RAMDEOBABA UNIVERSITY, NAGPUR–440013

Established by the Maharashtra Private Universities (Establishment and Regulation) Act 2023 (Mah. Act No VIII of 2024)
Formerly, Shri Ramdeobaba College of Engineering and Management, Nagpur 440013

School of Electrical and Electronics Engineering

Department of Electronics Engineering

PROGRAMME SCHEME & SYLLABI

of First Year as per National Education Policy (NEP)

(With effect from Academic Year 2025-26)

B.Tech. ELECTRONICS AND COMPUTER SCIENCE

Semester I

SN	Course Type	Code	Course	Hours/ week			Internal Evaluation (Th/Lab)	Mid Sem Exam	End Sem Exam	Continuous Evaluation	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	ESC	25EE01TP0101	Basic Electronics and Computer Fundamentals	3	2	4	20/25	30	50	25	150	3
2	BSC	25HS03TH0103	Probability and statistics	3	0	3	20	30	50	-	100	3
3	ESC	25EE01TP0102	Digital Logic Design	3	2	4	20/25	30	50	25	150	3
4	PCC	25EE01TP0103	Python Programming	3	2	4	20/25	30	50	25	150	3
5	CCA	25EE01TP0104	Design Thinking and Innovation	1	2	2	20/25	30	-	25	100	-
6	VSEC	25EE01PR0105	Computer Workshop	0	2	1	25	-	-	25	50	-
7	CCA	25HS02PR0102-18	Liberal / Performing Art Lab	0	2	1	25	-	-	25	50	-
TOTAL				**13**	**12**	**19**						

Semester II

SN	Course Type	Code	Course	Hours / week			Internal Evaluation (Th/Lab)	Mid Sem Exam	End Sem Exam	Continuous Evaluation	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	ESC	25EE01TP0201	Microcontroller and Interfacing	3	2	4	20/25	30	50	25	150	3
2	BSC	25HS03TH0213	Calculus & Linear Algebra	3	0	3	20	30	50	-	100	3
3	PCC	25EE01TP0202	Programming for Problem Solving	3	2	4	20/25	30	50	25	150	3
4	ESC	25EE01TH0203	AI and ML Essentials	3	0	3	20	30	50	-	100	3
5	ESC	25EE01TP0204	Computer Architecture and Organization	3	2	4	20/25	30	50	25	150	3
6	AEC	25HS02TP0201	English for Professional Communication	2	2	3	20/25	30	50	25	150	2
7	IKS	25HS02TH0203-1	Foundational Literature of Indian Civilization	1	0	1	20	30	--	--	50	-
8	VSEC	25EE01PR0205	Mini Project-I	0	2	1	25	-	-	25	50	-
9	CCA	25HS04PR0201	Health-Fitness-Wellbeing (HFW)	0	2	1	25	-	-	25	50	-
TOTAL				**18**	**12**	**24**						

`

Semester III

SN	Course Type	Code	Course	Hours/ week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	PCC	25EE01TP0301	Data Structures and Algorithm	3	2	4	20/25	30	50	25	150	3
2	PCC	25EE01TP0302	Fundamentals of Embedded System & IOT	4	2	5	20/25	30	50	25	150	3
3	PCC	25EE01TP0303	Machine Learning	3	2	4	20/25	30	50	25	150	3
4	VSEC	25EE01PR0304	Prompt Engineering	0	2	1	25	-	-	25	50	-
5	MDM	25EE01TH0305	MDM course-I	3	0	3	20	30	50	-	100	3
6	OE	25EEOEC01TH0306	Open Elective- I	2	0	2	50		50	-	100	2
7	BSC	25HS01TP0301	Environmental Science	1	2	2	20/25	30	-	25	100	-
TOTAL				**16**	**10**	**21**						
8#	PCC	25EE01TP0501	Operating System	3	2	4	20/25	30	50	25	150	3

#Accelerated Degree Program

Semester IV

	Course Type	Code	Course	Hours / week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	PCC	25EE01TP0401	Deep Learning	3	2	4	20/25	30	50	25	150	3
2	PCC	25EE01TP0402	Computer Network	3	2	4	20/25	30	50	25	150	3
3	PCC	25EE01TP0403	Database Management system	3	2	4	20/25	30	50	25	150	3
4	PCC	25EE01TP0404	Digital Signal Processing	3	2	4	20/25	30	50	25	150	3
5	MDM	25EE01TH0405	MDM Course-II	3	0	3	20	30	50	-	100	3
6	OE	25EEOEC01TH0406	Open Elective- II	2	0	2	50		50	-	100	2
7	VSEC	25EE01PR0407	Algorithmic Problem Solving -I	0	2	1	25	-	-	25	50	-
8	AEC	25HS04PR0401	Self Defense and Indian Martial Art	0	2	0	25	-	-	25	50	-
TOTAL				**17**	**12**	**22**						
9#	VEC	25HS02TH0602	Foundational Course in Universal Human Values	1	0	1	25	15	-	10	50	-
10#	AEC	25HS02TP0601	Business Communication	1	2	2	10/25	15	25	25	100	-

#Accelerated Degree Program

Exitoption:Award of UG Diploma with additional 8 credits			
Exit Course			
1	Android Application Development	Online/OfflineCertification Course	8
2	Python Programming		8
3	PCB Design		8

Semester V

SN	Course Type	Code	Course	Hours /week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	PCC	25EE01TP0501	Operating System	3	2	4	20/25	30	50	25	150	3
2	PEC	25EE01TP0502	Program Elective-I	3	2	4	20/25	30	50	25	150	3
3	PCC	25EE01TP0503-1/ 25EE01TP0503-2	Natural Language processing /Edge Computing	3	2	4	20/25	30	50	25	150	3
4	PCC	25EE01TP0504	Computer Vision	3	2	4	20/25	30	50	25	150	3
5	MDM	25EE01TH0505	MDM Course-III	3	0	3	20	30	50	-	100	3
6	OE	25EEOEC01TH0506	Open Elective-III	2	0	2	50		50	-	100	2
7	PRJ	25EE01PR0507	Capstone Project I	0	4	2	25	-	-	25	50	-
TOTAL				**17**	**12**	**23**						
8#	PEC	25EE10TP0701	Program Elective-III	3	2	4	20/25	30	50	25	150	3
9#	VSEC	25EE10PR0706	*Participative Learning	0	2	1	25	-	-	25	50	-

#Accelerated Degree Program

Semester VI

SN	Course Type	Code	Course	Hours/ week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	PCC	25EE01TP0601-1/ 25EE01TP0601-2	Multimodal AI and Language Visual Models / Edge AI	3	2	4	20/25	30	50	25	150	3
2	PEC	25EE01TP0602	Program Elective-II	3	2	4	20/25	30	50	25	150	3
3	PCC	25EE01TP0603-1/ 25EE01TP0603-2	Applied Large Language model development and deployment / Digital VLSI Design	3	2	4	20/25	30	50	25	150	3
4	VSEC	25EE01PR0604	Algorithmic Problem solving-II	0	2	1	25	-	-	25	50	-
5	MDM	25EE01TH0605	MDM-IV	3	0	3	20	30	50	-	100	3
6	VEC	25HS02TH0601	Foundational Course in Universal Human Values	1	0	1	25	15	-	10	50	-
7	PRJ	25EE01PR0606	Capstone Project II	0	4	2	25	-	-	25	50	-
8	AEC	25HS02TP0601	Business Communication	1	2	2	10/25	15	25	25	100	-
TOTAL				**14**	**18**	**21**						
9#	PCC	25EE10TH0801	Information Security and Cryptography	3	0	3	20	30	50	-	100	3
10#	VEC	25EE10TH0804	Cyber Laws and ethics in IT	2	0	2	20	30	50	-	100	3

#Accelerated Degree Program

Exit option: Award of BSc Degree with additional 8 credits			
Exit Course			
1	TBI/Industry/Research Internship	Online/offline Certification Course	8
2	AI & ML on Cloud Platform		8
3	HLS Design		8

Semester VII

SN	Course Type	Code	Course	Hours/week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	PCC	25EE01TP0701-1 /25EE01TP0701-2	Explainable AI / Smart Sensor Networks and Edge Gateways	3	2	4	20/25	30	50	25	150	3
2	PEC	25EE01TP0702	Program Elective-III	3	2	4	20/25	30	50	25	150	3
3	PEC	25EE01TP0703	Program Elective-IV	3	2	4	20/25	30	50	25	150	3
4	PRJ	25EE01PR0704	Capstone Project III	0	12	6	25	-	-	25	50	-
5	FP	25EE01PR0705	Internship Evaluation	0	2	0	25	-	-	25	50	-
6	VSEC	25EE01PR0706	*Participative Learning	0	2	1	25	-	-	25	50	-
TOTAL				**9**	**18**	**19**						
6#	PEC	25EE01TH0802	Program Elective-V	3	0	3	20	30	50	-	100	3
7#	IKS	25HS01TH0801	Foundational Literature of Indian Civilization	1	0	1	25	15	--	10	50	-
8#	HSSM	25EE01TH0803	Finance and Project Management	3	0	3	20	30	50	-	100	3

#Accelerated Degree Program

***Floating Credit: To be acquired before VI Semester.**

Semester VIII

SN	Course Type	Code	Course	Hours / week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	ESC	25EE01TH0801	Information Security and Cryptography	3	0	3	20	30	50	-	100	3
2	PEC	25EE01TH0802	Program Elective-V	3	0	3	20	30	50	-	100	3
3	IKS	25HS02TH0801	Foundational Literature of Indian Civilization	1	0	1	25	15	--	10	50	-
4	HSSM	25EE01TH0803	Finance and Project Management	3	0	3	20	30	50	-	100	3
5	VEC	25EE01TH0804	Cyber Laws and ethics in IT	2	0	2	20	30	50	-	100	2
TOTAL				**12**	**0**	**12**						
OR												
1	Internship	25EE01PR0804	Industry Internship / TBI Internship / Research Internship	0	24	**12**	50	-	-	50	100	-
TOTAL				**0**	**24**	**12**						

For one-year full time internship

SN	Course Type	Code	Course	Hours / week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	Internship	25EE01PR0805	Industry Internship / TBI Internship	0	46	23	50	-	-	50	100	-
2	PCC	25EE01TH0806	Information Security and Cryptography / MOOCs	3	0	3	20	30	50	-	100	3
3	HSSM	25EE01TH0803	Finance and Project Management / MOOCs	3	0	3	20	30	50	-	100	3
TOTAL				6	46	**29**						

HONORS Specialization in Industrial Automation and Smart Factory Systems

SN	Sem	Code	Course	Hours /week		C	Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P							
1	III	25EE01HT0301-1	Industrial Automation	3	0	3	20	30	50		100	3
2	IV	25EE01HT0401-1	Industrial Control Systems and Process Automation Fundamentals	3	0	3	20	30	50		100	3
3	V	25EE01HT0501-1	Power Electronics and Drives	4	0	4	20	30	50		100	3
4	VI	25EE01HT0601-1	Acquisition System Design	4	0	4	20	30	50		100	3
5	VII	25EE01HP0701-1	Project	0	8	4	25		50	25	100	
TOTAL				14	8	18						

HONORS Specialization in DevOps Engineering

SN	Sem	Code	Course	Hours /week		C	Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks
				L	P						
1	III	25EE01HT0301-2	Introduction to DevOps and Version Control	3	0	3	20	30	50		100
2	IV	25EE01HT0401-2	Continuous Integration and Continuous Delivery (CI/CD)	3	0	3	20	30	50		100
3	V	25EE01HT0501-2	Containerization and Orchestration	4	0	4	20	30	50		100
4	VI	25EE01HT0601-2	Cloud Infrastructure and Automation	4	0	4	20	30	50		100
5	VII	25EE01HP0701-2	Project	0	8	4	25		50	25	100
TOTAL				14	8	18					

HONORS Specialization in Research

SN	Sem	Code	Course	Hours /week		C	Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks
				L	P						
1	V	25EE01HT0501-3	Research Methodology / MOOC's	3	0	3	20	30	50	-	100
2	VI	25EE01HP0601-3	Research Project Phase -I	0	6	3	25	-	-	25	50
3	VII	25EE01HP0701-3	Research Project Phase-II	0	24	12	25	-	-	25	50
TOTAL				**3**	**30**	**18**					
				33 Hrs.							

MINOR Specialization in IoT

SN	Sem	Code	Course	Hours/ week				Continuous Evaluation	End Sem Exam	Total Marks	ESE duration(Hrs.)
				L	**T**	**P**	**C**				
1	III	25EE01MT0301	IoT fundamentals	3	0	0	3	50	50	100	3
2	IV	25EE01MT0401	Sensor Interfacing with Arduino and ESP8266	3	0	0	3	50	50	100	3
3	V	25EE01MT0501	Cloud Computing Using Raspberry Pi	3	1	0	4	50	50	100	3
4	VI	25EE01MT0601	Data Management and Analytics for IoT	3	1	0	4	50	50	100	3
5	VII	25EE01MP0701	Minor Project	0	0	8	4	50	50	100	
TOTAL				**12**	**2**	**8**	**18**				

Program Elective Basket

Specialization	Program Elective-I (V Sem)	Program Elective-II (VI Sem)	Program Elective-III (VII Sem)	Program Elective-IV (VII Sem)	Program Elective-V (VIII Sem)
Embedded Systems	25EE01TP0502-1	25EE01TP0602-1	25EE01TP0702-1	25EE01TP0703-1	25EE01TH0802-1
	Advanced Embedded System	Real-Time Operating System	Embedded Linux and Device driver	FPGA based System Design	Secure EmbeddedSystems
VLSI Design	25EE01TP0502-2	25EE01TP0602-2	25EE01TP0702-2	25EE01TP0703-2	25EE01TH0802-2
	Digital System Design	Advanced Digital System Design	System Verification	SOC Design	Design for Testability
Cloud Computing	25EE01TP0502-3	25EE01TP0602-3	25EE01TP0702-3	25EE01TP0703-3	25EE01TH0802-3
	Cloud Fundamentals	Distributed Systems	Big Data Analytics	Cloud Security	Cloud-Native Application Development

Multidisciplinary Minor [MDM] Track from EC Department

SN	Semester	Code	Course	Hours/ week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	III	*	Microelectronics Foundation	3	0	3	20	30	50	--	100	3
2	IV	*	Digital Subsystem Design	3	0	3	20	30	50	--	100	3
3	V	*	ASIC Design	3	0	3	20	30	50	--	100	3
4	VI	*	Project	0	6	3	20	30	50	--	100	3
TOTAL				**9**	**6**	**12**						

***Codes will be finalized by EC Department**

Multidisciplinary Minor [MDM] Track from CSE Computer Science & Engineering

SN	Semester	Code	Course	Hours/ week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	III	*	Introduction to Web Development	3	0	3	20	30	50	--	100	3
2	IV	*	Front End Development	3	0	3	20	30	50	--	100	3
3	V	*	Backend Technologies	3	0	3	20	30	50	--	100	3
4	VI	*	Cloud Computing	3	0	3	20	30	50	--	100	3
TOTAL				**12**	**0**	**12**						

* **Codes will be finalized by CSE Department**

Multidisciplinary Minor [MDM] Track from MBA Department

SN	Semester	Code	Course	Hours/ week			Internal Evaluation (Th/Lab)	Mid Sem Exam (Th)	End Sem Exam (Th)	Continuous Evaluation (Lab)	Total Marks	ESE Duration (Hrs)
				L	P	C						
1	III	*	Introduction to Engineering Project Management	3	0	3	20	30	50	--	100	3
2	IV	*	Project Planning , Scheduling and Cost Control	3	0	3	20	30	50	--	100	3
3	V	*	Engineering Project risk, Quality and Leadership	3	0	3	20	30	50	--	100	3
4	VI	*	Advanced Project Management & Capstone for Engineers	3	0	3	20	30	50	--	100	3
TOTAL				**12**	**0**	**12**						

***Codes will be finalized by MBA Department**

Multidisciplinary Minor (MDM) Track-1: Integrated Circuit Design (IC design)

Sr. No.	Semester	Course Code	Course Name
1	III	25EE01TH0305-1	Basics of Chip Design using Verilog HDL
2	IV	25EE01TH0405-1	MIPS Processor Design and Testing
3	V	25EE01TH0505-1	Chip Verification using System Verilog
4	VI	25EE01TH0605-1	VLSI Physical Design

Multidisciplinary Minor (MDM) Track-2: IoT for Environmental Sustainability

Sr. No.	Semester	Course Code	Course Name
1	III	25EE01TH0305-2	Introduction to IoT system Design
2	IV	25EE01TH0405-2	Programming for Environmental IoT
3	V	25EE01TH0505-2	IoT Privacy and Security
4	VI	25EE01TH0605-2	Use cases of Environmental IoT

SYLLABUS OF SEMESTER I

Course Code	25EE01TP0101			
Category	Engineering Science Course			
Course Title	Basic Electronics and Computer Fundamentals			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	I

Course Outcomes

On successful completion of the course, students will be able to

1. Apply the knowledge of basic laws to analyze simple DC circuits.
2. Design and analyze simple diode and MOSFET circuits.
3. Apply foundational knowledge to analyze and compare computer system based on their specifications and operations.
4. Explain the characteristics of network topologies and describe the functions and structure of the OSI and TCP/IP models.

Syllabus

Module I:

Basic circuit elements and RLC Circuit: circuit elements resistor, inductor and capacitor, Ohm's Law and Kirchhoff's Laws; Analysis of series, parallel circuits excited by independent voltage sources; energy sources, dependent sources, star- delta transformation.

Module II:

Semiconductors and p-n junction diode: Intrinsic and extrinsic semiconductors, Dependence of Fermi level on carrier-concentration and temperature, Current flow in semiconductors, diffusion and drift, p-n junction diode, forward bias, reverse bias, Application of diode, Introduction to Bipolar Junction Transistors

Module III

MOSFET Transistors and Biasing: MOSFET construction and working principle, VI characteristics, AC/DC load line concept, Operating Point Analysis, need of biasing, biasing techniques, bias stabilization, compensation techniques, Application of MOSFET as Amplifier.

Module IV

Computer Basics: Introduction to Computer: Generation of Computer, Computer Architecture: Input /Output devices, CPU, Memory: Primary and secondary, Basic data encoding, Software: System, Application, Driver, Ports in Computer: Display, Ethernet, USB.

Module V

Network Fundamentals: Network Topologies, Types of networks: LAN, WAN, MAN, Network Standards ,OSI and TCP/IP models.

Textbooks

1. D. P. Kothari, I J Nagrath, Basic Electrical and Electronics Engineering, Second Edition, Mc Graw Hill,2020.
2. Donald Neamen "Electronic Circuits: Analysis and Design" Third Edition, McGraw-Hill Publication
3. Brookshear J. G., "Computer science: an overview", Eleventh Edition, Addison-Wesley Publishing Company; 2011.

Re**ference Books**

1. E Balagurusamy, Fundamentals of Computers, Tata Mcgraw Hill Publications
2. Adel S. Sedra, Kenneth C. Smith, Arun N. Chandorkar: Microelectronic Circuits: Theory and Applications: Seventh Edition, Oxford University Press, 2017.
3. S. K. Bhattacharya, " Basic Electrical and Electronics Engineering", Pearson Education India Publications.

LIST OF EXPERIMENTS:

1. To measure voltage and current across a resistor and validate Ohm's Law.
2. To Verify Kirchhoff's Voltage and Current Laws using series and parallel circuit configurations.
3. To Convert a given resistive network from star to delta and vice versa and analyze the circuit behavior.
4. To Determine resonance frequency in a series RLC circuit and observe voltage variations.
5. Study the forward and reverse bias characteristics of a diode and determine threshold voltage.
6. To design and observe output waveforms of rectifier circuits using a diode.
7. To Plot VI characteristics of an n-channel MOSFET and analyze its operating regions.
8. To design and test a MOSFET-based amplifier circuit and observe its gain.
9. Identify and learn the functionality of different computer components like CPU, RAM, and storage devices.
10. To set up and test different network topologies (star, bus, ring) and analyze data transmission performance using Packet Tracer.

Course Code	25HS03TH0103			
Category	Basic Science Course			
Course Title	Probability and statistics			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	I

Course Pre-requisite : Basics of Probability and Statistics**.**

Course Objective:

The objective of this course is to expose student to understand the basic importance fundamental principles of probability, including probability distributions, random variables, basic statistical methods used for data analysis, inferential statistics, hypothesis testing, confidence intervals, and regression analysis in computer science and Information technology.

Course Outcomes

On successful completion of the course, student shall be able to

1. Identify and differentiate between discrete and continuous random variables, and interpret probabilities obtained from standard probability distributions.
2. To analyze and interpret stochastic models, including calculating probabilities, transition probabilities, and steady-state probabilities within stochastic systems.
3. Grasp the fundamental concepts of curve fitting like regression techniques, model selection, and the use of different types of curves or functions to approximate data.
4. Understand the fundamental concept of hypothesis testing, , significance levels, p-values, and the basic logic behind hypothesis testing.
5. To apply MLE to various statistical models, such as linear regression, exponential distribution, etc.

Syllabus

Module 1 (8 hours)

Measure of central tendency, quartile, inter quartile range and outliers**,** Probability spaces, conditional probability, independence, Discrete random variables, Continuous random variables , Expectation and variances, Binomial distribution, Poisson distribution, Normal distribution and their applications.

Module 2: (8 Lectures)

Joint probability function, Introduction to stochastic process, random walk, stationary and auto regressive process, transition probability Matrix, Discrete time Markov chain and its applications in queueing problems.

Module 3: (**8 hours)**

Curve fitting by the method of least squares- fitting of straight lines, second degree parabolas and more general curves, correlation and regression – Rank correlation, Multiple regression and correlation and its application in analysis of data.

Module 4: (8 Lectures)

Sampling Distributions, Point and Interval Estimations, Testing of Hypothesis for single mean and proportion.

Module 5: (7 Lectures):

Testing of Hypothesis for difference of mean and proportion, Test for ratio of variances - Chi-square test for goodness of fit and independence of attributes, maximum likelihood estimation

Text Books:

1. M R. Spiegal , Theory and Problems of probability and statistics :,2nded :,Schaum series
2. S. Ross, A First Course in Probability, 6th Ed., Pearson Education India, 2002.

Reference Books:

1. Maurtis Kaptein, Statistics for data science, An introduction to probability, statistics and Data Analysis, Springer 2022.
2. Jay L Devore, Probability and Statistics for Engineering and sciences, 8th edition, Cenage learning.

Course Code	25EE01TP0102			
Category	Engineering Science Course			
Course Title	Digital Logic Design			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	I

Course Outcomes

Upon the completion of this course, students will demonstrate the ability to:

1. Understand Number Systems and its conversions
2. Apply various optimization techniques to minimize digital circuits.
3. Design combinational logic circuits.
4. Analyze and design asynchronous and synchronous sequential circuits.

Syllabus

Module-I

Basics of Digital Electronics: Motivation for digital systems: Number Systems and arithmetic's, Representation of Signed Numbers, Boolean algebra, Logic gates, SOP, POS, Minimization of Switching functions using Karnaugh-maps.

Module-II

Timing issues in Digital Circuit: Fan-In, Fan-Out, Propagation Delay, Power Dissipation, Noise Margin

Module-III

Combinational Circuit Design: Adders, Subtractors, Multiplexer,,De-multiplexers, Encoders, Decoders, Code Converters, Comparators.

Module-IV

Sequential Circuit Design-I: Storage elements, Flip-flops and latches: D, T, JK, SR flip-flops: level triggered, edge triggered, Master Slave flip-flop, flip flop conversion, timing analysis.

Module-V

Sequential circuit Design-II: Design of asynchronous and synchronous counters, Registers & Shift registers, Application of shift register: Ring counter, Johnson counter.

Module-VI

Design of synchronous sequential circuit using Mealy model and Moore model

Textbooks:

1. D.V.Hall,"DigitalCircuitsandSystems",TataMcGrawHill,1989.
2. Modern Digital Electronics: R. P Jain ,TataMcGrawHill,3rd Edition.

Reference Books:

1. Digital Logic and Computer Design: Morris Mano, PHI,3rd Edition.

LIST OF EXPERIMENTS:

1. To verify truth table of different logic gates.
2. Design basic logic gates using universal gate and verify its truth table.
3. To verify following Boolean expressions.

   Y=A+AB+A'B'
4. To implement the following arithmetic circuits using logic gates IC's
   a) Half adder
   b) Full subtractor
5. Implement the function F = Σm (1,3,5,7,8,9,11,13,15) +d (12,14) using 16:1 and 8:1 multiplexer.
6. Verify the truth table of SR, JK, T and D flip flop.
7. To study the following functions of Shift register.
   a) SIPO
   b) PIPO
   c) PISO
   d) SISO
8. Design and verify 2-bit synchronous down counter using S-R flip-flop.
9. Design and verify the functionality of a sequence detector to detect the sequence 101 using Mealy and Moore model and use J-K flop-flop to implement the design.

Course Code	25EE01TP0103			
Category	Program Core Course			
Course Title	Python Programming			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	I

Pre-requisite Knowledge:
The following foundational knowledge is expected:

- **Basic Computer Literacy:** Familiarity with operating a computer, using the internet, and basic software applications.
- **Basic Programming Concepts (Desirable):** A conceptual understanding of programming concepts like variables, data types, and control flow would be beneficial but is not strictly required.
- **English Language Proficiency:** Ability to understand and articulate concepts in English, as Python documentation and programming resources are predominantly in English.

