.

Course Outcomes:

Upon completion of the course, a student will be able to

CO1. **Explain** the fundamental concepts of environmental science, including ecosystems and the causes of environmental degradation.

CO2. Analyze the sources, causes, and impacts of air, water, and solid waste pollution and propose appropriate mitigation strategies.

CO3. Evaluate the effectiveness of environmental policies and global frameworks in addressing environmental challenges.

CO4. Explain the concepts of sustainability and sustainable development goals.
CO5. Apply various solutions for achieving sustainable development.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1													
CO2													
CO3													
CO4													
CO5													

1 - Slightly; 2 - Moderately; 3 – Substantially

माधव प्राद्यागिका एव विज्ञान सस्थान, ग्वालियर (म.प्र.), भारत
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA
Deemed to be University
(Declared under Distinct Category by Ministry of Education, Government of India)
NAAC ACCREDITED WITH A++ GRADE

Scheme of Evaluation

B. Tech. IV Semester *(Computer Science and Design)*

(for batch admitted in academic session 2024-2

S. No.	Course Code	Category Code	Course Name	Maximum Marks Allotted						Total Marks	Contact Hours per week			Total Credits	Mode of Learning	Mode of Major Exam.	Duration of Major Exam.
				Theory Block				Practical Block									
				Continuous Evaluation			Major Evaluation	Continuous Evaluation	Major Evaluation		L	T	P				
				Minor Evaluation I	Minor Evaluation II	Quiz/ Assignment		Lab Work & Sessional									
1.	29242201	DC	**Data Science**	25	25	20	30	-	-	100	2	1	-	3	Face to Face	MCQ	2 Hrs
2.	29242202	DC	Design and Analysis of Algorithms	25	25	20	30	-	-	100	2	1	-	3	Face to Face	MCQ	2 Hrs
3.	29242203	DC	Theory of Computation	25	25	20	30	-	-	100	2	1	-	3	Face to Face	MCQ	2 Hrs
4.	29242204	DC	Data communication and Networks	25	25	20	30	-	-	100	2	1	-	3	Face to Face	MCQ	2 Hrs
5.	29242205	DC	Design Pattern	25	25	20	30	-	-	100	3	-	-	3	Face to Face	MCQ	2 Hrs
6.	29242206	DLC	**Data Science Lab**	-	-	-	-	70	30	100	-	-	2	1	Experimental	AO	-
7.	29242207	DLC	Design and Analysis of Algorithms Lab	-	-	-	-	70	30	100	-	-	2	1	Experimental	AO	-
8.	29242208	DLC	Competitive Programming	-	-	-	-	70	30	100	-	-	2	1	Experimental	AO	-
9.	29242209	SP	**Semester Proficiency$**	-	-	-	-	50	-	50	-	-	2	1	Face to Face	SO	-
10.	29242210	PBL	**Macro Project-II#**	-	-	-	-	70	30	100	-	-	2	1	Experiential	SO	-
11.	NECXXXXX	NEC	**Novel Engaging Course (Activity Based Learning)**	-	-	-	-	50	-	50	-	1	-	1	Interactive	SO	-
12.	SIP2XXXX	SIP	**Skill Internship Program**	-	-	-	-	60	-	60	-	-	-	2**	Experiential	SO	-
Total				**125**	**125**	**100**	**150**	**440**	**120**	**1060**	**11**	**05**	**10**	**23**	-	-	-
13.	29242211	MAC	Project Management, Economics & Financing	`	-	-	-	100	-	100	-	2	-	GRADE	Blended	SO	-
14.	29242212	MWS	Mandatory Workshop on Intellectual Property Rights at Department Level											GRADE	Interactive	MCQ	

Summer Semester of six-eight week duration will be conducted for makeup of previous semester examination.

Additional Course for Honours or Minor Degree: Permitted to opt for maximum two additional courses for the award of Honours or Minor Degree

$Semester Proficiency– includes the weightage towards ability/ skill/ competency /knowledge level /expertise attained etc. in the semester courses

MCQ: Multiple Choice Question **AO:** Assignment + Oral **PP:** Pen Paper **SO:** Submission + Oral **OB:** Open Book

Macro Project-II will be presented and evaluated through an interdisciplinary project evaluation committee.