Course Description:
This course provides a comprehensive introduction to Python programming, covering fundamental concepts and practical applications. Learners will begin with the basics of Python syntax, data types, operators, and control flow. The course progresses to object-oriented programming (OOP) principles, error and exception handling, file handling, and advanced Python programming concepts like regular expressions, modules, and libraries. The course also covers graphical user interface (GUI) development, web programming with CGI, and various Python applications, including networking, data processing, and database connectivity. The course integrates hands-on lab exercises to reinforce learning and develop practical programming skills.

Course Objectives:
Upon successful completion of this course, learners will be able to:

- Understand the core concepts of Python programming.
- Write and execute Python programs.
- Work with Python's data types, variables, operators, and control flow statements.
- Apply object-oriented programming (OOP) principles in Python.
- Implement error handling and exception handling in Python.
- Perform file input and output operations in Python.
- Use regular expressions for pattern matching and text processing.
- Create and use Python modules and libraries.
- Develop basic GUI applications.
- Build simple web applications using CGI.

- Apply Python for networking, data processing, and database applications.

Syllabus:

Unit 1: Introduction to Python Programming

1.1: Python Fundamentals

1.1.1: Introduction to Python
- 1.1.1.1: What is Python?
- 1.1.1.2: Python's Popularity and Use Cases

1.1.2: Python Syntax and Structure
- 1.1.2.1: Python's Indentation and Block Structure
- 1.1.2.2: Writing Your First Python Program

1.1.3: Writing and Executing Python Programs
- 1.1.3.1: Using Python Interactive Mode (REPL)
- 1.1.3.2: Running Python Scripts from Command Line

1.2: Python Basics

1.2.1: Variables and Data Types
- 1.2.1.1: Understanding Variables and Naming Conventions
- 1.2.1.2: Numeric Data Types (int, float)
- 1.2.1.3: Text Data Type (str)

1.2.2: Operators and Expressions
- 1.2.2.1: Arithmetic Operators
- 1.2.2.2: Comparison Operators
- 1.2.2.3: Logical Operators

1.2.3: Input and Output Operations
- 1.2.3.1: Using input() for User Input
- 1.2.3.2: Printing Output with print()

Lab Exercises -

Hello, World! Program

Write a Python program that prints "Hello, World!" to the console.

Interactive Mode Basics

Use Python's interactive mode to perform basic arithmetic operations like addition, subtraction, multiplication, and division.

User Input and Display

Create a Python script that takes user input for their name and displays a personalized greeting.

Calculate Rectangle Area

Write a Python program that calculates and prints the area of a rectangle. Prompt the user for the length and width.

Temperature Conversion

Create a Python script that converts a temperature from Fahrenheit to Celsius. Prompt the user for the temperature in Fahrenheit and display the result in Celsius.

Unit 2: Control Flow and Loops

2.1: Control Statements

2.1.1: Conditional Statements (if, elif, else)
- 2.1.1.1: Simple if Statements
- 2.1.1.2: elif for Multiple Conditions

2.1.2: Logical Operators and Conditions
- 2.1.2.1: Using Logical AND, OR, NOT
- 2.1.2.2: Complex Conditions

2.1.3: Switch-Case (if-elif-else)
2.1.3.1: Implementing Switch-Like Behavior

2.2: Loops

2.2.1: While Loops
2.2.1.1: Using while Loops for Iteration
2.2.1.2: Controlling Loops with break and continue
2.2.2: For Loops
2.2.2.1: Iterating Over Sequences (Lists, Strings)
2.2.2.2: Using range() for Controlled Iteration
2.2.3: Loop Control Statements (break, continue)
2.2.3.1: Breaking Out of a Loop
2.2.3.2: Skipping Iterations with continue

Lab Exercises -

Even or Odd Checker

Implement a Python program that checks if a given number is even or odd and prints the result.

Largest among Three Numbers

Write a Python script that finds and prints the largest among three numbers entered by the user.

Factorial Calculation with a While Loop

Create a Python program that calculates and prints the factorial of a number entered by the user using a while loop.

Fibonacci Sequence with For Loop

Use a for loop to generate and print the Fibonacci sequence up to a specified number of terms. Prompt the user for the number of terms.

Sum of Prime Numbers

Write a Python program that calculates and prints the sum of all prime numbers in a given range. Prompt the user for the range.

Unit 3: Object-Oriented Programming (OOP) in Python

3.1: OOP Principles

3.1.1: Classes and Objects
3.1.1.1: Defining Classes and Objects
3.1.1.2: Constructors and Instance Variables
3.1.2: Inheritance and Polymorphism
3.1.2.1: Creating Subclasses
3.1.2.2: Overriding Methods
3.1.3: Encapsulation and Abstraction
3.1.3.1: Access Modifiers (public, private, protected)
3.1.3.2: Achieving Abstraction through Interfaces

3.2: Advanced OOP Concepts

3.2.1: Constructors and Destructors
3.2.1.1: Parameterized Constructors
3.2.1.2: Destructor in Python
3.2.2: Method Overloading and Overriding
3.2.2.1: Overloading Methods
3.2.2.2: Overriding Methods with super()
3.2.3: Class Variables and Instance Variables
3.2.3.1: Understanding Class Variables
3.2.3.2: Using Instance Variables

Lab Exercises -

Simple Calculator Class

Define a Python class for a simple calculator that has methods for addition and subtraction. Allow the user to perform calculations using objects of this class.

Class Inheritance Hierarchy

Create a class hierarchy with a base class and two derived classes. Demonstrate inheritance by accessing attributes and methods of each class.

Method Overriding

Implement method overriding in a Python class. Create a base class with a method and override it in a derived class.

Encapsulation Demonstration

Use encapsulation to restrict access to class attributes. Create a class with private attributes and demonstrate encapsulation principles.

Abstract Geometric Shape Class

Create an abstract class representing a geometric shape with abstract methods like area and perimeter. Define derived classes (e.g., Circle, Rectangle) to implement these methods.

Unit 4: Error Handling and Exception Handling

4.1: Exception Handling

4.1.1: Introduction to Exceptions

4.1.1.1: Understanding Exceptions in Python

4.1.1.2: Common Built-in Exceptions

4.1.2: Handling Exceptions with try and except

4.1.2.1: Using try-except Blocks

4.1.2.2: Handling Multiple Exceptions

4.1.3: Custom Exceptions

4.1.3.1: Creating Custom Exception Classes

4.1.3.2: Raising Exceptions

4.2: File Handling (I/O)

4.2.1: Reading and Writing Files

4.2.1.1: Opening and Closing Files

4.2.1.2: Reading and Writing Text Files

4.2.2: Working with Text and Binary Files

4.2.2.1: Reading and Writing Binary Files

4.2.2.2: Text Encoding and Decoding

4.2.3: File Handling Best Practices

4.2.3.1: Using 'with' Statements

4.2.3.2: Error Handling in File Operations

Lab Exercises -

Custom Exception Handling

Write a Python program that raises and handles a custom exception. Define a custom exception class and demonstrate its usage.

File Exception Handling

Implement a function that reads data from a file and handles file-related exceptions such as FileNotFoundError and PermissionError.

Division by Zero Handling

Create a Python program that simulates division by zero and handles the ZeroDivisionError exception gracefully.

File Handling with Multiple Exceptions

Modify a file reading program to handle both FileNotFoundError and PermissionError exceptions.

Finally Block Usage

Develop a program that demonstrates the use of the finally block in exception handling. Open a file and ensure it is properly closed even if exceptions occur.

Unit 5: Advanced Python Programming

5.1: Regular Expressions

- 5.1.1: Introduction to Regular Expressions
  - 5.1.1.1: What are Regular Expressions?
  - 5.1.1.2: Use Cases for Regular Expressions
- 5.1.2: Pattern Matching and Text Processing
  - 5.1.2.1: Matching Patterns with re Module
  - 5.1.2.2: Extracting Data from Text
- 5.1.3: Regex in Python
  - 5.1.3.1: Using Regular Expressions in Python
  - 5.1.3.2: Regex Functions and Methods

5.2: Modules and Libraries

- 5.2.1: Creating and Using Modules
  - 5.2.1.1: Writing Your Own Modules
  - 5.2.1.2: Importing Modules
- 5.2.2: Standard Library Modules
  - 5.2.2.1: Exploring Built-in Modules (math, datetime)
  - 5.2.2.2: Working with OS and sys Modules
- 5.2.3: Third-party Libraries and Packages
  - 5.2.3.1: Using pip for Package Installation
  - 5.2.3.2: Popular Third-party Libraries (requests, pandas)

Lab Exercises -

Email Validation with Regular Expressions

Write a Python program that validates email addresses using regular expressions. Prompt the user for an email address and validate it.

Phone Number Extraction

Create a Python script that extracts phone numbers from a given text using regular expressions.

Custom Python Module

Develop a Python module with functions to perform basic arithmetic operations. Import this module into another script and use its functions.

Exploring Built-in Modules

Explore Python's built-in modules like math and datetime. Use them to perform mathematical operations and work with date and time.

Third-Party Library Usage

Install and use a third-party library (e.g., requests) to fetch data from a web API. Retrieve data and display it in your Python script.

Unit 6: Graphical User Interfaces (GUI) and Web Programming

6.1: GUI Development

- 6.1.1: Introduction to GUI Programming
  - 6.1.1.1: GUI vs. Command Line Interfaces
  - 6.1.1.2: GUI Frameworks in Python
- 6.1.2: Widgets and Event Handling
  - 6.1.2.1: Creating Widgets (Buttons, Labels)
  - 6.1.2.2: Handling User Events (Clicks, Input)

6.2: Web Programming with CGI

6.2.1: Introduction to CGI

6.2.1.1: What is CGI and Its Purpose

6.2.1.2: CGI in Web Development

6.2.2: Handling HTTP Requests

6.2.2.1: Receiving and Processing Requests

6.2.2.2: Generating HTTP Responses

6.2.3: Building Interactive Web Applications

6.2.3.1: Form Handling with CGI

6.2.3.2: Implementing Data Processing

Lab Exercises -

Simple tkinter GUI

Design a simple tkinter GUI application with buttons and labels. Implement functionality to update labels when buttons are clicked.

GUI Button Actions

Create a Python program that responds to user button clicks in a tkinter GUI. Perform actions like displaying messages when buttons are clicked.

HTML Form Handling

Build an HTML web form with input fields and a submit button. Create a Python CGI script to handle form submissions and display the entered data.

Form Data Validation

Extend the previous exercise to validate form data in the Python CGI script. Check for required fields and display validation messages.

Dynamic Web Content with CGI

Implement a Python CGI script that generates dynamic web content based on user requests. Create a simple web application that displays different content based on URL parameters.

Unit 7: Python Applications

7.1: Networking and Serialization

7.1.1: Networking Basics in Python

7.1.1.1: Introduction to Networking Protocols

7.1.1.2: Creating Client and Server Sockets

7.1.2: Socket Programming

7.1.2.1: Building Networked Applications

7.1.2.2: Data Transfer and Communication

7.1.3: Serialization (JSON and Pickle)

7.1.3.1: Serialization Overview

7.1.3.2: JSON and Pickle for Data Serialization

7.2: Data Processing and Analysis

7.2.1: Introduction to NumPy and Pandas

7.2.1.1: What is NumPy?

7.2.1.2: Overview of Pandas

7.2.2: Data Manipulation with NumPy

7.2.2.1: Creating NumPy Arrays

7.2.2.2: Array Operations and Manipulation

7.2.3: Data Analysis with Pandas

7.2.3.1: Working with DataFrames

7.2.3.2: Data Cleaning and Exploration

7.3: Database Applications

7.3.1: Database Connectivity in Python

7.3.1.1: Database Management Systems (DBMS)

7.3.1.2: Connecting to Databases

7.3.2: SQL Queries and Database Operations

7.3.2.1: Structured Query Language (SQL)

7.3.2.2: Executing SQL Queries from Python

7.3.3: Building Database-driven Applications

7.3.3.1: Integrating Python with Databases

7.3.3.2: CRUD Operations in Database Applications

Lab Exercises -

Client-Server Communication

Create a simple client-server program using socket programming in Python. Implement basic communication between the client and server.

Enhanced Client-Server Communication

Extend the client-server program to support data transfer between the client and server. Implement sending and receiving data between the two.

JSON Serialization and Deserialization

Serialize Python objects into JSON format and then deserialize them back into Python objects. Demonstrate data interchange between JSON and Python.

Pickle Serialization

Use Pickle to serialize and deserialize Python objects. Save Python objects to a file using Pickle and then load them back.

Networked Data Exchange

Develop a Python script that simulates data exchange over a network using sockets and serialization. Send and receive data between client and server.

Basic NumPy Operations

Create a NumPy array and perform basic operations like addition, subtraction, multiplication, and division.

Data Filtering and Selection

Implement data filtering and selection using NumPy arrays. Filter data based on specific conditions and criteria.

Data Analysis with Pandas

Use Pandas to read data from a CSV file and perform data analysis. Calculate statistics like mean, median, and standard deviation on the dataset.

DataFrame Manipulation

Explore DataFrame manipulation in Pandas. Sort, filter, and perform various operations on a dataset loaded into a DataFrame.

Database CRUD Operations

Connect to a SQLite database in Python and perform CRUD (Create, Read, Update, Delete) operations using SQL queries. Create a Python script that interacts with a database.

Course Summary:

This course provides a comprehensive introduction to Python programming, covering fundamental concepts and practical applications. The course covers Python basics, control flow, object-oriented programming, error handling, file handling, advanced Python programming, GUI development, web programming with

CGI, and Python applications in networking, data processing, and databases. The course is structured to provide a balance of theoretical knowledge and practical skills through hands-on lab exercises. By the end of the course, learners will be equipped with the Python programming skills necessary to develop a variety of applications.

Course Flow of Learning:
The course is structured into seven units, designed to build a progressive understanding of Python programming:

- **Unit 1: Introduction to Python Programming:** This unit introduces Python fundamentals, syntax, and basic input/output operations.
- **Unit 2: Control Flow and Loops:** This unit covers conditional statements and loop structures for controlling program execution.
- **Unit 3: Object-Oriented Programming (OOP) in Python:** This unit introduces the principles of object-oriented programming and their implementation in Python.
- **Unit 4: Error Handling and Exception Handling:** This unit covers handling exceptions and file input/output operations.
- **Unit 5: Advanced Python Programming:** This unit introduces regular expressions, modules, and libraries.
- **Unit 6: Graphical User Interfaces (GUI) and Web Programming:** This unit covers GUI development and web programming with CGI.
- **Unit 7: Python Applications:** This unit covers Python applications in networking, data processing, and database connectivity.

Course Outcomes:
Upon successful completion of this course, learners will be able to:

- Write Python programs to solve basic programming problems.
- Apply Python data structures and control flow statements.
- Develop object-oriented programs in Python.
- Handle errors and exceptions in Python programs.
- Perform file handling operations for data storage and retrieval.
- Use regular expressions for text manipulation.
- Create and use Python modules and libraries.
- Develop simple GUI and web applications.
- Apply Python for networking, data processing, and database connectivity.
- Design and implement Python-based applications.

Skills Gained:
Upon completion of this course, learners will gain the following skills:

- **Python Programming Fundamentals:** Writing and executing Python code, understanding syntax, data types, operators, and control flow.
- **Object-Oriented Programming (OOP):** Designing and implementing classes, objects, inheritance, and polymorphism.
- **Exception Handling:** Handling errors and exceptions in Python programs.
- **File Handling:** Reading from and writing to files in Python.
- **Regular Expressions:** Using regular expressions for pattern matching and text processing.
- **Modules and Libraries:** Creating and using Python modules and libraries.
- **GUI Development:** Developing basic graphical user interfaces.
- **Web Programming (CGI):** Building simple web applications with CGI.
- **Networking:** Creating client and server applications using sockets.
- **Data Processing:** Using NumPy and Pandas for data manipulation and analysis.
- **Database Connectivity:** Connecting to and interacting with databases using Python.
- **Problem-Solving:** Applying Python programming to solve real-world problems.
- **Application Development:** Building Python-based applications.

Course Modules:

Module 1: Python Fundamentals and Control Flow

- Unit 1: Introduction to Python Programming
- Unit 2: Control Flow and Loops

Module 2: Object-Oriented Programming and Error Handling

- Unit 3: Object-Oriented Programming (OOP) in Python
- Unit 4: Error Handling and Exception Handling

Module 3: Advanced Python and Applications

- Unit 5: Advanced Python Programming
- Unit 6: Graphical User Interfaces (GUI) and Web Programming
- Unit 7: Python Applications

Module 1: Python Fundamentals and Control Flow

Module 2: Object-Oriented Programming and Error Handling
Module 3: Advanced Python and Applications

Course Code	25EE01TP0104			
Category	Co-Curricular Activity			
Course Title	Design Thinking and Innovation			
Scheme & Credits	L	P	Credits	Semester
	1	2	2	I

Pre-requisite Knowledge:
A basic understanding of problem-solving, general awareness of business or societal challenges, and a willingness to learn and collaborate. No specific technical or design background is strictly necessary, but an open and curious mindset is crucial. Familiarity with common Indian contexts and user behaviors will be beneficial for case studies and application exercises.

Course Description:
This course provides a comprehensive introduction to Design Thinking, a human-centered and iterative problem-solving approach. Learners will explore the fundamental principles, key phases, and practical techniques of Design Thinking. Through case studies, discussions, and hands-on activities, participants will develop the skills to identify user needs, define problems effectively, generate innovative ideas, and create and test prototypes. This course equips learners with a valuable framework for driving innovation and addressing complex challenges across various industries.

Course Objectives:
Upon completion of this course, learners will be able to:

- Understand the core concepts, principles, and benefits of Design Thinking.
- Identify and define problems effectively using various Design Thinking techniques.
- Develop empathy for users and stakeholders to uncover their needs and pain points.
- Generate a wide range of creative ideas through diverse ideation methods.
- Evaluate and prioritize ideas based on feasibility and impact.
- Create low-fidelity prototypes to visualize and test potential solutions.
- Apply different testing methods to gather user feedback and iterate on designs.
- Communicate ideas effectively through storytelling and presentations.
- Recognize the applications of Design Thinking in various contexts.

Course Summary:
This course begins by establishing the foundational understanding of Design Thinking, its importance, and its various stages. It then delves into the crucial initial phases of defining the problem, emphasizing empathy and stakeholder analysis. The course progresses to ideation, where learners explore numerous techniques for generating creative solutions, followed by methods for evaluating and prioritizing these ideas. Finally, it covers the essential stages of prototyping and testing, highlighting the iterative nature of Design Thinking and the importance of user feedback. Throughout the course, practical application through case studies and exercises is emphasized.

Course Flow of Learning:

1. **Introduction to Design Thinking:** Understanding the what, why, and how of Design Thinking, its mindset, and key principles.
2. **Problem Definition:** Learning to identify and clearly articulate problems by understanding user

needs and stakeholder perspectives.

3. **Empathizing and Identifying Real Problems:** Deep diving into user observation and employing empathetic design techniques to uncover underlying issues.
4. **Ideation:** Exploring a range of creative techniques to generate a multitude of potential solutions.
5. **Prioritization:** Learning methods to evaluate and select the most promising ideas based on various criteria.
6. **Prototyping and Testing:** Creating models and MVPs to test ideas with users and gather feedback for iteration.
7. **Idea Pitching:** Developing effective storytelling and presentation skills to communicate design concepts.

Course Outcomes:

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

- Apply the Design Thinking framework to solve real-world problems.
- Conduct effective user research and develop empathy maps and personas.
- Formulate clear and concise problem statements.
- Facilitate and participate in brainstorming and ideation sessions.
- Utilize various techniques for evaluating and prioritizing ideas.
- Create basic prototypes to test design concepts.
- Gather and analyze user feedback to refine solutions.
- Present design ideas persuasively.

Skills Gained:

- Problem Identification and Definition
- Empathy Development
- User Research
- Stakeholder Analysis
- Brainstorming and Ideation
- Creative Thinking
- Idea Evaluation and Prioritization
- Prototyping (basic)
- Testing and Feedback Gathering
- Communication and Presentation
- Collaboration

Course Modules:

Module 1: Understanding the Foundation of Design Thinking

- Unit 1: Introduction and Background of Design Thinking
- Unit 2: Getting started with Problem Statement (Initial stages of understanding the problem)

Module 2: Deep Dive into Problem Understanding and Ideation

- Unit 2: Getting started with Problem Statement (In-depth problem analysis)
- Unit 3: Identifying real problem
- Unit 4: Deliver on Big Idea: Ideation and Prioritization

Module 3: Bringing Ideas to Life and Validation

- Unit 5: Model creation and Idea pitching

Syllabus:

Unit 1. Introduction and Background of Design Thinking

Introduction, Knowing the team and course, The strategy of Innovation in design thinking. Why is Design Thinking required? Industries in Design Thinking, what is Design Thinking? Design thinking is a way of thinking. Design Thinking Tips and Anecdotes, Design Thinking Mindset Design thinking vs Scientific approach, Analysis vs. Synthesis in design thinking, Divergent Thinking vs. Convergent Thinking in design thinking, key phases of the design thinking process, Case Studies on Design thinking, Fundamentals of Design Thinking, Stages of Design Thinking, Design Thinking Skills, The process of Design Thinking, Design thinking framework, Why Design Thinking Works, Incorporating design thinking into your work, Limitations of a design thinking process, Advantages of the design thinking approach, Examples of design thinking success, Planning a Design Thinking Project, Benefits of the design thinking approach, Applications of design thinking.

Unit 2. Getting started with Problem Statement

Introduction What is a Problem Statement? Initial questions, Case Studies, Problem Clarification, Role of the Stakeholders, Activities for Problem Clarification, Point-Of-View (POV) in Design Thinking, Empathy Map, Understand and Define Problem Statement, Problem Analysis, Root Cause analysis tools, Defining Metrics, Persona Identification, User Personas, Stakeholder Map, who are Stakeholders? What is their role? Stakeholders of Few Companies, Creating a Stakeholder Map, Example Scenarios, Dos and Don'ts During Problem statement identification, Reformulation of the Problem.

Unit 3. Identifying real problem

Introduction, what is a Design Thinking Problem Statement? steps to create a design thinking problem statement, Learn Why and How to Focus on User Problem, Observation Phase, The Power to Observe, Useful Instrument for Observation Sessions, Tips for Observing, Practice problem Identification using Empathetic Design, Methods for Empathetic Design, Inquiry Vs. Observation, Learn As -is-state, Practice problem Identification using As-is-state long answer, Point-of-View phase for defining a problem, Characterizing the target group, Top of Form, Description of customer needs.

Unit 4. Deliver on Big Idea: Ideation and Prioritization

Introduction, Ideation Phase, The Ideation Funnel, Divergent and Convergent Thinking, Techniques for Clearing the Mind, The creative process and creative principles, Understanding Creativity, Creative Principles, Creativity Techniques, Brainstorming, Example of a Brainstorming Session, Mind Mapping, SCAMPER, Random Word Association, Rapid Prototyping / Design Sprint, Learn various Idea Generation Techniques, Creativity and Idea Generation, Common Techniques, Reverse Brainstorming, Role Playing, Analogies, Real-life Examples, Forced Connection, Visual Thinking, Storytelling, Practice Ideation, Overcoming Creative Blocks, Unrealistic Expectations, The Right Way to Take Risks, Examples of Creative Thinking, Evaluation of Ideas, Criteria / Metrics for Evaluation, Testing the Ideas, Refining and Selecting the Best Ideas, Transparency in the Evaluation Process, Pass/Fail Method, Idea Evaluation Matrix, SWOT Analysis, Criteria for Prioritization, Idea Prioritization Techniques -

a. Dot Voting
b. Impact vs. feasibility matrix
c. Weighted Scoring
d. Affinity Mapping
e. Cost-Benefit Analysis (CBA)
f. MoSCoW Method

g. Kano Model
h. Eisenhower Matrix

Implementing Prioritization, Re-Prioritization, Learn To-be Scenario, To-Be Scenario Map, use a to-do list to plan, How to Write an effective to-do List, Using Technology for to-do lists, Strategies for practicing prioritization, Urgent vs. Important Matrix, ABC Analysis, Value-based prioritization, Time-based prioritization, The Pomodoro Technique, Time-Value Prioritization Funnel, Pareto Analysis, Time-blocking, Tips for maintaining focus and staying motivated when prioritizing tasks, How to adjust priorities as circumstances change.

Unit 5. Model creation and Idea pitching

Introduction, Prototyping Phase, Learn Wireframe and Model creation, Introduction to Minimum Viable Product, Benefits Of MVP Development, What is expected from Minimum Viable Product, Practice Wireframe creation, End to end connecting from Problem to MVP, Testing Phase, Types of Testing, White and Black Box Testing, Unit Testing, Integration testing, Unit Testing Vs Integration Testing, System Testing, System Testing Vs Functional Testing, Acceptance Testing, User Acceptance Testing (UAT), Operational Acceptance Testing (OAT), Contract Acceptance Testing, Alpha/Beta Testing, Performance Testing, Security Testing, Usability Testing, Testing Techniques, Test Cases and Test Scripts, Bug Tracking and Reporting, User Feedback and Testing, Continuous Testing, Challenges and Best Practices, Tips for Interviews and Surveys, Kano Model, Desirability testing, Storytelling and Idea Presentation.

Reference Books:

1. "Design Thinking: Integrating Innovation, Customer Experience, and Brand Value" by Thomas Lockwood
2. "The Design Thinking Playbook: Mindful Digital Transformation of Teams, Products, Services, Businesses and Ecosystems" by Michael Lewrick, Patrick Link, Larry Leifer
3. "Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days" by Jake Knapp, John Zeratsky, Braden Kowitz
4. "Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation" by Tim Brown
5. "Creative Confidence: Unleashing the Creative Potential Within Us All" by Tom Kelley, David Kelley
6. "The Art of Innovation: Lessons in Creativity from IDEO, America's Leading Design Firm" by Tom Kelley
7. "The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses" by Eric Ries

List of Experiments

EXP NO 1:

(a) Classify problem as a technical and Non technical

(b) Extract meaningful keyword from problem statements.

EXP NO. 2 :

(a) Problem statement on Root cause analysis, Empathy map.

EXP NO 3:

(a) Problem statement based on observation technique that accepts two lists >

As -is and To-be .Then print a Comparision table.

EXP NO 4:

(a) Problem statement based on creative technique SCAMPER.

EXP NO 5:

(a) Problem statements based on evaluating ideas, prioritization techniques.

EXP NO 6:

(a) Build a console based prototype of “Students notes sharing app” with basic

Menu options.

EXP NO 7:

(a) Write a python program that takes problem +solution as input and auto-

generates a basic pitch outline.

EXP NO 8 :

(a) Python scripts that collects user rating (1-5) stores them in a file and

calculates the average score. This simulates a simple feedback collection system.

EXP NO 9:

(a) Pitching ideas

Course Code	25EE01PR0105			
Category	Vocational and Skill Enhancement Course			
Course Title	Computer workshop			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Acquire a working knowledge of Linux fundamentals and Linux distributions.
2. Apply knowledge to comprehend system configurations and Linux graphical interfaces.
3. Independently perform fundamental command line operations in Linux.
4. Effectively employ common Linux applications for specific tasks and functionalities

Practical's based on:

The Linux Foundation: Linux Philosophy and Concepts, Linux Basics and System Startup.

Graphical Interface, System Configuration from the Graphical Interface, Common Applications, Command Line Operations, Finding Linux Documentation.

Processes, File Operations, Text Editors, User Environment, Manipulating Text, Network Operations.

The Bash Shell and bash Scripting, Introduction, Features and Capabilities, Syntax, Constructs.

Printing, Local Security Principles, Understanding Linux Security, root Privileges, sudo, Process Isolation, Limiting Hardware Access and Keeping Systems Current, Working with Passwords, Securing the Boot Process and Hardware Resources.

Remote access and managing processes through remote login

Text books

1) Linux BIBLE, Christopher Negus, Tenth Edition, Wiley 2020.

2) Linux for Beginners: An Introduction to the Linux Operating System and Command Line, Jason Cannon, O'Reilly, 2014.

Course Title: Liberal / Performing Arts

Course Code	Course Name	Sem	Hours/ week	Credits	Continuous Evaluation Max. marks
25HS02PR0102-02	Fundamentals of Indian Classical Dance: Bharatnatayam	I	2	1	50
25HS02PR0102-03	Fundamentals of Indian Classical Dance: Kathak	I	2	1	50
25HS02PR0102-04	Introduction to Digital Photography	I	2	1	50
25HS02PR0102-05	Introduction to Basic Japanese Language	I	2	1	50
25HS02PR0102-06	Art of Theatre	I	2	1	50
25HS02PR0102-07	Introduction to French Language	I	2	1	50
25HS02PR0102-08	Introduction to Spanish Language	I	2	1	50
25HS02PR0102-09	Art of Painting	I	2	1	50
25HS02PR0102-10	Art of Drawing	I	2	1	50
25HS02PR0102-11	Nature Camp	I	2	1	50
25HS02PR0102-12	Developing Self-awareness	I	2	1	50
25HS02PR0102-13	Art of Poetry	I	2	1	50
25HS02PR0102-14	Creative and content writing	I	2	1	50
25HS02PR0102-15	Science of life through Bhagwad Gita	I	2	1	50
25HS02PR0102-16	Sanskrit Sambhashan-Spoken Sanskrit	I	2	1	50
25HS02PR0102-17	Kirtan Kala	I	2	1	50
25HS02PR0102-18	Introduction to German Language and culture	I	2	1	50
25HS04PR0102-1	Adventure Sports	I	2	1	50
25HS04PR0102-2	Introduction to Défense Forces & Obstacle Training	I	2	1	50
25HS04PR0102-3	First Aid & Disaster Management	I	2	1	50
25HS04PR0102-4	Basic Nutritional Course	I	2	1	50

25HS04PR0102-5	Stress Management Through Yoga & Meditation	I	2	1	50

Course Code	25HS02PR0102-02			
Category	Co-Curricular Activity			
Course Title	Fundamentals of Indian Classical Dance: Bharatnatayam			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Understand the importance of dance and Bharatnataym as an Indian dance form
2. Develop skills to perform the dance form at its basic level.
3. Evaluate their strengths and interest to take bridge course to give Pratham (1st level formal exam of Bharatnatayam).

Syllabus

1.Orientation in Bharatnatayam
2.Tattu Adavu till 8, NaattaAdavu 4 Steps, PakkaAdavu 1 step, Metta Adavu 1 Step, Kuditta Metta Adavu 4 Steps
3. Practice sessions
4.Tatta Kuditta Adavu (Metta), Tatta Kuditta Adavu (Metta) 2 Steps, Tirmanam Adavu 3 Steps, Kattu Adav - 3 Steps, Kattu Adav - 3 Steps
5. Practice sessions
6. Tiramanam (front) 3 Steps, Repeat of Tiramanam (Overhead) 3 Steps
7.Practice sessions
8.Final practice sessions and performances.

Recommended reading

1.Introduction to Bharata's Natyasastra, Adya Rangacharya, 2011
2.The Natyasastra and the Body in Performance: Essays on the Ancient Text, edited by Sreenath Nair, 2015
3.Bharatanatyam How to ... : A Step-by-step Approach to Learn the Classical Form, Eshwar Jayalakshmi, 2011

Course Code	25HS02PR0102-03			
Category	Co-Curricular Activity			
Course Title	Fundamentals of Indian Classical Dance: Kathak			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Understand the importance of dance and Kathak as an Indian dance form
2. Develop skills to perform the dance form at its basic level.
3. Evaluate their strengths and interest to take bridge course to give Prarambhik (1st level formal exam of Kathak).

Syllabus

1.Orientation in Kathak. Correct posture of kathak, Basic Movements and exercise Stepping, Chakkar of 5 count (Bhramari),
2.practice sessions of practical 1
3.Hastaks, Hastaks and Steppings, Reciting asamyukta Mudra shloka, Hastak and steppings
4.practice sessions of practical 3
5.Todas and Asamyukta hasta mudra shlok, Vandana of Shlok, 2 Todas and Vandana, Ghante Ki Tihai,
6.practice sessions of practical 5
7.2 1 Chakkardar Toda and Ginnti Ki Tihai, 2 Todas and 1 Chakkardar Toda, practice sessions
8.Final performances.

Recommended reading

1.Kathak Volume1 A "Theoretical & Practical Guide" (Kathak Dance Book), Marami Medhi & Debasish Talukdar, 2022, Anshika Publication (13 September 2022)

Course Code	25HS02PR0102-04			
Category	Co-Curricular Activity			
Course Title	Introduction to Digital Photography			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Develop an understanding of the technical aspects and aesthetics of Photography.
2. Apply the rules of digital photography for creating photographs.
3. Develop skills to enhance photographs through post processing.
4. Create a portfolio of their photographs in selected genre.

Syllabus

1. Orientation in digital photography**:** Genres, camera handling and settings
2. Rules of Composition
3. Rules of Composition: practice sessions
4. Understanding Exposure and Art of Pre-Visualization
5. Rules of Composition and Art of Pre-Visualization: practice sessions
6. Post Processing Photographs and Portfolio creation
7. Post Processing Photographs: practice sessions
8. Portfolio finalization and presentation in selected genre.

Reference material

1. Scott Kelby (2020) The Digital Photography Book: The Step-by-Step Secrets for how to Make Your Photos Look Like the Pros, Rocky Nook, USA
2. Larry Hall (2014) Digital Photography Guide: From Beginner to Intermediate: A Compilation of Important Information in Digital Photography, Speedy Publishing LLC, Newark
3. J Miotke(2010) BetterPhoto Basics: The Absolute Beginner's Guide to Taking Photos Like a Pro, AMPHOTO Books, Crown Publishing Group, USA

Course Code	25HS02PR0102-05			
Category	Co-Curricular Activity			
Course Title	Introduction to Basic Japanese Language			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Gain a brief understanding about Japan as a country and Japanese culture.
2. Develop ability to use vocabulary required for basic level communication in Japanese language.
3. Write and read the first script in Japanese language.
4. Frame simple sentences in Japanese in order to handle everyday conversations
5. Write in basic Japanese about the topics closely related to the learner.

Syllabus

1.Orientation about Japan, its language, and its culture
2.Communication Skills 1: Vocabulary for basic Japanese language
3. Practice sessions
4.Writing Skills 1: Reading and writing first script in Japanese
5.Practice sessions
6.Communication Skills 2: framing sentences
7.Practice sessions
8.Writing Skills 2: Write basic Japanese and practice

Recommended reading

1.Marugoto Starter (A1) Rikai - Course Book for Communicative Language Competences, by The Japan Foundation, Goyal Publishers & Distributors Pvt. Ltd (ISBN: 9788183078047)
2.Japanese Kana Script Practice Book – Vol. 1 Hiragana, by AmeyaPatki, Daiichi Japanese Language Solutions (ISBN: 9788194562900)

Course Code	25HS02PR0102-06			
Category	Co-Curricular Activity			
Course Title	Art of Theatre			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Understand and synthesize the working of the prominent genres of theatre across the world.
2. Apply the skill of voice and speech in theatre and public speaking
3. Apply the art of acting and also develop generic skills such as confidence, communication skills, self- responsibility, motivation, commitment, interpersonal skills, problem solving, and self-discipline.
4. Apply the skills, acquired related to technical/production aspects of theatre and also develop problem solving and interpersonal skills.

Syllabus

1. Orientation in theatre
2. Voice and Speech training
3. Voice and Speech training: practice sessions
4. Art of acting
5. Art of acting: practice sessions
6. Art of script writing
7. Art of script writing: practice sessions
8. Final performances

Reference books

1. Boleslavsky, R. (2022). Acting: The First Six Lessons (1st ed., pp. 1-92). Delhi Open Books.
2. Shakthi, C. (2017). No Drama Just Theatre(1st ed., pp. 1-171). Partridge.
3. Bruder, M., Cohn, L. M., Olnek, M., Pollack, N., Previto, R., &Zigler, S. (1986). A Practical Handbook for the Actor (1st ed.). Vinatge Books New York.

Course Code	25HS02PR0102-07			
Category	Co-Curricular Activity			
Course Title	Introduction to French Language			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Demonstrate basic knowledge about France, the culture and similarities/differences between India and France
2. Learn to use simple language structures in everyday communication.
3. Develop ability to write in basic French about themselves and others.
4. Develop ability to understand beginner level texts in French

Syllabus

List of Practicals

1. Orientation about France, the language, and culture
2. Communication Skills 1: Vocabulary building for everyday conversations
3. Practice sessions
4. Reading and writing Skills : Reading and writing simple text in French
5. Practice sessions
6. Communication Skills 2: listening comprehension
7. Practice sessions
8. Writing Skills: Write basic French and practice

Recommended reading

1. 15-minute French by Caroline Lemoine
2. Cours de Langue et de Civilisation Françaises by G. Mauger Vol. 1.1
3. Cosmopolite I by Natalie Hirschsprung, Tony Tricot

Course Code	25HS02PR0102-08			
Category	Co-Curricular Activity			
Course Title	Introduction to Spanish Language			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Demonstrate basic knowledge about Spain, the culture and similarities/differences between India and France
2. Learn to use simple language structures in everyday communication.
3. Develop ability to write in basic Spanish about themselves and others.
4. Develop ability to read and understand beginner level texts in Spanish

Syllabus

List of Practicals

1. Orientation about Spain, the language, and culture
2. Communication Skills 1: Vocabulary building for everyday conversations
3. Practice sessions
4. Reading and writing Skills: Reading and writing simple text in Spanish
6. Communication Skills 2: listening comprehension
7. Practice sessions
8.Writing Skills: Write basic Spanish and practice

Recommended reading

1. 15-Minute Spanish by Ana Bremon
2. Aula Internacional 1 by Jaime Corpas ,Eva Garcia, Agustin Garmendia.
3.Chicos Chicas Libro del Alumno by María Ángeles Palomino

Course Code	25HS02PR0102-09			
Category	Co-Curricular Activity			
Course Title	Art of Painting			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1.Become familiar with the basic methods, techniques & tools of painting.

2. Train the eye and hand to develop sense of balance, proportion and rhythm.

3. Develop the ability to observe and render simple natural forms.

4. Enjoy the challenging and nuanced process of painting.

Syllabus

1.Orientation in Painting tools & basics of lines, shapes, light, shadows and textures

2.The art of observation how to see shapes in drawing

3.Introduction Water colorhow to handle water paints

4.Introduction to acrylic colorshow to handle acrylic paints

5.Explore layering paint and capturing the quality of light with paint.

6. Create landscape painting

7.Create Abstract painting

8. Paint on Canvas (try to recreate any famous painting)

Reference material

1. Drawing made easy by Navneet Gala; 2015th edition

2. Alla Prima II Everything I Know about Painting--And More by Richard Schmid with Katie Swatland 3.Daily Painting: Paint Small and Often To Become a More Creative, Productive, and Successful Artist by Carol Marine

Course Code	25HS02PR0102-10			
Category	Co-Curricular Activity			
Course Title	Art of Drawing			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Become familiar with the basic methods, techniques & tools of drawing.
2. Train the eye and hand to develop sense of balance, proportion and rhythm.
3. Develop the ability to observe and render simple natural forms.
4. Enjoy the challenging and nuanced process of drawing.

Syllabus

1. Orientation in Drawing tools & basics of lines, shapes, light, shadows and textures
2. The art of observation how to see shapes in drawing
3. One/two-point basic linear perspective
4. Nature drawing and landscapes
5. Gestalt principles of visual composition
6. Figure drawing: structure and proportions of human body
7. Gesture drawing: expression and compositions of human figures
8. Memory drawing: an exercise to combine the techniques learnt

Reference material

1. Drawing made easy by Navneet Gala; 2015th edition
2. Perspective Made Easy (Dover Art Instruction) by Ernest R. Norling

Course Code	25HS02PR0102-11			
Category	Co-Curricular Activity			
Course Title	Nature Camp			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes

On successful completion of the course, students will be able to

1. Develop an affinity with nature by observing and understanding it marvels with guidance from experts
2. Develop an understanding of the challenges and solutions associated with nature and its conservation.

Course content

In collaboration with the Forest Department and/or a local NGO working in the field of environment conservation, this course would be conducted in 24 hours. Students will be taken to a tiger reserve in Vidarbha region or Forest fringe villages or work with an NGO from Vidarbha region working on natural resource management. The camps (for 2 days) will cover any one of the following topics as decided by the course coordinator:

1. Awareness about each element of biodiversity (camps on moths, butterflies, birds, other wildlife etc)
2. Environment management (water, forest, wildlife) – practices of Forest Department in managing a tiger reserve, and other aspects of water and forest conservation.
3. Sustainable natural resource management - initiatives by rural communities and local NGOs
4. Man-animal conflict and solutions (socio-economic and technical) – role of local communities and Forest Department
5. Traditional practices in environment conservation – role of local communities and local NGOs

Course Code	25HS02PR0102-12			
Category	Co-Curricular Activity			
Course Title	Developing Self-awareness			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course objectives:
The course aims to develop students in their personal as well as professional life by means of graphotherapy, NLP, and Neurobics

Course Outcomes:
On completion of the course, students will be able to achieve the following:

CO1: Gain foundational understanding of graphology and through self-analysis will achieve greater awareness about their strengths and weaknesses & areas for personal growth
CO2: students will be equipped with tools and techniques for continuous self-improvement, using signature analysis and graphotherapy as part of their personal development journey
CO3: understand how to use Neuro Linguistic Programming (NLP) strategies to set and achieve goals effectively, overcoming mental blocks and limiting beliefs.
CO4: Enhance ability to absorb, retain, and recall information, which can benefit academic and professional performance.

Syllabus:

Practical 1: **The Power of Handwriting (Handwriting is Brainwriting)**
Practical 2: **Know yourself through handwriting**
Practical 3: **The Role of Signature in your life**
Practical 4: **Graphotherapy to enhance yourself in all ways**
Practical 5: **Neurolinguistic Programming , S.M.A.R.T Goal**
Practical 6: **Effective Communication Model, Rapport Building and Anchor**
Practical 7: **Brain Directives & Linguistic Presuppositions**
Practical 8: **Neurobics**

Course Code	25HS02PR0102-13			
Category	Co-Curricular Activity			
Course Title	Art of Poetry			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Outcomes:
To familiarize the students with the art of poetry and develop a sense of appreciation for the art

At the end of the course the student will be able to achieve the following:
CO1: Understand the origin and development of poetry
CO2: Appreciate the art of poetry in life
CO3: Develop aesthetic sense
CO4: Develop holistic perspective to their personality

Syllabus
Practical 1: **Art of poetry – orientation**
Practical 2: **Forms of poetry – orientation**
Practical 3: **Forms of poetry – recitation**
Practical 4: **Application of poetry – orientation**
Practical 5: **Application of poetry – practical session**
Practical 6: **Poetry and aesthetics**
Practical 7: **Writing poetry – orientation**
Practical 8: **Writing poetry – writing sessions**

Reading material

I. The Art of Poetry
1. Fry, S. (2005). The ode less travelled: Unlocking the poetic mind. HarperCollins.
2. Addonizio, K., & Laux, D. (1997). The poet's companion: A guide to the pleasures of writing poetry. W.W. Norton & Company.
3. Lucy, J. (Ed.). (2001). The art of poetry. Penguin Books.

II. Understanding and Interpretation of Poetry
1. Hirsch, E. (1999). How to read a poem: And fall in love with poetry. Harcourt Brace & Company.
2. Pinsky, R. (1998). The sounds of poetry: A brief history. Farrar, Straus and Giroux.
3. Meyer, M. (2005). Poetry: An introduction. Bedford/St. Martin's.

III. Writing Poetry
1. Hugo, R. (1979). The triggering town: Lectures and essays on poetry and writing. W.W. Norton & Company.
2. Bradbury, R. (1990). Zen in the art of writing: Releasing the creative genius within you. Bantam Books.
3. Behn, R., & Twichell, C. (Eds.). (1992). The practice of poetry: Writing exercises from poets who teach. HarperCollins

Course Code	25HS02PR0102-14			
Category	Co-Curricular Activity			
Course Title	Creative and Content Writing			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course objective:
The objective of the course is to equip students with comprehensive skills in creative and content writing through experiential learning and real-world applications.

Course outcomes:
On completion of the course, student will be able to achieve the following:
CO1: Understand and apply fundamental concepts and techniques of creative writing.
CO2: Apply storytelling techniques to create engaging narratives.
CO3: Develop and implement effective SEO and digital content strategies
CO4: Create and refine content using various tools and applying diverse writing styles and formats.
CO5: Utilize digital tools to craft multimedia narratives and create a professional portfolio.

Syllabus

Creative Writing
Practical 1: **Introduction to Creative and Content Writing**
Practical 2: **Character and Story Development**
Practical 3: **Crafting Compelling Narratives**

Content Writing
Practical 4: **SEO and Digital Content Strategies**
Practical 5**: Writing for Media**
Practical 6: **Tools**

Content Creation
Practical 7: **Digital Storytelling**
Practical 8: **Creative Portfolio Launch**

Course Code	25HS02PR0102-15			
Category	Co-Curricular Activity			
Course Title	Science of life through Bhagwad Gita			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Objective

The objective of the course is to seek directions from the Bhagwad Gita to garner life skills for a successful and happy life

Course Outcome

CO1: To understand the methodology to correctly interpret and analysis the scripture
CO2: To understand the application of various teaching of the Bhagwad Gita
CO3: Use meditation and breathing techniques for healthy mind and body.

Syllabus

Practical 1: **Introduction to Bhagwad Gita - methodology**
Practical 2: **Real life application of chapter 1-3**
Practical 3: **Real life application of chapter 4-6**
Practical 4: **Real life application of chapter 7-9**
Practical 5: **Real life application of chapter 10-12**
Practical 6: **Real life application of chapter 13-15**
Practical 7: **Real life application of chapter 16-18**
Practical 8: **Meditation and breathing techniques**

Course Code	25HS02PR0102-16			
Category	Co-Curricular Activity			
Course Title	Sanskrit Sambhashan- Spoken Sanskrit			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course objectives:
The objective of the course is to enhance the communication skills of the students in Sanskrit

Course outcome
At the end of the course, the students will be able to achieve the following:
CO1: Enhanced writing skills in Sanskrit
CO2: Enhanced speaking skills in Sanskrit
CO3: Enhanced listening skills in Sanskrit
CO4: Enhanced writing skills in Sanskrit

Syllabus:

संस्कृतसम्भाषणशिबिरस्य पाठ्यक्रमः

प्रथमं दिनम्

- गीतम् - पठत संस्कृतम्.....।
- मम नाम -भवतः नाम किम्? भवत्याः नाम किम्? द्वयोः मध्ये परिचयः। परस्परं 5 जनान्।
- सः कः? सा का? तत् किम्?
- एषः, एषा, एतत्।
- अहम्, भवान्, भवती..... ...अभिनयः।
- आम्,न, वा/किम्.....अभिनयः।
- अस्ति × नास्ति.....अभिनयः।
- अत्र, तत्र, कुत्र, सर्वत्र, अन्यत्र, एकत्र - अभिनयः।
- षष्ठी - तस्य, एतस्य, कस्य, तस्याः, एतस्याः, कस्याः, मम, भवतः, भवत्याः.....अभिनयः।
  मम नासिका, भवतः नासिका, भवत्याः नासिका।
  एतत् कस्य? अङ्गानि प्रदर्श्य प्रश्नः।
- दशरथस्य..., सीतायाः..., लेखन्याः..., पुस्तकस्य...,। स्फोरकपत्रस्य (Flash Card) उपयोगः करणीयः। 'पुत्रः' 'पतिः' इत्यादीनां वाक्यपत्राणाम् (Charts) उपयोगः करणीयः।
- गीतम् - मनसा सततं स्मरणीयम्।
- आवश्यकम्, मास्तु, पर्याप्तम्, धन्यवादः, स्वागतम्।
- पूर्वनिश्चितसम्भाषणप्रदर्शनम्।
- क्रियापदानां पाठनम् -
  गच्छति। आगच्छति। पठति। लिखति। खादति।पिबति। क्रीडति। वदति। उत्तिष्ठति। उपविशति।
- गच्छामि। आगच्छामि.....।
- गच्छतु। आगच्छतु.....।
- सङ्ख्याः - (अ) 1, 2, 3, 4,.....10।
  (आ) 10, 20, 30,.....100।
- समयः - 5.00, 5.15, 5.30, 4.45 ।
- कथा - गतानुगतिको लोकः। (काचित् कथा सरलया भाषया वक्तव्या)।
- रटनाभ्यासः (पूर्वमेव लिखितानि पठितानि च कानिचित् वाक्यानि वाचनीयानि)।
- एकं वाक्यम् (प्रत्येकं छात्रः एकं वाक्यं वदेत्।)
- सूचना।
- ऐक्यमत्यः।

द्वितीयं दिनम्

- गीतम्।
- पुनस्स्मारणम्।
- शब्देषु लिङ्गभेदज्ञापनम् - यथा -सः सुधाखण्डः, सा कुल्फिका, तत् पुष्पम्।
- बहुवचनपाठनम् –
  बालकाः..., बालिकाः..., लेखन्यः..., पुस्तकानि...।
- ते, के, ताः, काः, तानि, कानि, एते, एताः, एतानि, भवन्तः, भवत्यः, वयम्। (चित्राणि उपयोक्तव्यानि।)
- वचनपरिवर्तनाभ्यासः। यथा - सः बालकः - ते बालकाः।
- अस्ति - सन्ति।
- कति?
- सप्तमी - हस्ते। उत्पीठिकायाम्। लेखन्याम्। पुस्तके। (स्फोरकपत्रस्य प्रयोगः करणीयः।)
  वाक्यपत्रस्य उपयोगेन वाक्यानि वाचनीयानि।
- कदा?
- उत्तराणां प्रश्नाः। (शिक्षकः आरम्भे उत्तरं वदेत्, अनन्तरं छात्राः तस्य प्रश्नं पृच्छेयुः।)
  यथा - रामः प्रातःकाले शालां गच्छति।
  रामः कदा शालां गच्छति?
- अद्य, श्वः, परश्वः, प्रपरश्वः, ह्यः, परह्यः, प्रपरह्यः, इदानीम्।
- गीतम्।
- गच्छन्ति। गच्छामः। गच्छन्तु।
- शिष्टाचारः - सुप्रभातम्/नमस्कारः/शुभरात्रिः/हरिः ओम्/क्षम्यताम्/चिन्ता मास्तु।
- प्रातर्विधिः - दन्तधावनम् इत्यादयः शब्दाः पाठनीयाः।
- सङ्ख्या - 1-50।
- समयः - 6.05, 6.10, 5.55, 5.50
- स्वागतसम्भाषणम्। (शिक्षकः सहशिक्षकेण सह कृत्वा प्रदर्शयेत्)
- कथा।
- रटनाभ्यासः।
- वाक्यद्वयम् (प्रत्येकम् अपि छात्रः वाक्यद्वयं वदेत्।)
- सूचनाः।
- ऐक्यमत्यः।