** These credits will be transferred from Skill Internship Project.

PC	BSC	ESC	DC	DE	SPC	OC	DLC	NEC	SP	SIP	SLP	PDC	PBL	MAC	MWS
1	0	0	5	0	0	0	3	1	1	0	0	0	1	1	1

Mode of Learning				Mode of Examination				Total Credits
Theory	Lab	NEC	SIP	Theory	Lab	NEC	SIP	

Face to Face	Online	Face to Face	Blended	Experiential	Experimental	Interactive	Experiential	PP	AO	MCQ	OB	SO	AO	SO	SO	
15		1	1		3	1	2			15		2	3	1	2	23
65.1		**4.4**	**4.4**		**13**	**4.4**	**8.7**			**65.1**		**8.7**	**13.1**	**4.4**	**8.7**	**Credits %**

List of courses to be opted for Honors or Minor specialization in V Semester

Minor Specialization*(to be opted by students of Other Department)*
Introduction to Database Systems
Object Oriented System Development Using UML, Java And Patterns

Honors*(to be opted by students of Parent Department)*	
Track1	**Course**
Information Security	Foundations of Cryptography, IIIT Bangalore
	Cryptography and Network Security
IoT	Introduction To Internet Of Things
	Wireless Ad Hoc and Sensor Networks
High Performance Computing	Parallel Computer Architecture
	Advanced Computer Architecture

***Course run through SWAYAM/NPTEL/MOOC Learning Based Platform**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DATA SCIENCE

29242201

COURSE OBJECTIVES

- To provide the fundamental knowledge of Data Sciences, along with essential Python programming skills..
- Apply data manipulation, statistical analysis, and visualization techniques using Python libraries like NumPy and pandas.
- Develop, implement, and evaluate machine learning models while using statistical methods to derive insights and validate results.

Unit – I:

Introduction to Data Science: Introduction, Definition, applications of Data Science, Impact of Data Science, Data Analytics Life Cycle, role of Data Scientist.

Basics of Python: Essential Python libraries, Python Introduction- Features, Identifiers, Reserved words, Indentation, Comments, Built-in Data types and their Methods: Strings, List, Tuples, Dictionary, Set, Type Conversion- Operators. Decision Making: Looping-Loop Control statement, Math and Random number functions. User defined functions.

Vectorized Computation: The NumPy ndarray- Creating ndarrays- Data Types for ndarrays- Arithmetic with NumPy Arrays- Basic Indexing and Slicin

Unit-II

Data Analysis (with Pandas): Series, DataFrame, Essential Functionality: Dropping Entries, Indexing, Selection, and Filtering- Function Application and Mapping- Sorting and Ranking. Summarizing and Computing Descriptive Statistics – Mean, Standard Deviation, Skewness and Kurtosis. Unique Values, Value Counts, and Membership. Reading and Writing Data in Text Format.

Unit-III

Exploratory Data Analysis and Visualisation: Handling Missing Data, Data Transformation: Removing Duplicates, Transforming Data Using a Function or Mapping, Replacing Values, Detecting and Filtering Outliers, Functions in pandas. Plotting with pandas: Line Plots, Bar Plots, Histograms and Density Plots, Scatter or Point Plots.

Unit-IV

Introduction to Machine Learning: Types of Learning, Linear Regression- Simple Linear Regression, Implementation, plotting and fitting regression line, Logistic Regression, K-Nearest Neighbors (KNN), K-Means Clustering.

Unit-V

Hypothesis Testing: Mean and Variance Tests, p-value, Errors, Z-Test, t-Test, Paired t-Test, and F-Test, Analysis of Variance (ANOVA) and Contingency Table Analysis, Model Evaluation Metric

---

RECOMMENDED BOOKS

1. Cathy O'Neil and Rachel Schutt , "Doing Data Science", O'Reilly, 2015.
2. David Dietrich, Barry Heller, Beibei Yang, "Data Science and Big data Analytics", EMC 2013
3. Artificial Intelligence: A Modern Approach by Stuart J. Russell and Peter Norvig, Prentice Hall.
4. Pattern Recognition and Machine Learning, Christopher M. Bishop
5. James, Gareth, et al. An introduction to statistical learning. Vol. 112. New York: springer, 2013.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1: Analyze Data Science concepts and apply Python programming for data tasks, including data manipulation with NumPy.

CO2: Analysis of the data for applying various statistical modeling approaches.

CO3: Develop expertise in managing missing data and assessing the impact of visualizations on data insight communication.

CO4: Design and implement machine learning algorithms and assess model performance.

CO5: Develop statistical tests and **evaluate** machine learning models.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	1	3	2	-	-	1	2	1	1	3	2
CO2	3	3	2	2	3	2	-	1	1	2	2	1	3	3
CO3	2	2	2	1	3	2	-	1	1	3	3	2	3	2
CO4	3	3	3	2	3	3	-	1	1	2	3	2	3	3
CO5	3	3	2	2	3	3	1	1	1	2	2	2	3	3

(1 - Slightly; 2 - Moderately; 3 – Substantially)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DESIGN AND ANALYSIS OF ALGORITHMS

29242202

Course Objectives

- To develop the ability to analyze algorithm efficiency
- To learn and apply major algorithm design paradigms
- To understand computational complexity concepts

Unit-1

Introduction to Computational Model: RAM model, Algorithms and its importance, Recurrences and Asymptotic Notations, Growth of function, Mathematical Analysis of Non-Recursive and Recursive Algorithm, Review of Sorting & Searching Algorithms, Basic Tree and Graph Concept: Binary Search Trees, Height Balanced Tree, B-Trees and Traversal Techniques

Unit-2

Divide and Conquer Method: Introduction and its Examples such as Finding the maximum and minimum, Binary Search, Merge Sort, Quick Sort and Stassen's Matrix Multiplication.