तृतीयं दिनम्

- गीतम्।
- पुनस्स्मारणम्।
- क्रियापदानां बहुवचनरूपाणि।
  गच्छन्ति - गच्छामः - गच्छन्तु (Chart दर्शनीयम्)
  पिबन्ति - पिबामः - पिबन्तु।
  लिखन्ति - लिखामः - लिखन्तु।
  इत्यादिपरिवर्तनाभ्यासः कारणीयः।
- द्वितीयाविभक्तिः - स्फोरकपत्राणाम् उपयोगः।
  (वाक्यपत्राणि उपयुज्य वाक्यानि वाचनीयानि।)
- कृपया ददातु - वस्तूनि प्रदर्श्य।
  शिक्षकः एकैकं वस्तु प्रदर्शयति।
  उदा. – ग्रन्थः, घटी,.....
  छात्राः - कृपया ग्रन्थं ददातु, कृपया घटीं ददातु इत्यादि वदेयुः। (स्फोरकपत्रस्य उपयोगः)
- पुरतः, पृष्ठतः, वामतः, दक्षिणतः, उपरि, अधः।
  (चित्रं दर्शनीयम्)
- इतः, ततः, ...तः, गृहतः, कुतः?
  (स्फोरकपत्राणाम् उपयोगः)
  वाक्यपत्राणि उपयुज्य वाक्यानि वाचनीयानि।
- गीतम्।
- कथम्? सम्यक्।
- शीघ्रम् × मन्दम्। उच्चैः × शनैः।
- पठनार्थम्, किमर्थम्?
- सप्तककाराः - किम्, कुत्र, कति, कदा, कुतः, कथम्, किमर्थम् (Chart प्रदर्शनीयम्)।
  एकैकम् उपयुज्य परस्परं प्रश्नाः।
- अपि।
- अस्तु।
- अहं न जानामि। - कानिचन वाक्यानि।
- भूतकालीनक्रियापदानां पाठनम्।
  गतवान् - पठितवान् - लिखितवान्।
  गतवती - पठितवती - लिखितवती।
- क्रियापदकोष्ठकस्य प्रथमपृष्ठस्य अभ्यासः।
- द्वितीयपृष्ठस्य सर्वाणि क्रियापदानि उपयुज्य छात्राः वर्तमानकाले वाक्यानि वदन्ति। (ए.व - ब.व.)
- विशिष्टक्रियापदानाम् अभ्यासः -
  करोमि - कुर्मः। करोति - कुर्वन्ति।
  ददामि - दद्मः। ददाति - ददति।
  शृणोमि – शृणुमः। शृणोति - शृण्वन्ति।
  जानामि – जानीमः। जानाति – जानन्ति।
- सम्बोधनम् - भोः !, श्रीमन् !, मान्ये !, भगिनि!, मित्र !, ...महोदय!, राम !, सीते ! इत्यादि।
- सङ्ख्या- 1-100।
- समयः - 1.00, 2.00, 3.00, 4.00।
- सम्भाषणप्रदर्शनम् (मित्रसंलापः)।
- कथा।
- वाक्यत्रयम् एकैकोऽपि छात्रः वदेत्।
- सूचना।
- ऐक्यमत्यः।

चतुर्थं दिनम्

- गीतम्।
- पुनःस्मारणम्।
- च
- अतः
- एव
- इति
- अस्मि
- यदि -तर्हि
- यथा - तथा
- तः - पर्यन्तम् (वाक्यपत्रस्य उपयोगेन वाक्यानि वाचनीयानि।)
- अद्य आरभ्य
- कृते (वाक्यपत्रस्य उपयोगः करणीयः)
- क्तवतुप्रत्ययान्तानाम् अभ्यासः
  गतवान् - पठितवान् - लिखितवान् (ए.व. पुंलिङ्गे)।
  गतवती - पठितवती - लिखितवती (ए.व. स्त्रीलिङ्गे)।
  गतवन्तः - पठितवन्तः - लिखितवन्तः (ब.व. पुंलिङ्गे)।
  गतवत्यः - पठितवत्यः - लिखितवत्यः (ब.व. स्त्रीलिङ्गे)।
- सः गतवान् - सा गतवती - लिङ्गपरिवर्तनाभ्यासः।
- अहं गतवान् - अहं गतवती - लिङ्गपरिवर्तनाभ्यासः।
- क्रियापदानां कालपरिवर्तनाभ्यासः।
  यथा - गच्छति - गतवान्, गतवती।
- गीतम्।
- विशेषपाठनम् - आसीत्, आसन्, आसम्।
- एकः, एका, एकम् - लिङ्गभेदः ज्ञापनीयः।
  (स्फोरकपत्रस्य उपयोगः)

- भोजनसम्बन्धिशब्दाः यथा - सूपः, शाकम्, इत्यादयः।
- सङ्ख्या।
- समयः।
- ॐ – सङ्ख्याक्रीडा।
- कथा।
- सम्भाषणप्रदर्शनम्।
- चत्वारि वाक्यानि।
- सूचना।
- ऐकमत्यः।

पञ्चमं दिनम्

- गीतम्।
- पुनःस्मारणम्।
- वाहनानां नामानि।
- तृतीयाविभक्तिः - दण्डेन, मापिकया, लेखन्या, पुष्पेण।
  (वाक्यपत्रस्य आधारेण वाक्यानि वाचनीयानि।)
- सह, विना।
- अद्यतन, ह्यस्तन, श्वस्तन, पूर्वतन, इदानीन्तन
- भविष्यत्कालीनक्रियापदानां पाठनम्।
  गमिष्यति, पठिष्यति, लेखिष्यति।(कोष्ठकस्य साहाय्येन)
- गत,आगामि।
- गीतम्।
- स्म।
- अभवत्।
- क्त्वाप्रयोगः (कोष्ठकस्य साहाय्येन)।
- यदा – तदा।
- बन्धुवाचकशब्दाः।
- वेशभूषणानां नामानि।
- वर्णाः।
- रुचयः।
- क्रीडा - एकश्वासेन सङ्ख्याकथनम्।
- कथा।
- पञ्च वाक्यानि।
- सूचना।
- ऐकमत्यः।

षष्ठं दिनम्

- गीतम्।
- पुनःस्मारणम्।
- नूतनम् x पुरातनम्,
- बहु x किञ्चित्,
- दीर्घः x ह्रस्वः।
- उन्नतः x वामनः।
- स्थूलः x कृशः।
- एतादृश, तादृश, कीदृश?
- तुमुन् (कोष्टकस्य साहाय्येन)।
- किन्तु।
- निश्चयेन।
- बहुशः / प्रायशः।
- किल / खलु।
- शक्नोति।
- गीतम्।
- विशेषणविशेष्यभावस्य अभ्यासः।(प्रथमाविभक्तौ)
  सः उत्तमः बालकः।
  सा उत्तमा बालिका।
  तत् उत्तमं पुस्तकम्।
- इव। विनोदकणिका।(गतवान् 'इव' अभिनयं कृतवान्!)
- अपेक्षया।
- पशूनां नामानि।
- अवयवानां नामानि।
- वाक्यविस्तारणाभ्यासः।
  (सः मम पुस्तकं प्रातःकाले पञ्चवादने पठितवान्।)
- इतः पूर्वम् - इतः परम्।
- 'रामकृष्ण' सङ्ख्याक्रीडा।
- कथा।
- षट् वाक्यानि।
- सूचना।
- ऐकमत्यः।

सप्तमं दिनम्

- गीतम्।
- पुनःस्मारणम्।
- क्त्वा - तुमुन् - परिवर्तनाभ्यासः।
- बहिः x अन्तः।
- रिक्तम् x पूर्णम्।
- इतोऽपि।
- इत्युक्ते।
- अन्ते।
- चेत् - नो चेत्।

- गीतम्।
- आरोग्यसम्बन्धिशब्दाः – वैद्यरोगिसम्भाषणम्।
- प्रश्नोत्तरस्पर्धा।
- ऋषीणां नामानि।
- कथा - शिक्षकः एकां कथां वदति। अनन्तरं छात्रेषु एकैकः तस्याः कथायाः एकैकं वाक्यम् उक्त्वा कथां सम्पूर्णां करोति।
- सङ्ख्या - दीर्घसङ्ख्यापाठनम्।
- प्रश्नोत्तरम्।
- क्रीडा - (गणद्वये नामस्मरणक्रीडा)
- कथा।
- पुस्तकानां परिचयः।
- सप्त वाक्यानि।
- सूचना।
- ऐक्यमन्त्रः।

अष्टमं दिनम्

- गीतम्।
- पुनःस्मारणम्।
- वारम्।
- अतः - यतः परिवर्तनाभ्यासः।
- यद्यपि – तथापि।
- यत्र – तत्र।
- कति - कियत् - एतयोः भेदज्ञापनम्।
- यावत् – तावत्।
- यत् – तत्।
- यः - सः।
- या – सा।
- गीतम्।
- अस्माकम्।
- चर्चा।
- सङ्ख्या - 'शतायुः - गतायुः' क्रीडा।
- विनोदकणिकाकथनम्।
- कथा।
- अष्ट वाक्यानि।
- समाजनिधिविषये सूचना।
- ऐक्यमन्त्रः।

नवमं दिनम्

- गीतम्।
- पुनःस्मारणम्।
- चित्।
- ....द्वयम्।
- सङ्ख्यासु लिङ्गभेदः।
  एकः - एका - एकम्
  द्वयम् - द्वयम् - द्वयम्
  त्रयः - तिस्रः - त्रीणि
  चत्वारः - चतस्रः - चत्वारि
- शिश्रकः - अहं वैद्यः - मम नाम सुरेशः
  (छात्राः तमुद्दिश्य प्रश्नान् पृच्छेयुः।)
- ..... अर्थम् (समाजार्थम्, संस्कृतकार्यार्थम्...)।
- गीतम्।
- तव्यत् – अनीयर्।
- अनन्त्यकथारचना।
- सङ्ख्यान्वेषणम् (क्रीडा)।
- छात्रैः सह प्रश्नोत्तरम्।
- समाजनिधिविषये पुनःस्मारणम्।
- ऐक्यमन्त्रः।

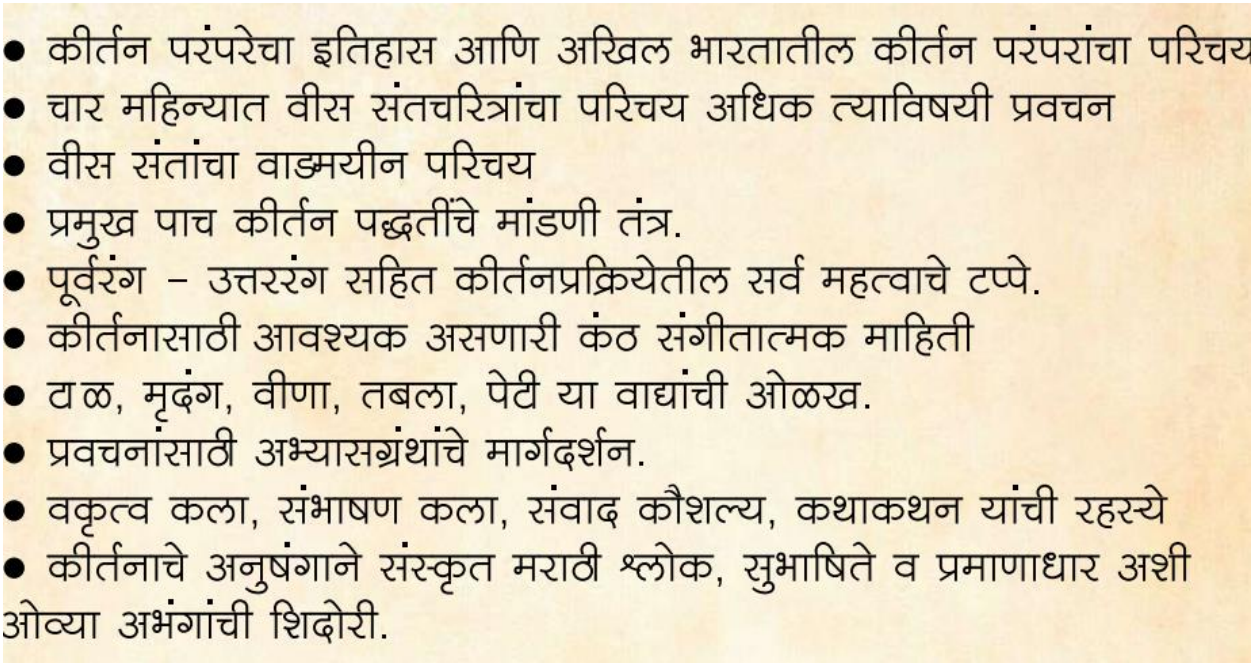
दशमं दिनम्

- गीतम्।
- पुनःस्मारणम्।
- पत्रलेखनम्।
- दूरवाणीसम्भाषणम्।
- मार्गनिर्देशः – कुत्र गन्तव्यम् इत्यादि।
- तव्यत् अभ्यासार्थम् – अद्य किं किं करणीयम् ?
- सान्दर्भिकभाषणम् –
  1. प्रवासात् प्रतिनिवर्तनस्य।
  2. आपणिकस्य इत्यादि।
- क्रीडा – सङ्ख्यायोजनम् (गणद्वये)।
- शुभाशयाः।
- असत्यकथनम् / कल्पनाकथनम्।
- समारोपः (सर्वैः शिक्षार्थिभिः भारतमातुः पूजां कृत्वा निधिसमर्पणं करणीयम्।)
- पत्राचारप्रगतशिक्षणादिविषये सूचना।
- ऐक्यमन्त्रः।

Course Code	25HS02PR0102-17			
Category	Co-Curricular Activity			
Course Title	Kirtan Kala			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course objectives:
The objective of the course is to provide the students with a spiritual experience as well as its benefits to them in the form of better abilities to concentrate and develop the ability to create a peaceful mind.

Course outcome
At the end of the course, the students will be able to achieve the following:
CO1: Learn from the inspiring spiritual journey of the saints and the history of Kirtan tradition
CO2: Learn about the musical instruments used in the art of Kirtan
CO3: Develop communication skills



Course Code	25HS02PR0102-18			
Category	Co-Curricular Activity			
Course Title	Introduction to German Language and culture			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course objective:

To help build a foundation and interest in German language so that the students can pursue the proficiency levels of the language in higher semesters.

Course outcomes:

On successful completion of the course the students will be able to achieve the following:
CO1. Demonstrate basic knowledge about Germany, the culture and similarities/differences between India and Germany
CO2. Learn to use simple language structures in everyday communication.
CO3. Develop ability to write in basic German about themselves and others.
CO4. Develop ability to read and understand beginner level texts in German.

Syllabus
List of Practicals

Practical-1: Orientation about Germany, the language, and culture
Practical-2: Vocabulary building for everyday conversations
Practical -3: Numbers, days and time
Practical-4: Introducing Oneself & Others
Practical-5: Reading Skills: Reading simple text in German language
Practical-6: Basic Verbs & Sentence Construction
Practical-7: Food & Dining, Giving Directions & Transportation
Practical-8: Writing Skills: Write basic German and practice

Recommended reading

1. German Made Easy by Diego A. Agundez
2. Teach Yourself Complete German: Learn to Read, write, Speak and Understand A new Language by Paul Coggle, Heiner Schenke
3. Netzwerk A1 by Helen Smitz, Stefanie Dengler and Paul Rusch
4. Deutsche Welle (DW) – www.dw.com/learngerman
5. BBC Languages – German – www.bbc.co.uk/languages/german
6. Goethe-Institut – www.goethe.de

Course Code	25HS04PR0102-1			
Category	Co-Curricular Activity			
Course Title	Adventure Sports			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Objective:

This course introduces adventure sports, emphasizing experiential learning through participation in various activities. The course will cover the fundamentals, safety procedures, and physical and mental benefits of adventure sports. Students will engage in outdoor activities such as wall climbing, rappelling, and more, fostering a connection with nature and understanding the principles of risk management.

Course Outcome: By the end of this course, students will:

- Understand the principles and benefits of adventure sports.
- Develop basic skills in selected adventure sports.
- Learn and apply safety measures and risk management techniques.
- Foster teamwork, leadership, and problem-solving skills.
-

Syllabus:

- Tent pitching, knot practice session and Tent allotment
- Activities like Jumarring and Climbing
- Individual challenge like Burma bridge, ladder bridge, multi vine
- Group Task like improvise raft making and Kayaking
- Activities like Archery rifle shooting, cycle ride

Pattern of Classes: 30 Hrs. Camp (2 Days and 1 Night)

Assessment Pattern:

Assessment Type	**Weightage in Marks**	**Total Marks**
Practical 25HS04PR0102-1	Internal Marks – 25 Marks External Marks – 25 Marks	50
		Total – 50 Marks

Course Code	25HS04PR0102-2			
Category	Co-Curricular Activity			
Course Title	Introduction to Defense Forces & Obstacle Training			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Objective:

- Understand the Structure and Function of Defense Forces
- Familiarize with Defense Force Training and Discipline
- Learn Basic Obstacle Course Techniques
- Apply Problem-Solving and Teamwork in Obstacle Training
- Explore the Role of Obstacle Training in Defense Preparedness
-

Course Outcome:

Upon successful completion of the course, students should be able to:

- Describe the Structure and Functions of Defense Forces
- Demonstrate Knowledge of Defense Training Protocols
- Navigate Basic Obstacle Courses & Connect Obstacle Training to Defense Preparedness
- Collaborate and Problem-Solve in Team-Based Exercises

Syllabus:

- Knot and Hitch practice session
- Activities like Rappelling & Wall Climbing
- Burma bridge & ladder bridge
- First Aid
- Rifle Shooting
- Horse riding
- Group Task and Team building activities

Pattern of Classes: 30 Hrs Camp (2 Days and 1 Night Camp)

Assessment Pattern:

Assessment Type	Weightage in Marks	Total Marks
Practical 25HS04PR0102-2	Internal Marks – 25 Marks External Marks – 25 Marks	50
		Total – 50 Marks

Course Code	25HS04PR0102-3			
Category	Co-Curricular Activity			
Course Title	First Aid & Disaster Management			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Objective:

- Understand Disaster Types and Characteristics
- Learn Risk Assessment and Management
- Master Emergency Preparedness and Response
- Explore Recovery and Reconstruction
- Develop Skills in Communication and Coordination
- Understand Legal and Ethical Considerations

Course Outcome:

Upon successful completion of the disaster management course, students should be able to:

- Identify and Categorize Disasters
- Conduct Risk Assessments
- Develop Emergency Plans
- Implement Response Strategies

Syllabus:

- Basic First Aid
- Transportation of Casualty
- Injury Prevention & Cure
- Various Types of Knots & Hitches
- Various team building activities
- Fire emergencies & use of extinguishers (Optional)
- Snake Bite & Environmental emergencies.

Assessment Pattern:

Assessment Type	Weightage in Marks	Total Marks
Practical 25HS04PR0102-3	Internal Marks – 25 Marks External Marks – 25 Marks	50
		Total – 50 Marks

Course Code	25HS04PR0102-4			
Category	Co-Curricular Activity			
Course Title	Basic Nutritional Course			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Objective:

In the "Basics of Nutrition" course, students will develop a comprehensive understanding of essential nutrients and their roles in supporting overall health. They will learn to apply dietary guidelines effectively, tailoring recommendations to various age groups and health conditions. Additionally, students will cultivate the skills needed to assess and improve their own and others' eating habits for better health outcomes.

Course Outcome:

By the end of the course, students will be able to

- Accurately describe the functions of key nutrients and their impact on health,
- create balanced meal plans based on established dietary guidelines,
- critically evaluate nutrition information to distinguish between credible and misleading sources.

Syllabus:

Unit I

- Introduction to Nutrition – Define Balanced Diet, Nutrition, Optimum Nutrition, Nutrients, Concept of Health, Recommended Dietary Allowances (RDA)
- Carbohydrates (sources, functions and digestion)
- Proteins (sources, functions and digestion)
- Fats (sources, functions and digestion)
- Micronutrients (vitamins and minerals-sources, functions and digestion)

Practical I

- Display of all the foods with the help of students and while demonstrating teacher will again explain role and importance of nutrition in daily life. Deficiency will lead to chronic diseases and its prevention is very necessary for the quality of life.

Unit II

- What is Body Mass Index?
- What is Basal Metabolic Rate?
- What is Ideal Body Weight? (Male/Female)
- How to read labels on Food Packets?
- How to choose smart food and Concept of Rainbow diet, My Food Pyramid or My plate given by ICMR- NIN.

Practical II

- Calculation of Body Mass Index, Basal Metabolic Rate, Ideal Body Weight (Male/Female) with the use of self-body measurements.
- Demonstration of Rainbow diet, My Food Pyramid or My plate in a class.

Assessment Pattern:

Assessment Type	Weightage in Marks	Total Marks
Practical 25HS04PR0102-4	Internal Marks – 25 Marks External Marks – 25 Marks	50
		Total – 50 Marks

Course Code	25HS04PR0102-5			
Category	Co-Curricular Activity			
Course Title	Stress Management Through Yoga & Meditation			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	I

Course Objective: Mental health is one of the most important facets of human life. Academic learning has emerged as a major source of stress among young students worldwide. Promoting mental well-being among students in India is a crucial step toward achieving Sustainable Development Goal 3 (Good Health and Well-being). Stress management involves using various techniques and strategies to control stress levels, improve how you react to stressful situations. Yoga combines physical movement with deep breathing and meditation, providing a holistic approach to stress relief.

Course Outcome:
Upon successful completion of the course, students should be able to:

- Understand the basics of stress management.
- Analyze stress triggers and to manage them.
- Evaluate the responses to stressful situations.
- Apply the techniques of Yog & Meditation for stress management in day-to-day life.

Syllabus:
Unit-1
Introduction to Stress: The Meaning of Stress, types of stress: distress, eustress
Stress Management Techniques I:
Treatment 1- (Asanas): Tadasana, Trikonasana, Vrikshasana, Garudasana,, Ardha-Padamasana, Padamasana, Vajarasana, Ushtrasana, Gomukhasna,, Paschimottanasan, ,Ardha Halasana, ,Setu-Bandhanasa,Naukasana, Bhujangasana, and Dhanurasana; along with relaxing asanas
Unit-2
Spiritual approach to stress management.
Stress Management Techniques II
Treatment 2 – (Prananyam) Deep breathing, Yoga, Mindfulness meditation
Rechak, Purak, Kumbhak, Nadi Suddhi and Bhramari Pranayama.
Measuring Academic stress- It can be measure using questionnaire: Academic stress Scale (Sun .et al 2011**).**

Assessment Pattern:

Assessment Type	**Weightage in Marks**	**Total Marks**
Practical 25HS04PR0102-5	Continuous Assessment – 25 Marks Internal Test Evaluation – 25 Marks	50
		Total – 50 Marks

SYLLABUS OF SEMESTER II

Course Code	25EE01TP0201			
Category	Engineering Science Course			
Course Title	Microcontroller and Interfacing			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	II

Course Outcomes

On successful completion of the course, students will be able to:

- Understand the architecture of microprocessor and microcontroller.
- Develop, understand and analyze programming of a microprocessor and microcontroller.
- Acquire the knowledge, techniques and skill to interface external peripheral devices with microprocessor or microcontroller.
- Design microcontroller-based system to solve the real-world problem

Syllabus

Module I: **(06 Hrs)**
Introduction to RISC and CISC processors, Harvard and Von Neumann architecture, Introduction to Intel's 8086 architecture, pin diagram, bus concepts, addressing modes, segmentation, pipelining.

Module II:(06 Hrs)
Instruction set, stack and subroutines- simple and nested, stack manipulation, Memory mapping, interrupts-concept and structure, interrupt service routines, simple programs.

Module III: (06 Hrs)
Introduction to x51 Family Microcontrollers, their Architecture,pindiagram,addressing mode SFRs.

Module IV:(06 Hrs)
Instruction of 8051 , stack and subroutines- Memory concept, interrupts-concept and structure, interrupt service routines, Sleep mode, idle mode, Run Mode. simple programming.

Module V:(06 Hrs)
Interfacing of Switches & Relays, Stepper motor, LED, SSD, LCD, Analog-to-Digital Converter (ADC), DC motor with x51controller.

Module 6: **(05 Hrs)**
Commutation Protocols used with microcontroller: Parallel communication, Serial communication, Serial Peripheral Interface (SPI), I2C Communication, Introduction to USB.

Textbook:

1.Advanced Microprocessors and Peripherals; A. K. Ray & K. M. Bhurchandi; McGraw Hill, 3rdEdition

2.The 8051 Microcontroller and Embedded Systems Using Assembly and C; Muhammad Ali Mazidi, Pearson, 2nd Edition

Reference Books:

1.Microcomputer systems: the 8086/8088 family: Architecture, Programming, and Design; Yu-chengnd Liu, Glenn A. Gibson; Prentice-Hall, 2nd Edition.
2.The 8051 Microcontroller: Architecture, Programming, and Applications Kenneth Ayala, Pearson, 3rd Edition.

LIST OF EXPERIMENTS

PART-A:

Conduct the following experiments by writing Assembly Language Program (ALP) using 8086 and 8051 Registers using an evaluation board/simulator and the required software tool.

1.Write an ALP to multiply two 16-bit binary numbers.
2.Write an ALP to find the sum of first 10 integer numbers
3. Write an ALP to find factorial of a number.
4. Write an ALP to add an array of 16-bit numbers and store the 32-bit result in internal RAM
5. Write an ALP to add two 64-bit numbers.
6. Write an ALP to find the square of a number (1 to 10) using look-up table.
7. Write an ALP to find the largest/smallest number in an array of 32 numbers.
8. Write an ALP to arrange a series of 32-bit numbers in ascending/descending order.
9. Write an ALP to count the number of ones and zeros in two consecutive memory locations.
10. Write an ALP to Scan a series of 32-bit numbers to find how many are negative.

PART-B:

Conduct the following experiments on 8086/8051 evaluation board using evaluation version of Embedded 'C' & Keil compiler.

1. Display “Hello World” message using Internal UART.
2. Interface and Control a DC Motor.
3. Interface a Stepper motor and rotate it in clockwise and anti-clockwise direction.
4. Determine Digital output for a given Analog input using Internal ADC of ARM controller.
5. Interface a DAC and generate Triangular and Square waveforms.
6. Interface a 4x4 keyboard and display the key code on an LCD.
7. Using the Internal PWM module of ARM controller generate PWM and vary its duty cycle.
8. Demonstrate the use of an external interrupt to toggle an LED On/Off.
9. Display the Hex digits 0 to F on a 7-segment LED interface, with an appropriate delay in between.
10. Interface a simple Switch and display its status through Relay, Buzzer and LED.

Course Code	25HS03TH0213			
Category	Basic Science Course			
Course Title	Calculus & Linear Algebra			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	II

Course Objective

The objective of this course is to familiarize the prospective engineers with techniques in Calculus. It aims to equip the students with standard concepts and tools at an intermediate to advanced level that will serve them well towards tackling more advanced level of mathematics and applications that they would find useful in their disciplines.

Course Outcomes

On successful completion of the course, student shall be able to

1. Apply the concepts of continuity and differentiability to find Taylor's and Maclaurin series.
2. Understand the methods of partial derivatives and apply these concepts to determine extreme values of the functions of two variables.
3. Demonstrate the basic knowledge of vector differentiation and line integral.
4. Interpret the solutions of system of linear equations and use the concepts of Eigen values, Eigen vectors to find diagonalization of matrices, reduction of quadratic form to canonical form.
5. Internalize convergence of sequences and apply it to determine whether infinite series convergent or divergent with appropriate tests.

Syllabus

Module 1: (8 Lectures)

Differential Calculus: Functions of univariate, Mean value theorems: Rolle's theorem, Lagrange's theorem, Cauchy's theorem, Taylor's theorem, Taylor's and Maclaurin series.

Module 2: (10 Lectures)

Partial Differentiation: Partial derivatives, Euler's Theorem, chain rule, application of partial differentiation: total derivative, Jacobians, Maxima, Minima for the functions of two variables., Extrema of function of multivariable,

Module 3: (8 Lectures)

Vector Calculus: Scalar and vector fields, gradient of scalar point function, directional derivatives, divergence and curl of vector point function, application of vector calculus: Line integral, Gradient Descent method.

Module 4: (8 Lectures)

Rank-nullity theorem; Consistency of system of linear equations and its solution, Orthogonal matrices, Eigen values and eigenvectors, Diagonalization of matrices, Orthogonal transformation and quadratic to canonical forms, Introduction to n-dimensional vector spaces, Singular value decomposition and its applications.

Module 5: (6 Lectures)
Infinite series: Sequences, Infinite series of real and complex numbers, Cauchy criterion, tests of convergence, absolute and conditional convergence, uniform convergence, power series, radius of convergence.

Textbooks/References
1. Erwin Kreyszig, Advanced Engineering Mathematics, 9th Edition, John Wiley & Sons, 2006.
2. Veerarajan T., Engineering Mathematics for first year, Tata McGraw-Hill, New Delhi, 2008.
3. N.P. Bali and Manish Goyal, A text book of Engineering Mathematics, Laxmi Publications, Reprint, 2010.
4. B.S. Grewal, Higher Engineering Mathematics, Khanna Publishers, 35th Edition, 2000.
5. Ramana B.V., Higher Engineering Mathematics, Tata McGraw Hill New Delhi, 11th Reprint, 2010.
6. P. N. Wartikar and J. N. Wartikar, A text book of Applied Mathematics Volume I & II, Pune Vidhyarthi Griha Prakashan, Pune-411030 (India).

Course Code	25EE01TP0202			
Category	Program Core Course			
Course Title	Programming for Problem Solving			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	II

Course Outcomes

On successful completion of the course, students will be able to:

1. Develop the fundamentals of C programming and choose the loops and decision-making statements to solve and execute the given problem.
2. Formulate simple algorithms for arithmetic and logical problems, translate the algorithms to programs, test and execute the programs, and correct syntax and logical errors.
3. Use arrays, pointers, structures, and I/O operations for the formulation of algorithms and programs.
4. Apply programming concepts to solve matrix addition, multiplication problems, and searching & sorting problems.
5. Implement iterations and recursions, decompose a problem into functions, and synthesize a complete program using divide and conquer approach.

Syllabus

Module I:
Introduction to Programming: Introduction to components of a computer system (disks, memory, processor, where a program is stored and executed, operating system, compilers etc.).
Idea of Algorithm: Steps to solve logical and numerical problems.
Representation of Algorithm: Flowchart/Pseudocode with examples.
Arithmetic expressions and precedence.

Module II:
C Programming Language: Introduction to C language: Keywords, Constant, Variable, Data types, Operators, Types of Statements, Pre-processor Directives,
Decision Control Statement - if, if-else, nested if-else statement, switch case, Loops, and writing and evaluation of conditionals and consequent branching.

Module III:
Arrays and Basic Algorithms: Arrays: 1-D, 2-D, Character arrays and Strings.
Searching, Basic Sorting Algorithms, Finding roots of equations, notion of order of complexity through example programs (no formal definition required).

Module IV:
Functions and Recursion: User-defined and Library Functions, Parameter passing in functions, call by value, passing arrays to functions: idea of call by reference.

Recursion: As a different way of solving problems. Example programs, such as Finding Factorial, Fibonacci series.

Module V:
Pointers and Structures: Structures, Defining structures, Array of Structures, Introduction to pointers, Defining pointers, Pointer arithmetic, Pointer operators, Use of Pointers in self-referential structures.

Module VI:
File Handling: Streams in C, Types of Files, File Input/Output Operations: Modes of file opening, Reading and writing the file, Closing the files using fflush().

Text Books

1. *Programming in ANSI C* – E. Balagurusamy, McGraw Hill
2. *Mastering C* – K. R. Venugopal and S. R. Prasad, Tata McGraw Hill

Reference Books

1. *Programming with C* – Byron Gottfried, Schaum's Outline Series
2. *Let Us C* – Yashwant Kanetkar, BPB Publication

. **LIST OF EXPERIMENTS**

1. Using basic data types of C, implement arithmetic expressions.
2. Implement programs using Decision Control Structures.
3. Demonstrate use of Loop Control Structures.
4. Implement programs using Multi-way Decision Control Structures (Switch Case).
5. Apply Functions and Recursion to simplify programs.
6. Initialize arrays and apply them to solve problems of 1D and 2D arrays.
7. Demonstrate use of Structures and Pointers.
8. Apply file handling concepts in C.

Note: 2 to 3 practice programs will be taken on each of the experiments mentioned above.

Course Code	25EE01TH0203			
Category	Engineering Science Course			
Course Title	AI and ML Essentials			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	II

Course Description:
The AI and ML Essentials course provides a comprehensive introduction to the fundamental concepts and applications of Artificial Intelligence (AI) and Machine Learning (ML). This 45-hour course covers key topics such as AI principles, machine learning techniques (supervised, unsupervised, reinforcement learning), natural language processing, data handling, popular AI/ML tools and platforms, and ethical considerations. The course also explores future trends in AI and ML, including deep learning, AI in robotics, healthcare, and automation.

Course Objectives:

- Understand the core concepts and principles of Artificial Intelligence (AI).
- Learn the basics of Machine Learning (ML) and its different types (supervised, unsupervised, reinforcement learning).
- Explore the key components of AI and ML, including data, algorithms, and computing power.
- Gain insights into Natural Language Processing (NLP) and its applications.
- Become familiar with popular AI/ML libraries and frameworks, particularly Python libraries like NumPy, Pandas, Scikit-learn, TensorFlow, and Keras.
- Understand the ethical considerations and challenges in AI, including bias, transparency, and accountability.
- Explore future trends in AI, such as deep learning, AI in robotics, healthcare, and automation.

Course Outline:

Unit 1: Overview of Artificial Intelligence (AI) - 8 hours

Definition of AI and Key Concepts

- **Understanding Artificial Intelligence**
  - Definition: AI refers to the simulation of human intelligence in machines programmed to think, learn, and make decisions.
  - Core Principles: Reasoning, problem-solving, perception, learning, and language understanding.
  - Difference between AI, Machine Learning (ML), and Deep Learning (DL).
- **Types of AI**
  - **Narrow AI (Weak AI):** Designed for specific tasks (e.g., facial recognition, chatbots).
  - **General AI (Strong AI):** Hypothetical AI with human-like cognitive abilities.
  - **Super AI:** AI surpassing human intelligence and capabilities.
- **Key Components of AI**
  - **Data:** The foundation of AI systems; types of data (structured, unstructured, semi-structured).

    - **Algorithms:** Rules and statistical models used to process data (e.g., decision trees, neural networks).
    - **Computing Power:** Role of GPUs, TPUs, and cloud computing in enabling AI.

Historical Perspective and Evolution of AI

- **Key Milestones in AI Development**
    - 1950: Alan Turing's Turing Test.
    - 1956: The Dartmouth Conference (birth of AI as a field).
    - 1997: IBM Deep Blue defeats chess champion Garry Kasparov.
    - 2011: IBM Watson wins *Jeopardy!*.
    - 2016: AlphaGo defeats world champion Go player.
- **Early AI Systems vs Modern AI**
    - Early AI: Rule-based systems, limited data, and computing power.
    - Modern AI: Data-driven approaches, deep learning, and scalable infrastructure.

Applications of AI in Everyday Life

- **AI in Personal Assistants**
    - How Siri, Alexa, and Google Assistant use Natural Language Processing (NLP) and speech recognition.
- **AI in Autonomous Vehicles**
    - Role of computer vision, sensor fusion, and reinforcement learning.
- **AI in Healthcare**
    - Applications in diagnostics, drug discovery, and personalized medicine.
- **AI in E-commerce**
    - Recommendation systems, fraud detection, and customer support chatbots.

Machine Learning Basics

- Supervised Learning: Training models with labeled data (e.g., spam detection).
- Unsupervised Learning: Finding patterns in unlabeled data (e.g., customer segmentation).
- Reinforcement Learning: Learning through trial and error (e.g., game-playing AI).

Natural Language Processing (NLP)

- Definition: A subfield of AI focused on interaction between computers and humans using natural language.
- Key Concepts:
    - Tokenization: Breaking text into words or phrases.
    - Sentiment Analysis: Determining the emotional tone of text.
    - Named Entity Recognition (NER): Identifying names, dates, and other entities.
- Applications: Chatbots, voice assistants, and language translation.

AI Ethics and Bias

- Importance of unbiased data and algorithms.
- Case studies of AI bias in facial recognition and hiring systems.

Future Trends in AI

- Explainable AI (XAI): Making AI decisions transparent and interpretable.
- AI in edge computing: Bringing AI capabilities to local devices.

- Quantum computing and its potential impact on AI.

Challenges and Opportunities in AI

- **Challenges**
  - Ethical concerns: Bias, privacy, and job displacement.
  - Technical limitations: Data quality, interpretability, and scalability.
- **Opportunities**
  - Advancements in healthcare, education, and sustainability.
  - Potential for solving complex global problems.

Unit 2: Introduction to Machine Learning (ML) - 9 hours

What is Machine Learning?

- **Basic Concepts of Machine Learning**
  - Definition of Machine Learning (ML): Teaching machines to learn from data without explicit programming.
  - Key components: Data, algorithms, models, and predictions.
  - Difference between traditional programming and ML.
  - Applications of ML in real-world scenarios.
- **Types of Machine Learning: Supervised, Unsupervised, and Reinforcement Learning**
  - **Supervised Learning**:
    - Definition: Learning from labeled data (input-output pairs).
    - Examples: Classification (e.g., spam detection) and regression (e.g., house price prediction).
  - **Unsupervised Learning**:
    - Definition: Learning from unlabeled data to find patterns or groupings.
    - Examples: Clustering (e.g., customer segmentation) and dimensionality reduction (e.g., PCA).
  - **Reinforcement Learning**:
    - Definition: Learning through trial and error using rewards and penalties.
    - Examples: Game playing (e.g., AlphaGo), robotics, and autonomous driving.

Relation between AI and ML

- **The Role of ML in AI Systems**
  - AI as the broader concept: Machines performing tasks that require human intelligence.
  - ML as a subset of AI: Enabling systems to learn and improve from experience.
  - Deep learning as a subset of ML: Neural networks for complex tasks like image and speech recognition.
- **Key Terminologies: Features, Labels, Training, and Testing Data**
  - **Features**: Input variables used to make predictions (e.g., age, income).
  - **Labels**: Output variables to be predicted (e.g., spam or not spam).
  - **Training Data**: Dataset used to train the model.
  - **Testing Data**: Dataset used to evaluate the model's performance.
  - **Validation Data**: Dataset used to tune hyperparameters and prevent overfitting.

Popular ML Algorithms (Overview)

- **Linear Regression**
  - Definition: A supervised learning algorithm for predicting continuous values.
  - Equation: y=mx+b$y=mx+b$, where yy is the dependent variable, xx is the independent variable, mm is the slope, and bb is the intercept.
  - Use cases: Predicting house prices, stock prices, etc.
- **Decision Trees**
  - Definition: A tree-like model for decision-making based on feature values.
  - Concepts: Nodes (decisions), branches (outcomes), and leaves (final predictions).
  - Use cases: Classification (e.g., loan approval) and regression.
- **K-Means Clustering**
  - Definition: An unsupervised learning algorithm for grouping data into clusters.
  - Concepts: Centroids, Euclidean distance, and convergence.
  - Use cases: Customer segmentation, image compression.

Key Use Cases of ML

- **Image Recognition**
  - Applications: Facial recognition, object detection, medical imaging.
  - Algorithms: Convolutional Neural Networks (CNNs).
- **Speech Recognition**
  - Applications: Virtual assistants (e.g., Siri, Alexa), transcription services.
  - Algorithms: Recurrent Neural Networks (RNNs), Transformers.
- **Recommendation Systems**
  - Applications: Netflix, Amazon, Spotify.
  - Algorithms: Collaborative filtering, matrix factorization.

Unit 3: Key Concepts in AI and ML - 9 hours

Data: The Foundation of AI and ML

- **Importance of Data in AI/ML Systems**
  - Data as the backbone of AI/ML: Why quality data is critical for model performance.
  - Data preprocessing: Cleaning, normalization, and transformation.
  - Data splitting: Training, validation, and test sets.
  - Data augmentation techniques for improving model robustness.
  - Challenges: Missing data, outliers, and imbalanced datasets.
- **Types of Data: Structured, Unstructured, and Semi-Structured**
  - **Structured Data**: Tabular data, relational databases, and CSV files.
  - **Unstructured Data**: Text, images, audio, and video.
  - **Semi-Structured Data**: JSON, XML, and NoSQL databases.
  - Data representation: Feature engineering, embeddings, and encoding (e.g., one-hot encoding, label encoding).

Algorithms: The Driving Force

- **Understanding the Role of Algorithms in ML Models**
  - What are algorithms? Mathematical frameworks for solving problems.

- Types of ML algorithms: Supervised, unsupervised, and reinforcement learning.
- Key concepts: Loss functions, optimization, and gradient descent.
- Overfitting vs. underfitting: Causes and solutions (e.g., regularization, dropout).

- **Concept of Model Training and Optimization**
  - Training process: Forward propagation, backpropagation, and weight updates.
  - Hyperparameters: Learning rate, batch size, and epochs.
  - Optimization techniques: Stochastic Gradient Descent (SGD), Adam, and RMSprop.
  - Evaluation metrics: Accuracy, precision, recall, F1-score, and ROC-AUC.

The AI/ML Development Lifecycle

- **Data Collection, Model Training, Model Evaluation, and Model Deployment**
  - Data collection: APIs, web scraping, and public datasets.
  - Model training: Iterative process of learning from data.
  - Model evaluation: Metrics for classification, regression, and clustering.
  - Model deployment: Exporting models (e.g., Pickle, ONNX), APIs, and cloud integration.
- **Introduction to Cross-Validation and Model Tuning**
  - Cross-validation: K-fold, stratified K-fold, and leave-one-out.
  - Hyperparameter tuning: Grid search, random search, and Bayesian optimization.
  - Model selection: Comparing multiple models for best performance.

Ethics in AI and ML

- **Ethical Considerations in AI: Bias, Transparency, and Accountability**
  - Bias in AI: Types (e.g., selection bias, confirmation bias) and sources (e.g., biased datasets).
  - Transparency: Explainable AI (XAI) and interpretability techniques (e.g., SHAP, LIME).
  - Accountability: Legal and ethical responsibilities of AI practitioners.
  - Case studies: Real-world examples of AI ethics violations.
- **Responsible AI Practices**
  - Fairness: Ensuring equitable outcomes for all groups.
  - Privacy: Data anonymization and GDPR compliance.
  - Sustainability: Reducing the environmental impact of AI systems.
  - Tools for ethical AI: AI Fairness 360, Fairlearn, and What-If Tool.

Unit 4: Key Tools and Platforms for AI and ML - 9 hours

Introduction to Popular AI/ML Libraries and Frameworks

- **Python for AI/ML**
  - Why Python? Easy-to-read syntax, extensive libraries, and community support.
  - Key Python libraries for AI/ML:
    - **NumPy:** Numerical computing with arrays and matrices.
    - **Pandas:** Data manipulation and analysis with DataFrames.
    - **Matplotlib:** Data visualization and plotting.
- **Machine Learning Libraries**

    - **Scikit-learn:** A versatile library for traditional ML algorithms (e.g., regression, classification, clustering).
    - **TensorFlow:** An open-source framework for deep learning and neural networks.
    - **Keras:** A high-level API built on TensorFlow for simplifying deep learning model development.

Data Preprocessing in Machine Learning

- **Normalization and Standardization**
    - Normalization: Scaling data to a range of [0, 1].
    - Standardization: Scaling data to have a mean of 0 and a standard deviation of 1.
- **Handling Missing Values**
    - Imputation: Replacing missing values with statistical measures (mean, median, mode).
    - Dropping: Removing rows or columns with missing values.
- **Encoding Categorical Variables**
    - Label Encoding: Converting categories into numerical labels.
    - One-Hot Encoding: Creating binary columns for each category.

Deep Learning Basics

- **Neural Networks**
    - Structure: Input layer, hidden layers, and output layer.
    - Activation Functions: ReLU, Sigmoid, and Softmax.
- **Training Process**
    - Forward Propagation: Passing data through the network.
    - Backpropagation: Adjusting weights using gradient descent.
- **Loss Functions and Optimizers**
    - Loss Functions: Mean Squared Error (MSE), Cross-Entropy.
    - Optimizers: SGD, Adam, RMSprop.

Natural Language Processing (NLP)

- **Text Preprocessing**
    - Tokenization: Splitting text into words or sentences.
    - Stemming and Lemmatization: Reducing words to their root forms.
- **Feature Extraction**
    - Bag of Words (BoW): Representing text as word frequency vectors.
    - TF-IDF: Weighing words based on their importance in a document.
- **Sentiment Analysis Techniques**
    - Rule-based: Using predefined rules and lexicons.
    - Machine Learning-based: Training models on labeled datasets.

Introduction to Data Visualization

- **Importance of Visualizing Data for AI/ML**
    - Helps in understanding data distributions, patterns, and outliers.
    - Essential for exploratory data analysis (EDA) and model evaluation.
- **Overview of Data Visualization Tools**
    - **Matplotlib:** Basic plotting library for creating static, animated, and interactive visualizations.

- **Seaborn:** Built on Matplotlib, provides high-level interfaces for statistical graphics.

Jupyter Notebooks for AI/ML Development

- **Setting Up Jupyter for ML Projects**
  - Installation and launching Jupyter Notebook.
  - Overview of the Jupyter interface (cells, kernels, and outputs).
- **Writing and Running Python Code for ML**
  - Using Jupyter for interactive coding, data exploration, and visualization.
  - Best practices for organizing ML projects in Jupyter Notebooks.

Unit 5: Future Trends in AI & ML - 10 hours

Convolutional Neural Networks (CNNs) and Image Classification

- **Fundamentals of CNNs**: Convolutional layers, pooling operations, activation functions, and fully connected layers.
- **Preprocessing Techniques**: Data normalization, augmentation, and dataset splitting strategies.
- **Performance Evaluation**: Metrics like accuracy, precision, recall, and F1-score.

Emerging AI Trends and Their Impact

- **AI in Robotics**: AI-driven robotics, reinforcement learning, and real-world industrial applications.
- **Natural Language Processing (NLP)**: Evolution from traditional models to transformers (GPT, BERT), applications in sentiment analysis, chatbots, and translation.
- **AI in Healthcare**: AI for medical imaging, disease prediction, drug discovery, and ethical concerns (bias, privacy, accountability).
- **AI in Automation**: Impact on industries like finance, retail, and manufacturing, along with job displacement concerns.

AI for Social Good

- **Tackling Global Challenges**:
  - **Climate Change**: AI in renewable energy, carbon footprint reduction, and environmental modeling.
  - **Healthcare**: AI in pandemic prediction, vaccine development, and medical accessibility.
  - **Education**: AI-driven personalized learning and tutoring systems.
- **Real-Time AI Applications**:
  - Chatbots: Intent recognition and response generation.
  - Recommendation Systems: Collaborative filtering and hybrid models.
  - Speech Recognition: Acoustic and language modeling.

AI in the Future: Opportunities and Challenges

- **Future AI Applications**:
  - **IoT & Smart Cities**: AI in traffic management, urban planning, and smart automation.
  - **Autonomous Systems**: AI in self-driving cars, drones, and logistics.
- **Challenges & Ethical Dilemmas**:
  - **Data Privacy & Security**: Risks of breaches and adversarial attacks.
  - **Bias & Transparency**: Algorithmic fairness and accountability.

Course Outcomes:

- Students will be able to define AI and differentiate between AI, Machine Learning, and Deep Learning.
- Students will be able to explain the key principles and components of AI systems.
- Students will be able to describe the different types of machine learning and their applications.
- Students will be able to apply Python and popular AI/ML libraries to perform basic machine learning tasks.
- Students will be able to analyze and discuss the ethical implications of AI technologies.
- Students will be able to identify and explain emerging trends in AI and their potential impact.

Skills Gained:

- Understanding of AI and ML concepts.
- Knowledge of machine learning algorithms.
- Data preprocessing and handling skills.
- Proficiency in Python and AI/ML libraries (NumPy, Pandas, Scikit-learn, TensorFlow, Keras).
- Ability to apply AI and ML techniques to real-world problems.
- Awareness of ethical considerations in AI.

Course Code	25EE01TP0204			
Category	Engineering Science Course			
Course Title	Computer Architecture and Organization			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	II

Course Outcomes:

Upon completion of this course, students will demonstrate the ability to:

1. **Understand** the common principles of computer architecture, multiprocessing and classify RISC/CISC.
2. **Develop** RISC-V assembly programs by understanding instruction formats
3. **Design** the RISC -V processor datapath and control on FPGA
4. **Apply** the concept of cache and virtual memory management in computer system.
5. **Evaluate** pipelined processor performance, identify hazards, and propose solutions to improve efficiency

Syllabus:

Module-I:

Introduction to computer system and its sub modules, Introduction to RISC and CISC paradigm, Functional Modeling of digital blocks.