Unit-3

Greedy Method: Introduction, Characteristics, greedy activity selection. Minimum Cost Spanning Trees: Prim's and Kruskal's Algorithm, knapsack Problem, Single Source Shortest Path: Dijkstra's single source shortest path algorithm, Huffman Coding.

Unit-4:

Dynamic Programming and Backtracking: Introduction, The principle of Optimality, Examples of Dynamic Programming Methods such 0/1 Knapsack, Travelling salesman problem, Floyds All Pairs Shortest Path, Longest Common Subsequence and Reliability Design. NP Completeness: Introduction, Class P and NP, Polynomial Reduction, NP-Hard and NP-Complete problem.

Backtracking Concept and its Examples like 4-Queen's Problem, Knapsack problem Hamiltonian Circuit Problem, etc.

Unit-5:

Randomized Algorithms: Las Vegas and Monte Carlo algorithms, Randomized Quicksort, Randomized Search, Randomized Hashing,

Approximation Algorithms: Approximation ratio & performance guarantees, Greedy approximation

REFERENCE BOOKS: -

- Fundamentals of Computer Algorithms, Horowitz & Sahani, Universities press
- Introduction to Algorithms, Coreman Thomas, Leiserson CE, Rivest RL, PHI.
- Design & Analysis of Computer Algorithms, Ullman, Pearson.
- Algorithm Design, Michael T Goodrich, Robarto Tamassia, Wiley India

---

COURSE OUTCOMES:

After Completion of this course, the students would be able to:

CO1 **analyze** the time and space complexity of algorithms using recurrence relations, asymptotic notations, and apply these concepts to basic sorting, searching, tree, and graph operations.

CO2: **Apply** the divide-and-conquer strategy to design and evaluate algorithms for problems such as searching, sorting, maximum–minimum finding, and matrix multiplication.

CO3: **Implement and analyze** greedy algorithms for optimization problems including activity selection, minimum spanning trees, shortest paths, knapsack, and Huffman coding.

CO4: **Use** dynamic programming principles to solve complex optimization problems such as knapsack, TSP, all-pairs shortest paths, LCS, and reliability design.

CO5: **Apply** randomized and approximation algorithm technique to design efficient solutions for computationally hard problems and evaluate their performance guarantees.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	3	3	1	2	1	1	1	1	2	2	2
CO2	3	3	3	3	3	1	1	1	1	1	1	2	2	2
CO3	3	3	3	3	3	2	1	1	1	1	1	1	2	2
CO4	3	3	2	3	2	1	1	1	1	1	1	1	2	2
CO5	3	3	3	3	2	1	1	1	1	1	1	2	2	2

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THEORY OF COMPUTATION

29242203

COURSE OBJECTIVE

- **Understand** the fundamental concepts of formal languages, automata, mathematical proofs for computation and algorithms and grammars.
- **Analyze** different types of automata and their capabilities in recognizing languages.
- **Design** finite automata, pushdown automata, and Turing machines for solving computational problems.

Unit-I

Introduction to Automata Theory: Automata, Computability and Complexity, Alphabet, Symbol, String, and Formal Languages, Examples of automata machines, Finite Automata as a language acceptor and translator, Moore machines and Mealy machines, Composite Machine, Conversion from Mealy to Moore and vice versa.

Unit-II

Types of Finite Automata: Non Deterministic Finite Automata (NDFA), Deterministic finite automata(DFA), conversion of NDFA to DFA, minimization of automata machines, regular expression, Arden's theorem. Pumping lemma, applications, Union, Concatenation, Closure properties of regular languages, 2 way DFA.

Unit-III

Grammars: Types of grammar, context sensitive grammar, context free grammar, regular grammar. Derivation trees, Rightmost and Leftmost derivations of Strings, ambiguity in grammar, simplification of context free grammar, killing null and unit productions, conversion of grammar to automata machine and vice versa, Chomsky hierarchy of grammar,. Chomsky Normal Form (CNF) and Greibach Normal Form (GNF).

Unit-IV

Push down Automata: Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack, Example of PDA, deterministic and non-deterministic PDA, conversion of PDA into context free grammar and vice versa, CFG equivalent to PDA.

Turing Machine: Techniques for construction. Universal Turing machine Multitape, Multihead and Multidimensional Turing machine, N-P complete problems. Decidability and Recursively Enumerable Languages, Halting problem of Turing machine, The post correspondence problem (PCB),

Unit-V

Application of automata machines in Switching Circuits, Digital circuit design, verify the behavior of complex digital circuits, Study and insights into the relationship between classical and quantum computing models, Application of automata machines in speech recognition and robotics

---

RECOMMENDED BOOKS

- Introduction to Automata Theory Language & Computation, Hopcroft& Ullman, Narosa Publication.
- Element of the Theory Computation, Lewis &Christors,Pearson.
- Theory of Computation, Chandrasekhar &Mishra,PHI.
- Theory of Computation, Wood, Harper &Row.
- Introduction to Computing Theory, Daniel I-A Cohen,Wiley.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1. Describe the fundamental concepts of finite automata.