Module-II:

Performance Equation, Common Principles of Computer organization: Amdahl's Law, Principle of Locality. Processor organization, instruction set (RISC-V), instruction formats, Representing Instructions in the Computer, Translating and Starting a Program, IEEE 754 floating point format.

Module-III:

Processor Design-Introduction, Datapath and control unit design, Performance Considerations, Multi-cycle design, Micro Programmed control design, Exception Handling.

Module-IV:

Motivation for Pipelining, Clock period and CPI, Pipelined data path, graphical representation, Pipelining Hazards.

Module-V:

Memory organization, concepts of semiconductor memory, memory management, concept of cache and associative memories, virtual memory.

Module-VI:

Parallel processing concepts, multiprocessors and its characteristics, Input/output Subsystem: -Interfaces and buses, I/O Operations, Designing I/O Systems, Overview of Domain-Specific Architectures

Text Books:

1. Computer Organization and Design RISC-V Edition - The Hardware/Software Interface, David A. Patterson, John L. Hennessy, 2nd Edition, 2021.

Reference Books:

1. Computer Architecture and Organization; J. P. Hayes; Third Edition (Fifth Reprint), McGraw Hill, 2012.
2. Computer Architecture And Parallel Processing; Kai Hawang, Faye A. Briggs, McGraw Hill, 2012

List of Experiments:

1. To design and integrate ALU and ALU control unit of RISC-V processor together and test it using a test bench. Encode the instructions to be executed by ALU and generate these encoded test vectors to control the ALU operations. Supply random test inputs from test bench to check the results using Vivado and FPGA.

2. Integrate register Files, ALU control, ALU together. Write a top module to connect these modules using wires in Verilog HDL. Demonstrate any one type of RISC-V instruction. Ensure the register files are getting initialized with the correct data and encoded instructions. Uses test bench to generate the relevant test vectors.

3. Write a top module to connect the instruction memory, data memory, register files, ALU and ALU control together. Encode one logical and one arithmetic instruction in the instruction memory. Initialize the data memory fields with the data. Verify the instructions fetch process from instruction memory. Write a Verilog test bench to verify arithmetic and logical instructions by applying suitable test cases and control signals.

4. Write a control path in Verilog HDL to control the execution of the data path of MIPS instructions in experiment3. Analyze the control signals for S type, R type and I type instructions using a suitable test bench.

5. Encode the assembly program in instruction memory with at least 1 R-type, 1 immediate type and 1 branch-type instruction in the memory. Demonstrate the execution of this program using Vivado tool and FPGA.

Course Code	25HS02TP0201			
Category	Ability Enhancement Course			
Course Title	English for Professional Communication			
Scheme & Credits	L	P	Credits	Semester
	2	2	3	II

Course Outcomes

On successful completion of the course, students will be able to:

1. Demonstrate effective use of word power in written as well as oral communication.
2. Understand the techniques of listening and apply the techniques of reading comprehension used in professional communication.
3. Apply the principles of functional grammar in everyday as well as professional communication.
4. Effectively implement the comprehensive principles of written communication by applying various writing styles.
5. Create precise and accurate written communication products.

Syllabus

Module I: Vocabulary Building

- Importance of using appropriate vocabulary
- Techniques of vocabulary development
- Commonly used power verbs, power adjectives and power adverbs
- Synonyms, antonyms, phrases & idioms, one-word substitutions, and standard abbreviations

Module II: Listening and Reading Comprehension

- **Listening Comprehension:** Active listening, reasons for poor listening, traits of a good listener, and barriers to effective listening
- **Reading Comprehension:** Types and strategies

Module III: Functional Grammar and Usage

- Identifying common errors in use of: articles, prepositions, modifiers, modal auxiliaries, redundancies, and clichés
- Tenses
- Subject-verb agreement, noun-pronoun agreement
- Voice

Module IV: Writing Skills

- Sentence Structures
- Sentence Types
- Paragraph Writing: Principles, Techniques, and Styles

Module V: Writing Practices

- Art of Condensation: Précis and Note Making
- Correspondence writing techniques and etiquettes – academic writing
- Group discussion, Résumé writing

Reference Books

1. *Communication Skills* – Sanjay Kumar and Pushp Lata, Oxford University Press, 2011
2. *Practical English Usage* – Michael Swan, Oxford University Press, 1995
3. *Remedial English Grammar* – F. T. Wood, Macmillan, 2007
4. *On Writing Well* – William Zinsser, Harper Resource Book, 2001
5. *Study Writing* – Liz Hamp-Lyons and Ben Heasly, Cambridge University Press, 2006
6. *Exercises in Spoken English* (Parts I–II), CIEFL, Hyderabad, Oxford University Press

English for Professional Communication Lab

Course Objective
To enhance competency of communication in English among learners

Course Outcomes
On completion of English Lab course, students will be able to achieve the following:
CO1: Apply effective speaking skills
CO2: Apply effective listening and reading skills
CO3: Demonstrate the techniques of effective public speaking and presentation skills
CO4: Evaluate and apply the effective strategies for Group Discussions

Syllabus
List of practicals

Computer Assisted + Activity Based Language Learning

Practical 1: Everyday Situations: Conversations and Dialogues – Speaking Skills
Practical 2: Pronunciation, Intonation, Stress, and Rhythm
Practical 3: Everyday Situations: Conversations and Dialogues – Listening Skills
Practical 4: Reading Skills

Activity Based Language Learning

Practical 5: Public Speaking
Practical 6: Presentation Skills: Orientation
Practical 7: Presentation Skills: Mock
Practical 8: Group Discussions: Practice
Practical 9: Group Discussions: Mock

Reference Books

1. *Communication Skills*. Sanjay Kumar and Pushp Lata. Oxford University Press. 2011.
2. *Practical English Usage.* Michael Swan. OUP. 1995.

Course Code	25HS02TH0203			
Category	Indian Knowledge System			
Course Title	Foundational Literature of Indian Civilization			
Scheme & Credits	L	P	Credits	Semester
	1	0	1	II

Course Outcomes

On successful completion of the course, students will be able to:

1. Understand the Indian knowledge system and its scientific approach.
2. Get introduced to the Vedic corpus and recognize the multi-faceted nature of the knowledge contained in the Vedic corpus.
3. Understand the salient features of the philosophical systems of the Vedic and non-Vedic schools.
4. Develop a basic understanding of the ancient wisdom recorded in various Indian literary works.

Syllabus

Module I: Overview of Indian Knowledge System

- Importance of ancient knowledge
- Defining IKS
- IKS classification framework
- Historicity of IKS
- Some unique aspects of IKS

Module II: The Vedic Corpus

- Introduction to Vedas
- Four Vedas and their divisions
- Six Vedangas
- Distinct features of Vedic life

Module III: Indian Philosophical Systems

- Development and unique features
- Vedic schools of philosophy:
    - Samkhya and Yoga School
    - Nyaya and Vaisheshika School
    - Purva-Mimamsa and Vedanta schools of philosophy
- Non-Vedic philosophies:
    - Jainism, Buddhism, and other approaches

Module IV: Vedic Maths -1

- Introduction of Vedic Mathematics
- Bodhyan geometry
- circular functions
- inverse circular functions

Module V: Vedic Maths -2

- Multiplication of polynomials using:
  - Nikhilaṃ Sutra
  - Ūrdhvatiryagbhyām Sutra
  - Verification using Gunitasamuccayaḥ Sutra
- Division of two polynomials using:
  - Parāvartya Yojayet Sutra
- HCF and LCM of two polynomials using:
  - Ādyamādyenāntyamantyena Sutra
  - Ānurūpyeṇa Sutra
- Factorization of polynomials up to degree 3 using:
  - Ānurūpyeṇa Sutra
  - Lopanasthāpanābhyām Sutra
  - Ādyamādyenāntyamantyena Sutra

Reference Material

1. B. Mahadevan, Vinayak Rajat Bhar, Nagendra Pavana R. N., *Introduction to Indian Knowledge System: Concepts and Applications*, PHI, 2022
2. S. C. Chatterjee and D. M. Datta, *An Introduction to Indian Philosophy*, University of Calcutta, 1984

Course Code	25EE01PR0205				
Category	VSEC				
Course Title	Mini Project I				
Scheme & Credits	L	T	P	Credits	Semester
	0	0	2	1	II

List of Experiments:

AI + NLP (Rule-based / ML-based) Projects

1. AI-Based FAQ Auto-Responder for College Queries
   - Build a system that answers student FAQs (admission, exams, fees) using keyword matching and basic NLP techniques.

2. Smart Email Categorization System
   - Classify emails into categories like Academic, Personal, Promotions, and Spam using supervised learning.

3. Sentiment Analyzer for Online Product Reviews
   - Analyze whether customer reviews are positive, negative, or neutral using text preprocessing and ML classification.

4. AI-Powered Feedback Analyzer for Teachers
   - Process student feedback forms and summarize common sentiments and issues.

5. Simple Chatbot for Campus Navigation
   - Create a chatbot that answers location-based questions inside a college campus using predefined intents.

6. Resume Skill Extractor Using NLP
   - Extract skills, education, and experience keywords from resumes using tokenization and NER concepts.

7. AI-Based Complaint Prioritization System
   - Automatically tag complaints as high, medium, or low priority based on keywords and sentiment.

8. Movie Review Emotion Classifier
   - Classify movie reviews into emotions such as happy, sad, angry, or neutral.

9. News Headline Categorizer
   - Categorize news headlines into Sports, Politics, Technology, Entertainment using ML.

10. AI-Based Customer Support Ticket Classifier
    - Classify incoming support tickets into Billing, Technical, Account, or General issues.

Machine Learning (Supervised / Unsupervised) Projects

11. Student Performance Prediction System
    - Predict student grades based on attendance, assignment scores, and internal assessments.

12. House Price Prediction (Beginner Level)
    - Build a linear regression model to predict house prices using limited features.

13. Loan Approval Prediction System
    - Classify loan applications as Approved or Rejected based on applicant data.

14. Employee Attrition Prediction Model
    - Predict whether an employee is likely to leave using basic HR data.

15. Customer Segmentation Using K-Means
    - Group customers based on purchase behavior for targeted marketing.

16. College Course Recommendation System (Rule + ML)
    - Recommend elective courses based on student interests and past performance.

17. Spam SMS Detection System
    - Detect spam messages using supervised learning and text preprocessing.

18. Used Car Price Estimation Tool
    - Estimate car prices based on age, mileage, and fuel type.

19. Online Shopping Fraud Detection (Basic)
    - Identify potentially fraudulent transactions using simple classification rules.

20. Student Dropout Risk Analysis System
    - Predict students at risk of dropping out using attendance and performance data.

AI Applications in Daily Life (Conceptual + Practical)

21. AI-Based Personal Study Planner
    - Suggest a weekly study schedule based on subjects, difficulty levels, and available hours.

22. Smart Food Recommendation System
    - Recommend meals based on user preferences and dietary restrictions.

23. AI-Based Weather Advisory System
    - Provide simple advice (carry umbrella, avoid travel) based on weather data patterns.

24. Career Path Suggestion Tool (Non-GenAI)
    - Suggest career options based on interests, skills, and academic performance.

25. AI-Based Book Recommendation System
    - Recommend books based on user ratings and preferences.

26. Smart Fitness Activity Recommendation App
    - Suggest exercises based on age, goals, and fitness level.

27. AI-Based Event Recommendation System for College

- Recommend events (seminars, fests) based on student interests.

28. Music Mood Classification System
    - Classify songs as energetic, calm, sad, or happy using metadata and simple rules.

29. AI-Based Attendance Pattern Analyzer
    - Identify attendance trends and risk patterns using historical data.

30. Smart Travel Destination Suggestion Tool
    - Suggest travel destinations based on budget, season, and interests.

Ethics, Bias & Explainability-Focused Projects

31. Bias Detection in Hiring Dataset
    - Analyze whether a hiring dataset shows bias based on gender or background.

32. Explainable Loan Approval System
    - Show why a loan was approved or rejected using feature importance.

33. AI Fairness Checker for Student Selection Models
    - Evaluate fairness across different student groups.

34. Ethical Analysis of Facial Recognition Systems
    - Study and demonstrate bias issues using simulated datasets.

35. AI Transparency Dashboard
    - Visualize how model predictions change with different input features.

36. Privacy-Aware Student Data Analyzer
    - Demonstrate anonymization techniques on student datasets.

Miscellaneous

37. AI-Based Learning Difficulty Identifier
    - Identify students struggling with specific topics using performance data.

38. Smart Question Difficulty Tagging System
    - Automatically label exam questions as Easy, Medium, or Hard.

39. AI-Based Classroom Engagement Analyzer
    - Analyze engagement using quiz scores and participation data.

40. Beginner AI Helpdesk System for New Students
    - A simple AI assistant that answers common academic and administrative questions.

- Problem understanding
- Dataset used / created
- AI/ML approach chosen
- Ethics considerations
- Demo + explanation

Course Code	25HS04PR0201			
Category	Co-Curricular Activity			
Course Title	Health-Fitness-Wellbeing (HFW)			
Scheme & Credits	L	P	Credit	Semester
	0	2	1	II

Aim of the Course: The course aims to foster Health and wellness through Healthy and Active Lifestyle and creating awareness about the fundamentals of Physical Education, Sports, Yoga, Recreation and its effectiveness through practical experiences and hands on activities.

Objectives of the Course:

1. To impart the students with basic concepts of Sports, Yoga and Recreational activities for health and wellness.
2. To familiarize the students with health-related Exercise and evaluate their Health-related Fitness.
3. To make Overall growth & development with team spirit, social values and leadership qualities among students through various sports, games and Yogic activities.
4. To create Environment for better interaction and recreation among students as neutralizer for stress through various minor and recreational games.

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

1. Understand fundamental skills, basic principle and practices of sports and Yoga.
2. Practically learn the principles of implementing general and specific conditioning of physical exercises and yoga.
3. Develop Health-related fitness and Body-mind co-ordination through various fitness activities, sports, recreational games and yoga.
4. Practice Healthy & active living with reducing Sedentary Life style.

Course Content:

Module 1:

- Warm up and Cool Down and Stretching Exercises.
- General and Specific Exercises.
- Calculation of BMI & Resting Pulse Rate.
- General and Specific exercises for strength, Speed, Agility, Cardiovascular Endurance, Flexibility, Coordinative abilities.
- Practice of Fundamental Skills of Volleyball, Table Tennis and Chess, etc.
- Knowledge and practice of the Equipment used in a Gymnasium and its application.

Module 2:

- Yoga: Standing, Sitting, Prone & Supine positions.
- Suryanamaskar.
- Pranayama, Meditation and Relaxation Techniques.
- Recreational Games.
- Practice of Fundamental Skills of Basketball, Football, Carrom, etc.

- Health related Physical Fitness Test.

Assessment Pattern:

Assessment Type	Weightage in Marks	Total Marks
Practical	Physical Efficiency Test – 30 Marks Sports/Games skill Activity/Project – 10 Marks Yoga Activities – 10 Marks	50
		Total – 50 Marks

References:

1. Russell, R.P. (1994). Health and Fitness Through Physical Education. USA: Human Kinetics.
2. Uppal, A.K. (1992). Physical Fitness. New Delhi: Friends Publication.
3. AAPHERD "Health related Physical Fitness Test Manual."1980 Published by Association drive Reston Virginia
4. Kumar, Ajith. (1984) Yoga Pravesha. Bengaluru: Rashtrothanna Prakashana.
5. Dr. Devinder K. Kansal, A Textbook of Test Evaluation, Accreditation, Measurements and Standards (TEAMS 'Science)

SYLLABUS OF SEMESTER III

Course Code	25EE01TP0301			
Category	Programme Core Course			
Course Title	Data Structures and Algorithms			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	III

Course Outcomes:

On successful completion of the course, students will be able to:

1. Understand the concepts of data structures.
2. Apply the concepts of linear (stacks, queues, linked lists) and non-linear (trees, graphs) data structures.
3. Implement different searching and sorting techniques.
4. Demonstrate the use and applicability of data conversion techniques
5. Devise algorithms for solving real-world problems.

Syllabus:

Module I: (07 Hrs)

Data Structures and Algorithm Basics:

Introduction: basic terminologies, elementary data organizations, data structure operations; abstract data types (ADT) and their characteristics. Algorithms: definition, characteristics, analysis of an algorithm, asymptotic notations, time and space tradeoffs. Array ADT: definition, operations and representations – row-major and column-major.

Module II: (07 Hrs)

Stacks and Queues:

Stack ADT: allowable operations, algorithms and their complexity analysis, applications of stacks – expression conversion and evaluation (algorithmic analysis), multiple stacks. Queue ADT: allowable operations, algorithms and their complexity analysis for simple queue and circular queue, introduction
to double-ended queues and priority queues.

Module III: (07 Hrs)

Linked Lists:

Singly Linked Lists: representation in memory, algorithms of several operations: traversing, searching, insertion, deletion, reversal, ordering, etc. Linked representation of stacks and queues, header node linked lists. Doubly and Circular Linked Lists: operations and algorithmic analysis.

Module IV: (07 Hrs)

Sorting, Searching and Hashing:

Sorting: Sorting: different approaches to sorting, properties of different sorting algorithms (insertion, Shell, quick, merge, heap, counting), performance analysis and comparison. Searching: necessity of a robust search mechanism, searching linear lists (linear search, binary search) and complexity analysis of search methods. Hashing: hash functions and hash tables, closed and open hashing, randomization methods (division method, mid-square method, folding), collision resolution techniques.

Module V: (8 Hrs)
Trees and Graphs:

Trees: basic tree terminologies, binary tree and operations, binary search tree [BST] and operations with time analysis of algorithms, threaded binary trees. Self-balancing Search Trees: tree rotations, AVL tree and operations, B+tree: definitions, characteristics, and operations (introductory)

Representation and Access: basic terminologies, representation of graphs, graph traversals: depth first search (DFS) and Breadth first search (BFS). Path Finding Algorithms: Dijkstra, Single Source Shortest Path (SSSP) algorithm, and Warshall- Floyds, All Sources Shortest Path (ASSP) algorithm. Spanning Trees: Introduction, minimum cost spanning trees, Prim's Method and Kruskal's Method for MSTs.

Text Book:
Data Structures and Algorithms in Java by Michael T. Goodrich, Roberto Tamassia, Michael H.Goldwasser, John Wiley & Sons, Inc., Sixth Edition, 2014

Reference Books:
1. Data Structures and Algorithms in Java, Robert Lafore, Second Edition, Sams Publishing, 2002
2. Thomas Cormen, Charles Leiserson, Ronald Rivest and Clifford Stein, Introduction to Algorithms,
Third Edition; Prentice Hall of India; 2009
3. G. A. V. Pai; Data Structures and Algorithms: Concepts: Techniques and Application; First Edition; McGraw Hill; 2008.

List of Experiments:

1. Implement recursive and non-recursive functions for the following searching techniques:
   a) Linear Search
   b) Binary Search
2. Develop and demonstrate the implementation of arrays and linked lists.
3. Construct and evaluate the following Abstract Data Types (ADTs) using arrays:
   a) Stack ADT
   b) Queue ADT
4. Analyze and convert an infix expression into postfix form using Stack ADT.
5. Design and verify a palindrome checker using both Stack and Queue ADTs.
6. Implement and demonstrate the following ADTs using singly linked lists:
   a) Stack ADT
   b) Queue ADT
7. Construct, manipulate, and evaluate a Binary Search Tree (BST) for the following operations:
   a) Insertion of elements
   b) Searching for a key element
   c) Deletion of an element
8. Develop and apply the functions of a Dictionary ADT using hashing techniques.
9. Demonstrate and compare recursive and non-recursive traversals of a binary tree:
   a) Preorder Traversal
   b) Inorder Traversal
   c) Postorder Traversal
10. Implement and analyze Breadth First Search (BFS) and Depth First Search (DFS) algorithms for a graph.
11. Implement, compare, and evaluate the following sorting techniques:
    a) Bubble Sort
    b) Insertion Sort
    c) Quick Sort
    d) Merge Sort
    e) Heap Sort
    f) Binary Tree Sort
12. Construct and perform the following operations on a B-Tree:
    a) Insertion
    b) Searching

Course Code	25EE01TP0302			
Category	Engineering Science Course			
Course Title	Fundamentals of Embedded system and IoT			
Scheme & Credits	L	P	Credits	Semester
	4	2	5	III

Course Outcomes

Course Objectives:

CO1: Explain the fundamentals of embedded systems, their architecture, design flow, development ecosystems, and the basic concepts of Internet of Things (IoT).
CO2: Describe and classify various IoT hardware platforms, sensors, actuators, and their roles in data acquisition and control for IoT applications.
CO3: Analyze IoT communication architectures, access technologies, and networking protocols such as Wi-Fi, Bluetooth, Zigbee, 6LoWPAN, MQTT, CoAP, and HTTP/HTTPS for different IoT scenarios.
CO4: Design and develop IoT-based solutions using suitable embedded platforms, sensors, actuators, and communication protocols for real-world applications in smart homes, industries, infrastructure, and Industry 4.0 environments.

Syllabus:

Module I: (06 hrs)
Introduction to embedded system: characteristics, classification, hardware/software components, design metrics and challenges, embedded system design flow, requirement analysis, controller and memory selection, controller comparison, Introduction to RISC Processor families, modern development platforms, and embedded development ecosystems.

Module II: (07 hrs)
Introduction to Internet of Things (IoT), Functional Characteristics, Architecture: 3 layers/5 layer model, Recent Trends in the Adoption of IoT, Role of cloud in IoT, Main design principles and needed capabilities, M2M and IoT Technology, Societal Benefits of IoT. Development boards for IoT and Platforms: - Introduction to Arduino, Raspberry Pi, ESP8266.

Module III: (07 hrs)
Sensing and Actuation: -Role of Sensors and Actuators in IoT, Sensors- Definition, characteristics, types and their working- Soil moisture sensor, DHT11, Ultrasonic sensor, IR sensor, PIR sensor, sound sensor, touch sensor, LDR, Gas sensor, etc. Actuator- Definition, characteristics, types and their working- LED, Relay, DC motor, stepper motor, servo motor, SSD etc.

Module IV: (07 hrs)
IoT PROTOCOLS- IoT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4 and Lora WAN, Connectivity and Communication Protocols: WiFi, Bluetooth, Zigbee, 6LoWPAN, Application Layer Protocols: CoAP, MQTT, HTTP/HTTPS.

Module V: (08 hrs)

Case Studies/Industrial Applications: IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipment's, Industry 4.0 concepts.

Text Books:

1. Embedded System by Raj Kamal, McGraw Hill Higher Education, 2008, Second Edition.
2. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017, First Edition.
3. Internet of Things – A hands-on approach, Arshdeep Bahga, Vijay Madisetti, Universities Press, 2015, First Edition.
4. Internet of Things: Architecture, Design Principles and Applications, Raj kamal, McGraw Hill Higher Education, 2017, First Edition.
5. Internet of Things: Principles and Paradigms, Rajkumar Buyya, Amir Vahid Dastjerdi (Elsevier), 2016, First edition.

Reference Books:

1. The Internet of Things – Key applications and Protocols, Olivier Hersent, David Boswarthick, Omar Elloumi and Wiley, 2012.
2. From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence",Jan Ho¨ ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand. David Boyle and Elsevier, 2014.
3. Architecting the Internet of Things, Dieter Uckelmann, Mark Harrison, Michahelles and Florian (Eds), Springer,2011.

List of Experiments

Mini Projects based on the following topics

1: IoT-Based Environmental Monitoring System

- Interface a DHT11 temperature and humidity sensor with ESP8266.
- Acquire environmental data and display it on the serial monitor and a web-based dashboard.
- Analyze real-time variations in temperature and humidity.

2: Smart Lighting Control Using Ambient Light Sensing

- Interface an LDR sensor and LED with a microcontroller.
- Design an automatic lighting control system that responds to ambient light intensity.
- Evaluate the application in smart homes and street-light automation.

3: IoT-Based Smart Irrigation System

- Interface a soil moisture sensor and relay-controlled water pump with ESP8266.
- Monitor soil moisture levels and automate irrigation based on predefined thresholds.
- Visualize sensor data using an IoT dashboard.

4: Smart Water Tank Level Monitoring System

- Interface an ultrasonic sensor for non-contact water level measurement.
- Develop a system for real-time monitoring of tank levels.
- Generate alerts and display water level information on a cloud platform.

5: IoT-Enabled Home Security and Intrusion Detection System

- Interface a PIR motion sensor and alarm module with ESP8266.
- Detect human movement and trigger local alarms.
- Send notifications to a remote user through an IoT platform.

6: Gas Leakage Detection and Alert System

- Interface an MQ-series gas sensor with a microcontroller.
- Monitor the presence of combustible or harmful gases.
- Generate local alarms and remote alerts for enhanced safety.

7: Smart Touchless Appliance Control System

- Interface IR or capacitive touch sensors with relay modules.
- Develop a contactless control system for electrical appliances.
- Implement remote monitoring and control using IoT connectivity.

8: Remote Monitoring and Device Control Using MQTT Protocol

- Configure an MQTT broker and MQTT client devices.
- Publish sensor data and subscribe to control commands.
- Implement remote control of LEDs or relays through MQTT communication.

9: Cloud-Based IoT Data Acquisition and Visualization

- Connect sensor nodes to cloud platforms such as ThingSpeak or Blynk.
- Upload real-time sensor data to the cloud.
- Visualize, analyze, and interpret collected data using graphical dashboards.

10: Development of a Wi-Fi-Based Smart Device Control System

- Configure ESP8266 as a web server.
- Design a browser-based interface for monitoring and controlling connected devices.
- Demonstrate wireless control of LEDs and household appliances.

11: IoT-Based Smart Parking Management System

- Interface IR or ultrasonic sensors for vehicle detection.
- Monitor parking slot occupancy in real time.
- Display parking availability information through a cloud dashboard.

12: Industrial Equipment Monitoring and Predictive Maintenance

- Monitor machine parameters such as temperature, vibration, or operational status.
- Acquire and transmit equipment data to a cloud platform.
- Demonstrate the concept of predictive maintenance using IoT technologies.

13: Smart home automation solution

- Design and implement an integrated smart home solution using multiple sensors and actuators.
- Enable real-time monitoring and remote control through a mobile application or cloud platform.
- Demonstrate automation features such as lighting control, security monitoring, and environmental sensing.

Course Code	25EE01TP0303			
Category	Programme Core Course			
Course Title	Machine Learning			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	III

Course Outcomes

On successful completion of the course, students will be able to:

1. Apply fundamental concepts, mathematical foundations, and data preprocessing techniques to analyze datasets and formulate machine-learning solutions for real-world applications.
2. Build predictive and descriptive machine learning models using supervised and unsupervised algorithms for regression, classification, clustering, and dimensionality reduction tasks.
3. Evaluate and interpret the performance of machine learning models using appropriate evaluation metrics, validation methods, and inference techniques to derive meaningful insights.
4. Apply advanced machine learning techniques including SVM, ensemble learning, neural networks, and basic reinforcement learning concepts to solve complex data-driven problems.
5. Optimize and design reliable, trustworthy machine learning systems for practical deployment using hyperparameter tuning, AutoML, explainability, and responsible AI principles.

Syllabus

Module I: (6 Hrs)

Foundations of Machine Learning: Introduction to Machine Learning: definition, evolution, key components, and applications. Learning paradigms: supervised, unsupervised, semi-supervised, and reinforcement learning. Foundations for ML including review of linear algebra, optimization, parametric vs non-parametric models, abstraction, generalization, model complexity, and interpretability. Exploratory Data Analysis (EDA), complete ML pipeline.

Module II: (9 Hrs)

Supervised Learning: Regression and Classification algorithms: Simple linear regression, multivariate regression, polynomial regression and nonlinear methods, exponential and logarithmic regression. Logistic Regression, Tree-based methods including Decision Trees. Bias-variance trade-off, overfitting and underfitting, regularization, gradient descent algorithm. Performance Metrics -- R^2, MAE, MSE, and RMSE, Accuracy, Precision, Recall, F1-score, Confusion Matrix, ROC & AUC Curves, Evaluation Measures, Cross-Validation techniques. Development, evaluation, and interpretation of models for real-world prediction tasks.

Module III: (6 Hrs)

Unsupervised Learning and Dimensionality Reduction Algorithms: Analysis of unlabelled data to discover hidden patterns and structures. Clustering techniques: K-Means, hierarchical clustering with dendrograms, DBSCAN, Gaussian Mixture Models, and fuzzy clustering. Cluster validation using Silhouette Score and Davies–Bouldin Index. Dimensionality reduction methods: Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for visualization, data compression, and improved model efficiency.

Module IV: (8 Hrs)

Advanced Machine Learning: Support Vector Machines and Kernel functions in SVM, advanced methods to overcome limitations of individual models. Ensemble learning: Bagging, Boosting, and Random Forests for improved accuracy and robustness. Artificial Neural Networks, perceptron models, feed-forward neural network architectures, and fundamental concepts of deep learning. Fundamentals of Reinforcement learning — agents, environments, states, actions, rewards, and trial-and error learning.

Module V: (6 Hrs)

Model Optimization, Responsible AI & Emerging Trends: Model selection and optimization: hyperparameter tuning using Grid Search and Random Search, feature selection, model pipelines, data preprocessing workflows, handling class imbalance, error analysis, and introduction to AutoML for automated model selection and optimization. Advanced validation strategies for reliable model comparison. Emerging topics: Explainable AI (XAI), adversarial machine learning (attacks and defenses), fairness, transparency, responsible AI, privacy-preserving machine learning, and real-world deployment case studies.

Text Book:

1. The Elements of Statistical Learning Data Mining, Inference, and Prediction by Trevor Hastie, Robert Tibshirani, Jerome Friedman, Second Edition, Springer, 2009.
2. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron, Third edition, O'Reilly Media, 2022.

Reference Books:

1. Pattern Recognition and Machine Learning by Christopher M. Bishop, First edition, Springer, 2006.
2. Machine Learning: A Probabilistic Perspective by Kevin P. Murphy, Francis Bach; MIT Press, 2012.
3. Understanding Machine Learning: From Theory to Algorithms by Shai Shale-Shwartz, and Shai Ben-David, Cambridge University Press, 2014.
4. Python Machine Learning by Sebastian Raschka and Vahid Mirjalili, Third edition, Packt Publishing, 2019.
5. Introduction to Machine Learning with Python by Andreas C. Müller and Sarah Guido, First edition, O'Reilly Media, 2016.
6. Reinforcement Learning: An Introduction by Richard S. Sutton and Andrew G. Barto, Second edition, MIT Press, 2018.

List of Experiments

Module I: Foundations and Data Preparation:

Lab 01: Data preprocessing and cleaning using pandas and NumPy.

Lab 02: Implement Simple Linear Regression from scratch to understand conceptual foundations.

Lab 03: Evaluate regression models using MAE and MSE metrics.

Lab 04: Perform bias and fairness analysis in machine learning models.

Lab 05: Exploratory analysis of machine learning pipeline components.

Module II: Supervised Learning:

Lab 06: Implement Simple Linear Regression using scikit-learn.

Lab 07: Implement Multiple Linear Regression using scikit-learn.

Lab 08: Implement Logistic Regression for binary classification.

Lab 09: Implement Decision Tree Classification.

Lab 10: Implement Support Vector Machine (SVM) for classification.

Lab 11: Evaluate supervised learning models using R^2, RMSE, Accuracy, and Confusion Matrix.

Module III: Unsupervised Learning:

Lab 12: Implement K-Means clustering using scikit-learn.

Lab 13: Perform Hierarchical Clustering with dendrogram visualization.

Lab 14: Apply DBSCAN clustering on noisy datasets.

Lab 15: Implement Gaussian Mixture Model (GMM) clustering.

Lab 16: Perform dimensionality reduction using PCA.

Lab 17: Validate clustering performance using Silhouette Score.

Module IV: Advanced Machine Learning:

Lab 18: Implement ensemble learning models: Bagging, Random Forest, and Boosting.

Lab 19: Build a basic Neural Network for classification.

Lab 20: Implement a simple Reinforcement Learning simulation.

Lab 21: Perform comparative evaluation of advanced machine learning models.

Module V: Optimization and Emerging Concepts:

Lab 22: Implement Cross-Validation techniques: K-Fold and Stratified K-Fold.

Lab 23: Perform hyperparameter tuning using GridSearchCV.

Lab 24: Perform model selection based on comparative performance analysis.

Lab 24: Simulate adversarial attacks and defense mechanisms.

Lab 25: Conduct fairness, accountability, and transparency analysis in machine learning systems.

Course Code	25EE01PR0304			
Category	Programme Core Course			
Course Title	Prompt Engineering			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	III

Module I: Introduction to Prompt Engineering

- Lab 1: What is Prompt Engineering, and why do we care?
- Lab 2: English as a new programming language

Module II: Getting Started with Prompt Engineering

- Lab 3: Getting to know our GPT-based AI tool
- Lab 4: The Naive Prompting Approach and the Persona Pattern
- Lab 5: The Interview Pattern

Module III: The Chain-of-Thought Approach

- Lab 6: The Chain-of-Thought Approach in Prompt Engineering

Module IV: Advanced Techniques

- Lab 7: The Tree-of-Thought Approach in Prompt Engineering
- Lab 8: Controlling Verbosity and the Nova System
- Lab 9: Getting to Know watsonx Prompt Lab

Module V: Capstone Project

Multidisciplinary Minor (MDM) Track-1: Integrated Circuit Design (IC design)

Course Code	25EE01TH0305-1			
Category	Multidisciplinary Minor -1			
Course Title	Basics of Chip Design using Verilog HDL			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	III

Course Outcomes:

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

1. Design and model combinational and sequential digital circuits using various modeling styles (Gate, Dataflow, Behavioral) in Verilog HDL.
2. Develop comprehensive testbenches and functional verification environments to validate the logic and timing of digital designs.
3. Construct synthesizable Verilog code by adhering to RTL coding guidelines for efficient hardware mapping by EDA synthesis tools.
4. Analyze critical timing parameters, including setup/hold times and propagation delays, to ensure the stability of synchronous digital systems.
5. Describe the FPGA implementation flow and the process of mapping digital designs onto hardware resources, including synthesis, placement, routing, and bitstream generation.

Syllabus:

Module I: Introduction to Digital Design and FPGA(07 Hrs)

VLSI Design Flow, FPGA Architectural Overview, Exploration of FPGA Development Boards, Introduction to Hardware Description Languages (HDL), Fundamental Language Elements, Syntax and Semantics**.**

Module II: Modeling Styles and Verification(07 Hrs)

Gate-level Modeling, Dataflow Modeling, Behavioral Modeling of Combinational Logic, Design of Multiplexers, De-multiplexers, Encoders, and Decoders, Implementation of Flip-flops and Counters, Development of Verification Testbenches, File I/O Operations for Automated Testing.

Module III: Arithmetic Circuit Architectures(07 Hrs)

Design and Performance Analysis of Combinational Blocks, Algorithm-to-Architecture Translation for Arithmetic Circuits, Design of Adders, Subtractors, Multipliers, and Dividers, Implementation of Shifters, Arithmetic Logic Units (ALU), and Comparators.

Module IV: Sequential Logic and State Machines (07 Hrs)

Advanced Design and Analysis of Synchronous Sequential Blocks, Finite State Machine (FSM) Design Methodologies, Mealy and Moore Machine Implementations, State Encoding and Optimization Techniques.

Module V: System-Level Design, Synthesis and Timing (07 Hrs)

Architecting the Datapath and Control Unit, Partitioning of Digital Systems, Micro-architecture Design Strategies, Practical Case Studies in Integrated System Design. Logic Synthesis Principles, Optimization Strategies for Area, Power, and Delay, Static Timing Analysis (STA),

Management of Setup and Hold Time Violations, Synthesis Flow for FPGA Platforms, Identification and Mitigation of Critical Path Delays.

Text Book:

1. Digital Logic Design Using Verilog Coding and RTL Synthesis, Vaibbhav Taraate, Springer India, 2016

2. Fundamentals of Digital Logic with Verilog; Stephen Brown and ZvonkoVranesic; McGraw Hill, 2nd Edition

Reference Books:

1. Digital Systems Design Using Verilog; Charles Roth, Lizy K. John, ByeongKil Lee; Cengage Learning 2nd Edition
2. Verilog HDL: A Guide to Digital Design and Synthesis; Samir Palnitkar, Prentice Hall PTR; 2nd Edition

Multidisciplinary Minor (MDM) Track-2: IoT for Environmental Sustainability

Course Code	25EE01TH0305-2			
Category	Multidisciplinary Minor-1			
Course Title	Introduction to IoT System Design			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	III

Course Outcomes:
At the end of this course, students will be able to:
1) Implement the state of the Architecture of an IoT system
2) Design methodology and circuits using hardware platforms involved in IoT.
3) Analyze and organize the IoT data.
4) Compare IOT Applications in Industrial & real world.

Module 1:
Introduction to IoT: - Introduction to Internet of Things (IoT), Functional Characteristics, Architecture: 3 layer/5 layer models, Recent Trends in the Adoption of IoT, Role of cloud in IoT, Main design principles and needed capabilities, M2M and IoT Technology, Societal Benefits of IoT.

Module 2:
IoT PROTOCOLS- IoT Access Technologies: Physical and MAC layers, topology and Security of IEEE 802.15.4 and Lora WAN, Connectivity and Communication Protocols: WiFi, Bluetooth, Zigbee, 6LoWPAN, Application Layer Protocols: CoAP, MQTT, HTTP/HTTPS.

Module 3:
Development boards for IoT and Platforms: -
Introduction to Arduino, Raspberry Pi, ESP8266, - Their versions, models, physical design, specifications, GPIO pin structure, OS requirement, capabilities, reading datasheet, C/C++ for Arduino/ESP32, Python for RPi.

Module 4:
Sensing and Actuation: -Role of Sensors and Actuators in IoT
Sensors- Definition, characteristics, types and their working- Soil moisture sensor, DHT11, Ultrasonic sensor, IR sensor, PIR sensor, sound sensor, touch sensor, LDR, Gas sensor, etc.
Actuator- Definition, characteristics, types and their working- LED, Relay, DC motor, stepper motor, servo motor, SSD etc.

Module 5:
Case Studies/Industrial Applications: IoT applications in home, infrastructures, buildings, security, Industries, Home appliances, other IoT electronic equipment's, Industry 4.0 concepts.

Text Books:
1. IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things, David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton and Jerome Henry, Cisco Press, 2017
2. Internet of Things – A hands-on approach, Arshdeep Bahga, Vijay Madisetti, Universities Press, 2015

3. Internet of Things: Architecture, Design Principles and Applications, Rajkamal, McGraw Hill
HigherEducation
4. “Internet of Things: Principles and Paradigms”, Rajkumar Buyya, Amir Vahid Dastjerdi (Elsevier)

Reference Books:

1. The Internet of Things – Key applications and Protocols, Olivier Hersent, David Boswarthick, Omar Elloumi and Wiley, 2012 (for Unit2).
2. “From Machine-to-Machine to the Internet of Things – Introduction to a New Age of Intelligence”,Jan Ho¨ ller, VlasiosTsiatsis, Catherine Mulligan, Stamatis, Karnouskos, Stefan Avesand. David Boyle and Elsevier, 2014.
3. Architecting the Internet of Things, Dieter Uckelmann, Mark Harrison, Michahelles and Florian (Eds), Springer,2011.

Course Code	25HS01TP0301			
Category	Basic Science Course			
Course Title	Environmental Science			
Scheme & Credits	L	P	Credits	Semester
	1	2	2	III

Course Outcomes

After successful completion of this course, students will learn to;

CO1:Explain sustainability principles, climate change impacts, and environmental regulations, and analyze the role of data analytics in addressing global environmental challenges.

CO2:Apply concepts of green computing and circular economy to evaluate IT infrastructure, e-waste management, and emerging environmental impacts of modern technologies.

CO3: Apply chemical principles and analytical techniques to safely analyze solutions and environmental samples for sustainable resource management and public health.

Module I: Sustainability & Digital Transformation

Sustainability Engineering: Introduction to the five pillars of sustainability and the 17 UN Sustainable Development Goals (SDGs).

Climate Change & Big Data: The role of data analytics in predicting climate patterns and tracking carbon footprints, Ozone layer depletion, Global warming and Air pollution.

Environment-Related Legislation: Digital compliance, Carbon Credits, Carbon sink, carbon taxes, and international agreements like the Paris Accord in the context of IT industries.

Module II: Green Computing & Circular Economy

Green IT Infrastructure: Energy-efficient data centers, green networking, and "Sustainable AI" (reducing the carbon cost of training large models).

E-Waste & Circularity: Life cycle assessment (LCA) of hardware; "Right to Repair" movement, waste management, recycling and circular economy.

Emerging Threats: Environmental impact of mining materials for high-tech devices (Nano-materials & Micro-plastics).

Text Books:

1.Shikha Agrawal, Engineering Chemistry: Fundamentals and Applications, Cambridge University Press.

2. Dr. Rajshree Khare, A Textbook of Engineering Chemistry (AICTE), S.K. Kataria & amp; Sons.

3. S. S. Dara, A Textbook of Engineering Chemistry, S. Chand Publications.

4. M Afshar Alam, Sapna Jain, Hena Parveen, Green Computing Approach Towards Sustainable Development, Wiley Interscience Publications.

Reference Books:

1. E-waste recycling and management: present scenarios and environmental issues, Khan, Anish, and Abdullah M. Asiri. 2019, Springer, Vol. 33. ISBN: 978-3-030-14186-8.

2. Hans-Eckhardt Schaefer, Nanoscience: The Science of the Small in Physics, Engineering, Chemistry, Biology and Medicine, Springer-Verlag Berlin Heidelberg.

List of Experiments:

Any Eight Experiments from the following:

1) Safe Handling of Hazardous Chemicals for Sustainable Communities (Handling hazardous chemicals, MSDS, waste minimization, disposal) SDGs: 3, 12, 13
2) Preparation of Standard Solutions for Sustainable Water Chemistry (Preparation of molar/normal/percent solutions and concentration determination) SDGs: 4, 9
3) Statistical Analysis of Acid–Base Neutralization for Green Decision-Making (Neutralization of acid against base and graph preparation using software) SDGs: 4, 9, 12, 13
4) Copper Estimation from PCB E-Waste for Resource Recovery (Estimation of Copper ions from acid-digested PCB solution) SDGs: 12, 14
5) Chromium Estimation from E-Waste for Water and Health Safety (Estimation of Chromium ions from e-waste sample) SDGs: 3, 6, 12
6) Estimation of Fe (II) Ions from E-Waste Rust for Sustainable Resource Management (Fe (II) ions using 1,10-Phenanthroline method) SDGs: 12, 15, 3, 9
7) Analysis of Beverage Acidity for Consumer Health Safety and Sustainable Production Compliance (Aligns with SDG 3 & 12)
8) Determination and Computational Analysis of AQI Data for Indian Cities; SDG: 3, 11, 13, 9
9) Estimation of Dissolved Oxygen in Wastewater for Clean Water Sustainability; SDG: 3, 11, 14, 6
10) Data-Driven Modeling of Organic Pollutant Adsorption on Activated Charcoal for Sustainable Wastewater Treatment

Suggested Books/Reference Books:

(1) S. S. Dara, A Textbook on Experiments and Calculations in Engineering Chemistry, S. Chand Publications.
(2) J. B. Yadav, Advanced Practical Physical Chemistry, Krishna's Prakashan Media (P) Limited.
(3) A. J. Elias, Collection of Interesting General Chemistry Experiments, Universities Press Publications.
(4) V. K. Ahluwalia, S. Dhingra and A. Gulati, College Practical Chemistry, Universities Press Publications.

Course Code	25EE01HT0301-1			
Category	Honors			
Course Title	Industrial Automation			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	III

Unit 1: Fundamentals of Industrial Measurement

Measurement system, need of instrumentation in industry, static and dynamic characteristics, accuracy, precision, sensitivity, resolution, linearity, hysteresis, repeatability, calibration, drift, loading effect, errors in measurement, measurement standards, industrial signal ranges, voltage signal, current signal, 4–20 mA current loop, 0–10 V signal, two-wire, three-wire and four-wire transmitters.

Unit 2: Industrial Sensors and Transducers

Temperature measurement using RTD, thermocouple, thermistor, infrared sensors; pressure measurement using strain gauge, capacitive and piezoelectric sensors; flow measurement using orifice meter, venturi meter, rotameter, turbine flow meter, electromagnetic flow meter and ultrasonic flow meter; level measurement using float, capacitive, ultrasonic and radar level sensors; proximity sensors, photoelectric sensors, inductive and capacitive sensors.

Unit 3: Signal Conditioning and Industrial Interfacing

Bridge circuits, Wheatstone bridge, instrumentation amplifier, isolation amplifier, filters, noise reduction, shielding, grounding, signal isolation, analog-to-digital conversion, digital-to-analog conversion, sampling, multiplexing, sensor interfacing with microcontroller and PLC, industrial signal conditioning modules, loop-powered transmitters, signal converters and isolators.

Unit 4: Smart Instruments and Industrial Communication

Smart transmitters, HART basics, field instruments, intelligent sensors, calibration of smart instruments, distributed measurement systems, sensor networks, industrial data communication basics, Modbus-enabled instruments, wireless sensor basics for industrial applications, data logging and remote monitoring of field parameters.

Unit 5: Calibration, Testing and Instrumentation Practices

Calibration procedure, zero and span adjustment, dead weight tester, temperature bath, pressure calibrator, loop calibrator, documentation of calibration, instrument datasheets, sensor selection criteria, installation practices, cable selection, panel wiring basics, safety practices in instrumentation, case studies of instrumentation in boiler, water treatment plant, chemical process plant and manufacturing line.

Course Code	25EE01HT0301-2			
Category	Honors			
Course Title	Introduction to DevOps and Version Control			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	III

Course outcome

On successful completion of course students will be able to

CO1: Understand DevOps concepts, culture, and lifecycle.

CO2: Develop shell scripts to automate administrative and DevOps tasks.

CO3: Apply Git for source code management and version control.

CO4: Demonstrate the use of Git workflows, including branching, merging, rebasing, and conflict resolution in collaborative development environments.

CO5: Utilize repository hosting platforms such as GitHub or GitLab for team collaboration, pull requests, code reviews, and project management.

Syllabus

Module I :Foundations of DevOps (7 Hrs)

Evolution of Software Development, Traditional Development vs Agile vs DevOps, DevOps Principles and Culture , DevOps Lifecycle , Benefits and Challenges of DevOps Roles and Responsibilities in DevOps Teams

Module II: Linux for Devops (7 Hrs)

User and Group Management, Password Policies, Environment Variables , Process Management , Job Scheduling , System Services and Daemons , Systemd Fundamentals , Resource Monitoring ,Input and Output Handling , Text Processing Tools (grep, awk, sed) , Automation Scripts for DevOps Tasks, Log File Analysis

Module III: Linux Networking and Security (7 Hrs)

Linux Networking Fundamentals, IP Addressing and DNS , Network Configuration ,SSH and Remote Access , Linux Firewall Configuration , Security Best Practices ,Access Control and Sudo ,SSL/TLS Concepts ,Security Monitoring, SCP and Rsync , Network Troubleshooting Tools

Module IV: Version Control in DevOps and Project Management (7 Hrs)

Introduction to Version Control , Centralized vs Distributed Version Control Systems ,Git Architecture and Workflow , Repository Creation and Management , Branching, Branch Protection Rules Merging, and Rebasing ,Conflict Resolution and Best Practices

Module V: Collaborative Development with GitHub/GitLab (7 Hrs)

Remote Repositories , GitHub/GitLab Features , Forking and Cloning Repositories ,Pull Requests and Merge Requests ,Code Review Practices ,team Collaboration and Repository Management

Text Books

1. Pro Git
   Authors: Scott Chacon and Ben Straub
   Publisher: Apress

Reference Books

DevOps Handbook
Publisher: IT Revolution Press

Course Code	25EE01MT0301				
Category	Minor Course				
Course Title	IoT fundamentals				
Scheme & Credits	L	T	P	Credits	Semester
	3	0	0	3	III

Course Outcomes

After learning the course, the student will be able to:

1. Understand the basics of networking
2. Gain the knowledge about iot standards
3. Realize the basic applications using arduino and raspberry pi
4. Illustrate different real world applications syllabus

Syllabus

Module - I: (7Hrs)

Basics of Networks, TCP/IP model,IP Addresses, application layer protocols, HTTP, MQTT, WWW, constraint application protocol,stacks.

Module - II: (7Hrs)

Introduction to IoT, evolution of IoT, IoT and SCADA, Big Data, IoT Standards, requirement, Platforms, relevance of IoT, security

Module - III: (7Hrs)

Interoperability in IoT, Machine-to-Machine Communications, Introduction to Arduino Programming, Integration of Sensors and Actuators with Arduino, Sensing, Actuation, Sensor Networks

Module - IV: (7Hrs)

Introduction to Python programming, Introduction to Raspberry Pi, Implementation of IoT with Raspberry Pi.

Module - V: (7Hrs)

Introduction to SDN, Fog Computing, IoT application case studies: Smart Cities and Smart Homes, Connected Vehicles, Smart Grid, Industrial IoT, Agriculture, Healthcare, Activity Monitoring, IoT in India: Smart India projects, Challenges in IoT

Text Books

1. Computer Networks: A Top-Down Approach; Behrouz A Forouzan, Firouz Mosharraf, McGraw Hill Education. Special Indian Edition 2012
2. Arduino Cookbookby Michael Margolis, O'Reilly Media, Inc., 1st edition
3. Raspberry pi Cookbook by Simon Monk, O'Reilly Media, Inc., 3rd edition.

Syllabus of IV semester

Course Code	25EE01TP0401			
Category	Programme Core Course			
Course Title	Deep Learning			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	IV

Course Outcomes:

On successful completion of the course, students will be able to:

1. Design and implement deep neural network models using appropriate architectures, and vectorized computation techniques for solving supervised learning problems.
2. Design and implement convolutional neural network-based solutions for visual recognition tasks, and analyze the effect of architectural parameters on model performance using suitable evaluation measures.
3. Apply suitable training strategies to improve the convergence, performance, and generalization of deep learning models.
4. Analyze bias–variance trade-offs, model behavior, model design choices, and performance measures to improve deep learning models for different applications.
5. Analyze and compare sequence models, attention mechanisms, and Transformer-based approaches for solving sequential data and natural language processing problems.
6. Build and critically assess an end-to-end deep learning application by applying the basic steps of data preparation, model selection, training, evaluation, and deployment while considering ethical issues in AI.