CO2. Explain Finite State Machine, basic grammar concepts and context-free language principles.

CO3. Design Automata machines and their relation to CFLs.

CO4. Analyze decidable and undecidable problems using standard theoretical methods.

CO5. Apply automata theory, languages and computation in engineering application.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1		1	2				2	1	2		
CO2	3		1	3	2		3	2	2		2		1	2
CO3	3	3	2		1	3		1	2	3		2	2	2
CO4	3	2	1	2	1	2	2		2	3		2	1	2
CO5	3	2		1	2	2		3		2		1	2	1

Slightly; 2 - Moderately; 3 - Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DATA COMMUNICATION AND NETWORKS

29242204

COURSE OBJECTIVES

- Provide foundational knowledge of data communication principles, transmission media, and network components.
- Develop understanding of network protocols, architectures (OSI, TCP/IP), routing, switching, and addressing.
- Enable students to analyze, design, and troubleshoot data communication systems and modern computer networks.

UNIT–1: Introduction to Data Communication & Networking

Fundamentals of data communication: components, data flow, signals, and communication models, Types of data: analog and digital, transmission modes (simplex, half-duplex, full-duplex).Types of networks: LAN, MAN, WAN, PAN, and internetworks, Network topologies: mesh, star, bus, ring, hybrid, OSI Reference Model: functions of each layer, service primitives, service interfaces, TCP/IP Protocol Suite: layers, protocol overview, comparison with OSI model. Standards and organizations: IEEE, ISO, ITU-T, IETF.

UNIT–2: Physical Layer & Transmission Technologies

Transmission media: **Guided media:** twisted pair cables, coaxial cables, optical fiber; characteristics, advantages, limitations,**Unguided media:** radio waves, microwaves, infrared; spectrum, propagation, Switching technologies: circuit switching, packet switching, message switching, virtual circuits, Multiplexing techniques: Frequency Division (FDM), Time Division (TDM), Wavelength Division (WDM), Encoding and modulation: line coding schemes (NRZ, RZ, Manchester), ASK, FSK, PSK,Transmission impairments and performance: attenuation, distortion, noise, data rate limits (Nyquist, Shannon).

UNIT–3: Data Link Layer & Media Access Control

Data link layer design issues: framing, flow control, and error control, Error detection and correction: parity, checksums, CRC; Hamming code, Flow control protocols: Stop-and-Wait, Go-Back-N, Selective Repeat ARQ,Medium Access Control (MAC):Random access protocols: ALOHA, Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA,Controlled access: polling, token passing, Link-layer technologies: Ethernet (traditional and switched), Fast Ethernet, Gigabit Ethernet, Wireless LANs: IEEE 802.11 standards, Bluetooth basics,Link-layer devices: hub, switch, bridge; VLAN concept.

UNIT–4: Network Layer & Transport Layer

Functions of network layer: logical addressing, routing, forwarding, IPv4 addressing: classes, subnetting, supernetting, CIDR, VLSM, IPv4: structure, addressing types, Routing algorithms: Static vs dynamic routing, distance vector routing, link state routing, Bellman-Ford, Dijkstra algorithms, Congestion control.

Transport Layer: TCP: TCP Reliable data transfer, Connection Establishment & Release, TCP Frame,. UDP: Format, Pseudo header.

Unit- 5

Study of network packet analyzer tools: Wireshark, CISCO packet Tracer etc. Scanner Tools: Nmap, Nessus etc.

Software-Defined Networking (SDN), Network Function Virtualization (NFV), Green Networking, Programmable Networks - Introduction to P4, SmartNICS and P4 switches, Data Center Networking (DCN) – Introduction, DCN - Deep Dive (Network topologies, Container Network Interfaces)

RECOMMENDED BOOKS

- Data and Computer Communication - W. Stalling, Pearson
- Internetworking with TCP/IP - Vol. - I - D.E. Comer, PHI
- Data Communication & Networking -B.A. Forouzan
- ISDN and Broad band ISDN with Frame Relay & ATM - W. Stalling
- LANs - Keiser

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1: Explain the fundamental concepts of data communication, network architectures, and modern communication standards.

CO2: Describe the characteristics of transmission media, modulation techniques, switching methods.

CO3: Apply data link layer mechanisms including framing, error control, flow control, MAC protocols, and LAN/WLAN technologies for reliable communication.

CO4: Analyze IP addressing, subnetting, routing algorithms, routing protocols, and advanced networking concepts.

CO5: Analyze programmable networking concepts including P4 language, SmartNICs, and P4-based switching for high-performance and flexible packet processing.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	1	1	2	1				2		2	2	1
CO2	3	2	1	2	1		1			2		2	2	1
CO3	2	3	2	2	2	2		1		1	1	2	2	1
CO4	3	3	2	3	2		1		1	3	1	3	1	2
CO5	2	3	2	2	3	1	2	1	2	3	2	3		1

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DESIGN PATTERN

29242205

COURSE OBJECTIVES

- To provide the fundamental concepts and principles of design patterns in software development.
- Explore the different types of design patterns and their classifications: creational, structural, and behavioral.
- Learn how to analyze and identify design pattern opportunities in software design and architecture.

Unit-I

Introduction to Design Patterns: Overview of design patterns, Importance of design patterns in software development, Types of design patterns: creational, structural, and behavioral, Object-Oriented Principles in Python, Role of Python Features in Implementing Design Patterns, UML diagrams for design patterns, Common design principles and SOLID principles.

Unit-II

Creational Design Patterns: Singleton pattern: Definition and purpose of the Singleton pattern, single instance and global access, Factory pattern: Factory pattern and its role in creating objects, Abstract factory pattern, Prototype pattern, Case study for creational design pattern.

Unit-III

Structural Design Patterns: Adapter pattern: Definition and purpose of the Adapter pattern, interfaces and the need for adaptation, Decorator pattern: Decorator pattern and its role in dynamically adding behavior to objects, Facade pattern, Bridge pattern: decoupling abstractions from their implementations, Case study for structural design pattern.

Unit-IV

Behavioral Design Patterns: Observer pattern: subject and observer interfaces, Strategy pattern: strategy interfaces and concrete strategies, Template pattern, State pattern: state interfaces, concrete states, and context objects, Case study for Behavioral design pattern.

Unit-V

Design patterns in Microservices Architecture, Concepts of DevOps and Continuous Integration and Continuous Deployment (CI/CD) Design Patterns, Design Patterns for AI/ML Systems: Model Registry Pattern, Feature Store Pattern, Online/Offline Serving Patterns, Event-driven systems : Circuit Breaker, Retry, Saga, CQRS, Event Sourcing, Sidecar, API Gateways.

RECOMMENDED BOOKS

- "Head First Design Patterns" by Eric Freeman, Elisabeth Robson, Bert Bates, and Kathy Sierra.
- "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
- "Design Patterns Explained: A New Perspective on Object-Oriented Design" by Alan Shalloway and James Trott
- "Design Patterns in Java" by Steven John Metsker and William C. Wake
- "Design Patterns in Python" by Rahul Verma
- "Modern C++ Design: Generic Programming and Design Patterns Applied" by Andrei Alexandrescu

---

COURSE OUTCOMES

After completion of the course students would be able to:

CO1. **Identify** and classify design patterns based on their purpose and characteristics.
CO2. **Implement** design patterns using appropriate programming languages and frameworks.
CO3. **Analyze** software design problems and select appropriate design patterns to address them.
CO4. **Understand** and adhere to best practices when utilizing design patterns in software development.
CO5. **Evaluate** the effectiveness and efficiency of design pattern implementations in software projects.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1		3	3	3	3	1	1	1	1	1	1	3	3	3
CO2	3	3	3	3	3	1	1	1	1	1	1	3	3	3
CO3	3	3	3	3	3	2	1	1	1	1	1	3	3	3
CO4	3	3	2	3	3	1	2	1	1	1	1	3	3	3
CO5	3	3	3	3	3	1	1	1	1	1	2	3	3	3

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DATA SCIENCE LAB

29242206

List of Experiments

1. Perform Creation, indexing, slicing, concatenation and repetition operations on Python built-in data types: Strings, List, Tuples, Dictionary, Set
2. Solve problems using decision and looping statements.
3. Apply Python built-in data types: Strings, List, Tuples, Dictionary, Set and their methods to solve any given problem
4. Handle numerical operations using math and random number functions.
5. Manipulation of NumPy arrays- Indexing, Slicing, Reshaping, Joining and Splitting.
6. Computation on NumPy arrays using Universal Functions and Mathematical methods.
7. Import a CSV file and perform various Statistical and Comparison operations on rows/columns.
8. Create Pandas Series and DataFrame from various inputs.
9. Import any CSV file to Pandas DataFrame and perform the following:
    1. Visualize the first and last 10 records
    2. Get the shape, index and column details
    3. Select/Delete the records(rows)/columns based on conditions.
    4. Perform ranking and sorting operations.
    5. Do required statistical operations on the given columns.
    6. Find the count and uniqueness of the given categorical values.
    7. Rename single/multiple columns.
10. Import any CSV file to Pandas DataFrame and perform the following:
    1. Handle missing data by detecting and dropping/ filling missing values.
    2. Transform data using different methods.
    3. Detect and filter outliers.
    4. Perform Vectorized String operations on Pandas Series.
    5. Visualize data using Line Plots, Bar Plots, Histograms, Density Plots and Scatter Plots.
11. Use the scikit-learn package in python to implement the regression model and its related methods.

Course Outcomes (COs):

CO1: Apply fundamental Python programming constructs such as data types, control structures, and functions to design ethical and efficient solutions for real-life problems.

CO2: Analyze and process structured and unstructured data using Python libraries like NumPy and Pandas to derive meaningful insights while considering societal relevance and responsible data handling.

CO3: Develop real world data science applications using Python.

Course Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3			2	2		2		2	2		3	2
CO2	3	3		2	2	2		2		2	2		3	3
CO3	3	2	3	2	2	2			3	3	2	2	3	3

(1 - Slightly; 2 - Moderately; 3 - Substantially)

माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर (म.प्र.), भारत
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR (M.P.), INDIA
Deemed to be University
(Declared under Distinct Category by Ministry of Education, Government of India)
NAAC ACCREDITED WITH A++ GRADE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DESIGN AND ANALYSIS OF ALGORITHMS LAB

29242207

List of Experiments

1. WAP to implement the following using array as data structure and analyze its time Complexity.
   a. Insertion sort b. Selection sort c. Bubble sort d. Quick sort e. Bucket sort f. Radix sort g. Heap sort h. Merge sort
2. WAP to implement Linear and Binary Search and analyze its time complexity.
3. WAP to implement Matrix Chain Multiplication and analyze its time complexity.
4. WAP to implement Longest Common Subsequence Problem and analyze its time Complexity.
5. WAP to implement Optimal Binary Search Tree Problem and analyze its time complexity.
6. WAP to implement Huffman Coding and analyze its time complexity.
7. WAP to implement Dijkstra's Algorithm and analyze its time complexity. 8. WAP to implement Bellman Ford Algorithm and analyze its time complexity.
8. WAP to implement DFS and BFS and analyze their time complexities. 10. WAP to Implement 0/1 knapsack using dynamic programming.
9. Implement greedy-based approximation algorithms for problems such as Vertex Cover or Knapsack and compute their approximation ratios and performance guarantees.
10. Implement Las Vegas and Monte Carlo algorithms such as Randomized Quicksort, Randomized Search, and Randomized Hashing, and analyze their expected time complexities.

COURSE OUTCOMES:

CO1: **Implement and compare** various algorithmic techniques (sorting, searching, graph traversal, greedy, and dynamic programming) using appropriate data structures.

CO2: **Analyze and evaluate** the time and space complexities of algorithms through experimental execution and theoretical assessment.

CO3: Implement Las Vegas and Monte Carlo algorithms such as Randomized Quicksort, Randomized Search, and Randomized Hashing, and analyze their expected time complexities.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	2	2	1	3	3	2	1	1	1	1	1	2	2	1
CO2	2	2	2	3	3		1	1	1	1	1	2	1	2
CO3	2	3	2	3	3	2	1	1	1	1	1	1	2	2

(1 - Slightly; 2 - Moderately; 3 - Substantially)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COMPETITIVE PROGRAMMING

29242208

Course Objectives:

- To build strong foundational skills in programming logic, time–space complexity, and problem-solving techniques used in competitive programming.
- To enable students to apply data structures, algorithms, graphs, dynamic programming, and number theory concepts for solving medium to advanced-level coding problems.
- To prepare students for national and global coding contests by developing speed, accuracy, debugging skills, and consistent practice on online judge platforms.

UNIT 1:

Basics of C++/Python, input–output, loops, conditionals, functions. Introduction to time–space complexity and constraints. Arrays, strings, frequency maps, simple logic problems. Basics of STL (vector, set, map, pair). Practice: 40–50 beginner problems on LeetCode/CodeChef.

UNIT 2:

Core Data Structures & Techniques: Stacks, queues, linked lists, heaps, hash maps. Two-pointer, sliding window, prefix sums, hashing. Binary search and sorting-based logic patterns. Practice on GFG DS Track & LeetCode/CodeChef Data Structures. Target: 30–40 DS-based problems.

UNIT 3:

Graphs & Dynamic Programming, Graph basics: BFS, DFS, components, cycles. Shortest paths: Dijkstra, Bellman-Ford, MST algorithms. Recursion, memoization, DP introduction. DP topics: knapsack, LIS, LCS, grid DP.Practice: 30-40 graph & DP problems on LeetCode/CodeChef.

UNIT 4:

Advanced Algorithms & Number Theory, Sieve, modular arithmetic, combinatorics, modular inverse. Fast exponentiation, prime factorization. Greedy strategies and optimization techniques. Bit manipulation & bitmask DP. Practice: 30–40 advanced-level problems across platforms.

UNIT 5:

Contest Practice & Deployment Registration on LeetCode, CodeChef, GFG, HackerRank. Weekly contests, mock tests, debugging strategies. Editorial writing and clean coding practices. Mini-project: CP notebook/GitHub repository. Target: 30–40 problems + 6 mock contests.

Course Outcomes (COs):

Upon successful completion of this course, students will be able to:

CO1: Solve beginner-level CP problems using programming fundamentals.
CO2:Apply appropriate data structures and algorithmic techniques for intermediate problems.
CO3: Use graph algorithms and dynamic programming to solve complex problems.
CO4:Implement advanced number theory, greedy methods, and optimization in contest environments.
CO5: Participate effectively in coding contests and maintain consistent problem-solving practice.

Reference Books

1. "Competitive Programming 4" by Steven Halim & Felix Halim (2018).
2. "Guide to Competitive Programming" by Antti Laaksonen (Springer, 2017).
3. "Programming Challenges: The Competitive Programmer's Handbook" by Laaksonen (Revised Edition 2016).
4. Elements of Programming Interviews in C++" by Adnan Aziz et al. (2016).

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1													
CO2													
CO3													
CO4													
CO5													

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MACRO PROJECT-II

29242210

COURSE OBJECTIVES:

- To strengthen students' capability to design and develop full-scale software systems using intermediate-to-advanced programming, database, and networking concepts.
- To introduce students to multi-tier application development, including GUI/web interfaces, backend services, and AI-driven modules.
- To cultivate teamwork, documentation, testing practices, and deployment skills consistent with professional software development standards.

UNIT 1: ADVANCED PROGRAMMING & SOFTWARE DESIGN

Experiments

1. Modular programming and multi-file project setup
2. Implementing advanced DS (trees, graphs, hashing)
3. Applying software design principles—cohesion, coupling, modularity
4. Implementation of basic design patterns (MVC, Singleton, Factory)

Projects

1. Graph-Based Campus Navigation Tool
2. Hash-Map Based Student Lookup System
3. Modular Inventory Tracking Tool
4. Event Scheduler using Min-Heap
5. Course Prerequisite Checker
6. Event Scheduler using Min-Heap
7. Role-Based Access Control (RBAC) System
8. Path-Finder Engine
9. Mini Search Engine using Inverted Inde

UNIT 2: DATABASE SYSTEMS & FULL-STACK BACKEND DEVELOPMENT

Experiments

1. Designing normalized relational schemas (1NF–BCNF)

2. SQL implementation—joins, triggers, views, stored procedures
3. NoSQL basic CRUD operations (MongoDB / similar)
4. Transaction management & ACID demonstration

Projects

1. Library Database + Query Engine
2. College Course Registration System
3. Banking Mini-Backend with Transaction Logs
4. Hostel/Room Allocation with Constraints Handling
5. E-Commerce Product Catalog with Cart & Orders
6. Student Grievance Management System
7. Hospital Appointment Scheduling System
8. Employee Payroll Management System

UNIT 3: NETWORK PROGRAMMING, API DEVELOPMENT & INTRODUCTION TO AI SERVICES

Experiments

1. TCP/UDP socket programming
2. REST API creation (Flask/Django/FastAPI)
3. WebSockets for real-time communication
4. Intro to AI services:
    - Using NLP models for classification/sentiment analysis
    - Simple ML model integration (pretrained)
    - API-based AI inference (OpenAI/Gemini/HuggingFace-lite)

Projects

1. Real-Time Chat Application using Sockets
2. Weather & News App using Public APIs
3. AI-Powered Email Classifier (spam/ham) using NLP API
4. Client-Server File Transfer Tool
5. Mini Recommendation Engine (rule-based or ML API powered)
6. Multiplayer Game Backend
7. Voice Command Recognition Tool
8. Network Intrusion Detection Mini-System
9. Real-Time Stock Price Monitor

UNIT 4: GUI / WEB DEVELOPMENT + AI-ENABLED FEATURES

Experiments

1. Tkinter / PyQt or Java Swing for GUI
2. Event-driven programming
3. Basic dashboard and visualizations
4. Integrating AI components into UI (e.g., embedding ML predictions)

Projects

1. GUI-Based DB Query Interface
2. Student Performance Dashboard
3. AI-Augmented To-Do Manager (priority prediction)
4. Python/Java Image Classifier GUI (pretrained model)
5. Web-Based Mini-LMS with Quiz Auto-Grading
6. Smart Resume Analyzer GUI
7. Face Recognition Attendance System
8. Medical Symptoms Checker GUI
9. Smart Image Caption Generator
10. Expense Analyzer Web App
11. Voice-Assisted Notes App

UNIT 5: MACRO PROJECT INTEGRATION

Experiments

1. Requirement gathering + Writing SRS (Software Engineering alignment)
2. UML diagrams, architectural design, and documentation
3. Git workflow: branching, merging, collaborative development
4. Testing (unit, integration, system, black-box & white-box)
5. Deployment basics (local server / lightweight containers)

Projects:

1. Online Examination System with Proctoring Assistance **(AI + DB + Web)**
2. E-Commerce Platform with Recommendation Module
3. Smart Attendance System (AI face recognition)
4. Hospital Management & Scheduling System
5. Campus Navigation + Event Notification Platform
6. Secure Chat App with Spam Detection Module
7. College Fest/Event Management Portal with Chatbot
8. Cab Booking System with Route Optimization
9. Cloud-Based File Storage with Auto-Tagging (AI)
10. Hostel/Room Allotment using constraints + dashboard

11. Personal Finance Manager with AI Expense Categorization
12. Social Media Mini-Platform with Content Moderation

RECOMMENDED BOOKS:

- Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, Prentice Hall.
- Software Engineering: A Practitioner's Approach, Roger S. Pressman & Bruce R. Maxim, McGraw-Hill Education.
- Python Crash Course: A Hands-On, Project-Based Introduction to Programming, Eric Matthes.
- Database System Concepts, Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill Education
- Computer Networks, Andrew S. Tanenbaum, David J. Wetherall, Pearson.
- Artificial Intelligence: A Modern Approach, Stuart Russell & Peter Norvig, Pearson.

COURSE OUTCOMES:

Upon successful completion of this course, students will be able to:

CO1: Design and implement intermediate to advanced multi-tier applications integrating programming constructs, data structures, databases, APIs, and networking concepts.

CO2: Apply software engineering principles to build applications with proper architecture, documentation, testing, version control, and AI-enhanced functional components.

CO3: Collaborate effectively to plan, design, build, test, and deploy real-world software systems, demonstrating professional communication, teamwork, and project management skills.

Course Articulation Matrix

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	2	3	1	-	-	2	2	2	3	3	2
CO2	2	3	2	3	3	1	-	2	1	1	2	3	3	3
CO3	1	1	2	2	3	-	-	2	3	3	3	2	2	1

1 - Slightly; 2 - Moderately; 3 – Substantially

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Project Management, Economics & Financing

29242211

Course Objectives:

To provide knowledge about project attributes and planning essentials, develop project networks, make rational decisions for project completion, utilize resources effectively, and understand the basics of project finances and management.

Unit I:

Project Planning: Introduction to Project Management, Difference between Project and Production, Attributes of a Project: Time, Cost, Quality and Safety. Stakeholders of a Project, Project life cycle. Project Planning: Types of Project Plans and feasibility.

Unit II:

Project Network logic: Project Networking and work flows, Activity duration and methods of estimating activity duration – One time estimate three time estimates, Duration estimation procedure. Use of Bar Charts, Mile stone charts and networks, Network representation schemes: Activity on Arrow and Activity on Node Networks (A-o-A & A-o-N), Logic behind developing project network and simple network calculations, Critical paths and floats.

Unit III:

Decision making through networks: CPM, PERT & PDM: Use of network in Decision Making: Importance of critical path, Monitoring the progress and updating the project plan. Use of floats in Resource smoothening, Introduction to Precedence Diagramming Method (PDM), Different lag and lead relations in terms of SS(Start to Start), SF(Start to Finish), Finish to Start(FS), and Finish to Finish(FF) and composite relations.

Unit IV:

Project Cost Control: Breakeven analysis in planning stage, Direct and indirect cost, slope of direct cost curve, Total project cost and optimum duration, contracting the network for cost optimization. Escalation & Variation in prices.

Unit V:

Projects Financing: Introduction to project financing; Role of governments in financing projects, Funder and Concessionaire: Economic multiplier effects of Projects; Means of financing-public finance and private finance, Granting authority: World Bank Group, IMF,ADB, Micro and Small

Enterprises Funding Scheme (MSME), Elementary understanding of Procurement of infrastructure projects through Public Private Partnership (PPP) route, Build Operate Transfer (BOT), Build Operate Own & Transfer (BOOT); Stakeholders' perspectives, Lifecycle of PPP projects, Micro & Macro economics concepts and its application in Project Financing.

Course Outcomes: At the end of the course student will be able to

CO 1: Know the attributes of project and its different phases.
CO 2: Develop the project network based on work breakdown structure and estimation of activity durations.
CO 3: Analyze the project network and make decide the various alternates.
CO 4: Evaluate the optimum cost of project for assigned deadlines.
CO 5: Understand the different options to arrange the finances to complete it within stipulated time.

Text-Books:

1. Project Management Scheduling PERT and CPM by Dr. B.C. Punmia, K.K. Khandelawal

2. PERT & CPM Principles and Applications by L.S. Srinath, Affiliated EWP Pvt. Ltd.

3. Project Planning and Control by Albert Lester, Fourth Edition Elsevier Butterworth-Heinemann.

Reference Books:

1. A Management Guide to PERT/CPM With GERT/PDM/DCPM and Other networks by Jerome D. Wiest, Ferdinand K. Levy, Prentice Hall.

2. Project Management with CPM and PERT by Joseph J . Moder, Cecil R . Phillips, Van Nostrand Reinhold Company.

Course Articulation Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2
CO1													
CO2													
CO3													
CO4													
CO5													

1 - Slightly; 2 - Moderately; 3 – Substantially