Syllabus

Module I: (6 Hrs)

Introduction to Deep Learning: Evolution from Machine Learning to Deep Learning, Perceptron, Multi-Layer Perceptron (MLP), Sigmoid Neurons, Supervised Learning with Neural Networks, Activation Functions (ReLU, Sigmoid, Tanh), Loss functions (MSE, Cross-Entropy), Feedforward Neural Networks, Computation Graph, Vectorization, Gradient Descent (GD) for Neural Networks, Backpropagation algorithm.

Module II: (6 Hrs)

Training Deep Neural Networks: Challenges in training deep networks, Weight initialization (Xavier, He), Learning rate selection and its impact, Optimization Algorithms: Batch vs Mini-batch vs Stochastic Gradient Descent, Momentum Based GD, Nesterov Accelerated GD, AdaGrad, RMSProp, Adam. Regularization: L2 regularization, Dropout, Early stopping, Bias-variance tradeoff, Overfitting/Underfitting.

Module III: (7 Hrs)

Fundamentals of Convolutional Neural Networks: Convolution operation, feature maps, Padding, stride, Pooling layers, Convolutional Neural Networks, LeNet, AlexNet, VGGNet, GoogLeNet, ResNet, Image classification. Batch Normalization, Layer Normalization, and Transfer learning: feature extraction vs fine-tuning, Pre-trained models and their applications.

Module IV: (6 Hrs)

Advanced Applications of CNN: Object localization vs object detection, Two-stage detectors: R-CNN, Fast R-CNN, Faster R-CNN (evolution overview only), One-stage detectors: YOLO,

Evaluation metrics: Intersection over Union (IoU), mean Average Precision (mAP), Applications in real-world systems. Introduction to Image segmentation, Applications in medical imaging and autonomous systems.

Module V: (7 Hrs)
Sequence Models:
Sequential data and time dependencies, Recurrent Neural Networks (RNN), Backpropagation Through Time (BPTT), Vanishing and Exploding Gradient problem. LSTM, GRU (concept + working). Applications: Text classification. Introduction to Attention & Transformers: Basic Attention Mechanism, Transformer architecture (high-level only), Self-attention concept, Applications in NLP.

Module VI: (03 Hrs)
Introduction to Deep Learning Applications: Overview of DL applications (CV, NLP, Speech), End-to-end Deep Learning pipeline: Data → Model → Training → Evaluation → Deployment. One case study example. Ethical considerations in AI.

Text Book:
1.Understanding Deep Learning by Simon J. D. Prince, The MIT Press, 2023.
2.Deep Learning by Ian Goodfellow, Yoshua Bengio, Aaron Courville, & Francis Bach, MIT Press, 2017.

Reference Books:
1. Deep Learning: A Practitioner's Approach by Adam Gibson and Josh Patterson, Shroff/O'Reilly; First Edition, 2017.
2. Dive into Deep Learning by Aston Zhang, Zachary C. Lipton, Mu Li , Alexander J. Smola, Cambridge University Press, 2024
3. Neural Networks and Deep Learning: A Textbook" by Charu C. Aggarwal, Springer International Publishing, 2018.
4. Deep Learning Illustrated by Jon Krohn, Grant Beyleveld, and Aglaé Bassens, Addison-Wesley; 1st edition, 2019.
5. Recent Research Papers from Reputed Journals and Conferences such as CVPR, ICLR, NIPS,
ICML, PAMI etc.

List of Experiments

Lab-01: Implement and use gradient descent with backpropagation for a classification task using a single neuron (logistic Regression).

Problem Statement: Implement a feedforward neural network and write the backpropagation code for training the network. We strongly recommend using numpy for all matrix/vector operations. You are not allowed to use any automatic differentiation packages. This network will be trained and tested using the Fashion-MNIST dataset. Specifically, given an input image (28 x 28 = 784 pixels) from the Fashion-MNIST dataset, the network will be trained to classify the image into 1 of 10 classes.

Lab-02: Implement and use gradient descent with backpropagation for a classification task using a fully connected Neural Network with one hidden layer.

Problem Statement: Implement a fully connected feedforward neural network with one hidden layer from scratch to perform a classification task. Train the model using gradient descent with backpropagation, and evaluate its performance in terms of accuracy and convergence.

Lab-03: Implementation of an L-layer Feedforward Neural Network using Backpropagation for a Classification Task.

Problem Statement: Implement a feedforward neural network with multiple hidden layers (L-layer architecture) from scratch using NumPy (or equivalent), and train it using gradient descent with backpropagation for a classification task such as binary or multi-class classification.

Lab-04: Implementation of a Convolutional Neural Network (CNN) from Scratch for Image Classification.

Problem Statement: Implement a Convolutional Neural Network (CNN) from scratch using a suitable programming language (e.g., Python with NumPy) to perform image classification. The model should include convolution, activation, pooling, and fully connected layers, and be trained using backpropagation and gradient descent.

Fine-tuning a pre-trained model: In most DL applications, instead of training a model from scratch, you would use a model pre-trained on a similar/related task/dataset. From torchvision, you can load **ANY ONE** model (GoogLeNet, InceptionV3, ResNet50, VGG, EfficientNetV2, VisionTransformer etc.) pre-trained on the ImageNet dataset. Given that ImageNet also contains many animal images, it stands to reason that using a model pre-trained on ImageNet maybe helpful for this task. You will load a pre-trained model and then fine-tune it using the naturalist data that you used in the previous question. Simply put, instead of randomly initializing the weights of a network you will use the weights resulting from training the model on the ImageNet data (torchvision directly provides these weights). Please answer the following questions:

Lab-05: Implementation of Convolutional Neural Network (CNN) using TensorFlow/PyTorch for Image Classification.

Problem Statement: Implement a Convolutional Neural Network (CNN) using TensorFlow or PyTorch to perform image classification on a standard dataset such as MNIST or CIFAR-10. Train the model and analyze its performance in terms of accuracy. Study the effect of varying kernel sizes and pooling operations on the model performance.

Lab-06: Implementation and Comparative Analysis of Optimization Algorithms for Neural Network Training.

Problem Statement: Implement and compare different optimization algorithms such as Stochastic Gradient Descent (SGD), Momentum, RMSProp, and Adam for training a neural network on a classification task. Analyze and compare their convergence behavior by plotting loss versus epochs and evaluating model performance.

Lab-07: Application of transfer learning using pre-trained convolutional neural network models, with performance evaluation on a custom dataset.

Lab-08: Analysis of the impact of data augmentation techniques on model generalization and performance.

Lab-09: Implementation of object detection using pre-trained models, followed by analysis of detection performance on images or video.

Lab-10: Implementation of image segmentation models with visualization of segmented outputs.

Lab-11: Regularization and Hyper parameter Tuning

a) Analyze the bias-variance tradeoff and implementing L2 regularization in neural networks.
b) Experiment with early stopping and dataset augmentation techniques.
c) Implementing dropout regularization and comparing its effectiveness with other techniques.
d) Explore hyper parameter tuning strategies for optimizing neural network performance.

Lab-12: Deployment of deep learning model:

a) Deploy the pertained model directly on edge devices (e.g., smartphones, IoT devices) to assess real-time performance and resource constraints.

b) Deploy the model on cloud platforms (e.g., AWS, Google Cloud, Azure) using services like AWS Lambda, Google Cloud Functions, or Kubernetes clusters. Evaluate scalability, cost, and ease of deployment.

c) Use Docker or similar containerization tools to encapsulate the model and its dependencies. Compare deployment and runtime performance with non-containerized approaches.

Lab-13: Implementation of Sequence Modeling using RNN/LSTM for Text Classification or Sequence Prediction.

Problem Statement: Implement a Recurrent Neural Network (RNN) or Long Short-

Term Memory (LSTM) network using PyTorch or Keras to perform a sequence modeling task such as text classification or sequence prediction. Train the model and analyze its performance, and study the effect of varying sequence length on the model's accuracy and learning behavior.

Lab-14: Implementation of End-to-End Deep Learning Pipeline.

Problem Statement: Develop an end-to-end deep learning pipeline for a chosen task (image or text classification) by performing dataset preprocessing, model development, training, and evaluation. Document each stage of the pipeline and analyze the overall model performance.

Lab-15: Mini-Project: Deep Learning-Based Solution for a Real-World Problem.

Problem Statement: Design and implement a deep learning-based solution for a real-world problem such as image or text classification. The solution should include data preprocessing, model selection, hyper parameter tuning, and performance evaluation, followed by analysis of results.

Course Code	25EE01TP0402			
Category	Programme Core Course			
Course Title	Computer Network			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	IV

Course Outcomes:

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

1. Explain networking fundamentals including TCP/IP, OSI model, network architectures, and modern communication technologies.

2. Apply data link layer techniques such as framing, CRC, flow control, and MAC protocols to evaluate network performance.

3. Design and analyse IP addressing (IPv4/IPv6, subnetting, VLSM) and routing protocols like OSPF and BGP.

4. Analyze transport and application protocols (TCP/UDP, QUIC, DNS, HTTP/HTTPS) and evaluate network services and security mechanisms.

5. Evaluate modern networking technologies including SDN, NFV, and AI/ML-based network management

Syllabus

Module I: (6Hours)
Module 1. TCP/IP Protocol Suite, OSI Reference Model, LAN/WAN/MAN architectures, Physical Media (Fiber, Wi-Fi 6/7, 5G Slicing), Packet vs Circuit Switching

Module II: (6 Hrs)
Module 2. Framing, Error Control (CRC), Flow Control, CSMA/CD vs CSMA/CA, VLAN Segmentation, EtherChannel (LACP), Spanning Tree (RSTP)

Module III: (6 Hrs)
IPv4/IPv6 Addressing, Subnetting/VLSM, ICMP, Routing Logic, OSPF (Single/Multi-area), BGP for Enterprise, SD-WAN concepts

Module IV: (11 Hrs)
TCP 3-way Handshake, UDP, Port Multiplexing, Congestion Control Algorithms (BBR/Cubic), QUIC Protocol (HTTP/3), Load Balancing (L4 vs L7), DNS, DHCP, HTTP/HTTPS, FTP, SMTP, API Gateways, RESTful Web Services, Zero Trust Network Access (ZTNA) Basics, SSH Security.

Module V: (6 Hrs)
SDN Architecture (Control/Data plane), OpenFlow vs API-centric networking, Network Function Virtualization (NFV), AI in Networking: AOLs, ML for Traffic Classification, Predictive Troubleshooting

Textbook

1. Data Communications and Networking with TCPIP Protocol Suite; Behrouz A. Forouzan, McGraw-Hill Education, 6th Edition, 2022

2. Computer Networking: A Top-Down Approach; James F. Kurose, Keith W. Ross, Pearson Education, 8th Edition, 2021.

Reference Book

1. Computer Networks; Andrew S. Tanenbaum, Pearson Education, 5th Edition, 2011.
2. Computer Networking Problems and Solutions; Russ White, Pearson / Cisco Press, 2018.
3. SDN: Software Defined Networks; Thomas D. Nadeau, O'Reilly Media, 2013.

List of Experiments

Experiment 1: Network Models Study (CO1)
Objective:
Study TCP/IP and OSI layers using simulation.
Activity:

- Identify the functions of each layer.
- Map common network protocols to the corresponding layers.

Experiment 2: LAN Setup and Packet Analysis (CO1)
Objective:
Configure a basic Local Area Network (LAN) and analyze packet flow.
Activity:

- Create a LAN using switches and PCs.
- Capture and analyze network packets.

Experiment 3: Framing and Error Detection (CRC) (CO2)
Objective:
Implement Cyclic Redundancy Check (CRC) for error detection.
Activity:

- Develop a program in C or Python for CRC generation and verification.
- Test error detection capabilities using sample data.

Experiment 4: MAC Protocol Simulation (CO2)
Objective:
Analyze the performance of CSMA/CD and CSMA/CA protocols.
Activity:

- Simulate network collisions.
- Observe and compare backoff algorithms.

Experiment 5: VLAN and Switching (CO3)
Objective:
Configure VLANs and trunking in a switched network.
Activity:

- Create VLAN segmentation.
- Configure inter-VLAN routing.
- Verify communication between VLANs.

Experiment 6: IP Addressing and Subnetting (CO3)
Objective:
Perform IPv4/IPv6 addressing and Variable Length Subnet Masking (VLSM).
Activity:

- Design an efficient subnetting scheme.
- Assign IP addresses.

- Verify network connectivity.

Experiment 7: Routing Protocol Configuration (CO4)
Objective:
Configure and analyze OSPF routing.
Activity:
- Configure single-area OSPF.
- Configure multi-area OSPF.
- Analyze routing table updates.

Experiment 8: Transport Layer Analysis (CO4)
Objective:
Analyze TCP and UDP protocols, including the TCP three-way handshake.
Activity:
- Capture transport layer packets.
- Study TCP connection establishment and termination.
- Compare TCP and UDP communication.

Experiment 9: Application Layer Protocols (CO4)
Objective:
Configure and analyze DNS, HTTP, and FTP services.
Activity:
- Configure application layer services.
- Simulate client-server communication.
- Verify protocol functionality.

Experiment 10: Network Security Basics (CO5)
Objective:
Study Secure Shell (SSH) and basic network security mechanisms.
Activity:
- Configure secure remote login using SSH.
- Analyze authentication and encryption mechanisms.

Tool Summary
Simulation Tools
1. NetSim
2. Cisco Packet Tracer
3. GNS3

Course Code	25EE01TP0404			
Category	Programme Core Course			
Course Title	Database Management system			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	IV

Objective: To introduce fundamental concepts of Computing System, Database system, Database Model , Database Management System, SQL queries , usage of DBMS using SQL/PL SQL

Course Outcomes:

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

1. Apply the basic concepts of Database Systems and Applications
2. Use the basics of SQL and construct queries using SQL in database creation and interaction
3. Design a commercial relational database system (Oracle, MySQL) by writing SQL using the system

Syllabus

Module I: (5 Hrs)

Overview of Database systems: Basic concepts Database & Database Users, File System vs. DBMS, Database System Concepts & Architecture, Data models, Schemas & Instances, Structure of Relational Database, The Relational Algebra-Fundamental operators and syntax, Extended Relational Algebra operations

Module II: (7 Hrs)

Entity-Relational Model: Basic concepts, Constraints, Keys, Design Issues, Entity-Relational ship Diagram, Weak Entity Sets, Extended E-R Features, Design of an E-R Features, Transforming ER Model to Relational Data Model.

Module III:(7 Hrs)

Introduction to SQL: Basic Structure, DDL, DML, DCL, structure-creation, alteration, defining constraints- Primary Key, foreign key, unique key, not null, check, IN operator, Set Operations Aggregate Functions, Null Values, Nested Sub-queries, Views, Complex queries, Modification of Database, Joined Relations, Data-Definition Language, Dynamic SQL.

Module IV: (6 Hrs)

Indexing and Hashing: Basic Concepts, Ordered Indices, B+ - Tree Index Files, B-Tree Index Files, Static Hashing, Dynamic Hashing, Comparison of Ordered Indexing and Hashing, Index Definition in SQL, Multiple-Key Access.

Module V: (6 Hrs)

Transaction Processing: Transactions, ACID Properties, Transaction Schedules & Types, Serializability, Conflict – Serializability, View – Serializability, Testing for Serializability, Concurrency Control: Lock-based Protocols, Time Stamp-based Protocols, Enforcing, Different Locking Modes, 2PL (Two Phase Locking protocol), Multiple Granularity.

Modules VI: (4 Hrs)

Introduction to Advances in Databases: Object-Oriented Databases, Web Databases, Data Warehousing and Mining, Parallel Databases, Distributed Databases.

Text Book:

1. Database System Concepts by Abraham Silberschatz, Henry F. Korth and S Sudershan, McGraw Hill Education, 6th Edition, 2013
2. Database Management System by Raghu Ramkrishna and Johannes Gehrke, McGraw Hill Education 3rd Edition 2014.
3. SQL-PL/SQL the Programming Language of Oracle by Ivan Bayross, BPB Publications 4th Edition 2017.

Reference Book:

1. Fundamentals of Database Systems by Ramez Elmasri and Shamkant Navathe, Addision Wesley 6th Edition 2010
2. PL/SQL Programming: Oracle Programming 11g by Michale McLaughlin, Oracle Press, McGraw Hill Publications, 2013

List of Experiments that can be conducted in Oracle RDBMS/Live SQL server

1. Introduction to SQL
2. Installation and usage of Oracle RDBMS/Live SQL server
3. Create tables, Inserting Data, Creating queries
4. Creating triggers, indexes
5. Creating views,
6. Deleting data, table, views
7. An introduction to analytic functions.
8. Creating Analytic Views
9. Joining Tables
10. Sorting and Limiting Rows
11. Union, Minus, and Intersect
12. Merge: Databases
13. Case studies

a. Simplified and Fast Fraud Detection using SQL Pattern Matching
b. Creating an Analytic View for the Sales History (SH) Sample Schema
c. Update & Transactions in Databases
d. Mortgage calculator with the SQL MODEL Claus
e. Database to perform Advanced DML, PL/SQL, Triggers, Procedure and Functions.

Course Code	25EE01TP0404			
Category	Programme Core Course			
Course Title	Digital Signal Processing			
Scheme & Credits	L	P	Credits	Semester
	3	2	4	IV

Course Outcomes:

On successful completion of the course, students will be able to:

- **CO1:** Apply fundamental concepts of continuous-time and discrete-time signals and systems, including sampling and signal transformations, for digital representation of real-world signals.
- **CO2:** Analyze discrete-time LTI systems using Z-transform, pole-zero representation, and frequency-domain techniques.
- **CO3:** Implement and evaluate DFT and FFT algorithms for spectral analysis and efficient digital signal processing applications.
- **CO4:** Design and realize FIR and IIR digital filters using standard techniques and evaluate their performance for practical signal processing applications.
- **CO5:** Apply spatial- and frequency-domain image processing techniques for image enhancement and edge detection applications.

Syllabus

Module I: Fundamentals of Signals and Systems:

Introduction to continuous-time and discrete-time signals and systems, classification of signals (energy/power, periodic/aperiodic, deterministic/random), basic operations on signals, elementary signals, system properties (linearity, causality, time invariance, stability), convolution of continuous and discrete signals, correlation, impulse and step response of LTI systems. Frequency-domain representation of signals, sampling theorem, aliasing, reconstruction of signals, quantization and A/D–D/A conversion fundamentals.

Module II: Z-Transform and Discrete-Time System Analysis:

Need for Z-transform, relation with Fourier and Laplace transforms, Region of Convergence (ROC), properties of Z-transform, pole-zero plots, system function representation, analysis of causality and stability of LTI systems, difference equations, Inverse Z-transform, partial fraction expansion, power series method, frequency response of discrete-time systems, realization of digital systems.

Module III: DFT and FFT Algorithms Discrete-Time Fourier Transform (DTFT), frequency domain sampling, Discrete Fourier Transform (DFT), DFT properties, circular convolution, linear convolution using DFT, computation efficiency, Fast Fourier Transform (FFT), Decimation-In-Time (DIT), Decimation-In-Frequency (DIF), Radix-2 FFT algorithms. Computational complexity and applications of FFT in signal analysis.

Module IV: IIR Filter Design and Realization Analog filter approximations (Butterworth, Chebyshev), IIR design via Impulse Invariance, Bilinear Transformation Method (BTM), frequency warping, IIR realization structures, Direct Form I, Direct Form II, Cascade form, Parallel form, finite word-length effects.

Module V: FIR Filter Design: FIR vs IIR comparison, Linear phase FIR filters, symmetry and antisymmetric conditions, Gibbs phenomenon, windowing techniques (Rectangular, Hamming, Hanning, Blackman), frequency sampling method, FIR realization, Direct form, Cascade form, Lattice structures.

Module VI: Fundamentals of Digital Image Processing:
Introduction to digital image processing, image acquisition and representation, basic properties of images, point processing operations and gray-level transformations. Histogram processing and histogram equalization techniques for image enhancement.
Spatial domain filtering techniques, 2D convolution, smoothing and sharpening filters. Edge detection fundamentals. Frequency-domain image enhancement, 2D Fourier Transform, 2D FFT for image processing, frequency-domain filtering, low-pass and high-pass filters, and basic applications of frequency-domain enhancement.

Lab Experiments:

1. Design a MATLAB script to synthesize a musical chord by combining multiple sinusoidal sequences. Use the soundsc() function to play the result. Observe how changing the phase and amplitude of harmonics changes the "timbre" of the sound.
2. Capture a high-frequency pure tone (e.g., 15 kHz) and "under sample" it below the Nyquist rate (f_s < 2f_{max}). Use MATLAB to visualize how the signal "folds" into a lower frequency (the ghost frequency).
3. Drone flight controllers and industrial robotics: Given the transfer function of a PID controller for a self-balancing "Segway" robot, map the poles and zeros in the z-plane. Move a pole outside the Unit Circle and simulate the "catastrophic failure" (divergence) of the system.
4. Digital signal processing in smart speakers (Alexa/Google Home) to handle room acoustics- Implement a Recursive (IIR) system that simulates a "Room Impulse Response." Pass a dry voice recording through the system to create a realistic "Cathedral Reverb" effect.
5. Take a noisy ECG signal (simulated) contaminated with 50 Hz power-line interference. Use the FFT to identify the noise peak and apply a "Spectral Subtraction" technique to recover the heartbeat. Students create a "Live Spectrogram" that reacts to their own voice via the computer microphone.
6. Design a 10-band Graphic Equalizer. Students must create a set of FIR filters (using fir1 or kaiser windows) that allow them to boost the "Bass" or "Treble" of a song while keeping the phase linear to avoid distortion. Compare the computational "cost" (number of multiplications) of a high-order FIR vs. a low-order IIR for the same task.
7. Pre-process a dataset of "Human Activity" (Accelerometry data from a phone). Extract features like Mean, Variance, and Dominant Frequency (FFT). Use MATLAB's Machine Learning Toolbox to classify if the person is "Walking" or "Running.

8. Medical Imaging Enhancement and Edge Detection:

   Load a low-contrast MRI/X-ray image, apply histogram equalization, and contrast stretching techniques to enhance visibility of anatomical structures. Implement spatial filtering and edge detection operators such as Sobel, Prewitt, and Laplacian to highlight boundaries and abnormalities. Compare the effectiveness of each operator in terms of edge clarity and noise sensitivity.
9. Frequency Domain Image Filtering using FFT:

   Perform 2D FFT on a noisy grayscale image and visualize its frequency spectrum. Design and apply Low-Pass and High-Pass frequency-domain filters to remove noise and enhance edges. Reconstruct the processed image using inverse FFT and compare the results with spatial-domain filtering techniques in terms of image sharpness and computational efficiency.
10. Mini Project-: AI-DSP combined processing technique.

Text Books:

1. Digital Signal Processing: Principles, Algorithms and Applications by John G. Proakis & Dimitris G. Manolakis, Pearson Education, 2007.
2. Digital Image Processing by Rafael C. Gonzalez & Richard E. Woods, Pearson Education, 2018.

Reference Books:

1. Signals and Systems by Alan V. Oppenheim, Alan S. Willsky, & S. Hamid Nawab, Pearson Education, 1997.
2. Discrete-Time Signal Processing by Alan V. Oppenheim & Ronald W. Schafer, Pearson Education, 2010.
3. Digital Signal Processing by S. Salivahanan, A. Vallavaraj, & C. Gnanapriya, McGraw Hill Education, 2018.
4. Digital Image Processing Using MATLAB by Rafael C. Gonzalez, Richard E. Woods, & Steven L. Eddins, McGraw Hill Education, 2020.
5. Digital Signal Processing: A Computer-Based Approach by Sanjit K. Mitra, McGraw Hill Education, 2011.

Course Code	25EE01TH0405-1			
Category	Multidisciplinary Minor -2			
Course Title	MIPS Processor Design and Testing			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	IV

Course Outcomes:

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

1. Apply MIPS Instruction Set Architecture (ISA) principles to translate high-level C programs into assembly code and instruction formats.
2. Design MIPS hardware components (ALU, Register File, Control Unit, Memory) by formulating Verilog HDL models based on architectural specifications.
3. Analyze single-cycle datapath operations and compute path latencies to determine the maximum operating frequency.
4. Analyze and debug the functional integrity of processor modules the complete integrated system
5. Examine advanced architectural features including pipelining hazards, I/O subsystems, and domain-specific architectures to identify their role in modern computer architecture.

Syllabus:

Module I: Introduction to MIPS Processor and Design Flow(07 Hrs)
Introduction to computer system and its sub modules, Calculate Processor performance metrics, Introduction to RISC and CISC paradigm, overview of the MIPS Instruction Set Architecture (ISA), registers, instruction formats, and addressing modes, single-cycle, breakdown of the MIPS data path into instruction fetch, decode, execute, memory, and writeback stages. Code Conversion C to MIPS

Module II: Designing the MIPS ALU and Register File(07 Hrs)
Hardware modeling for Arithmetic and logical operations performed by the MIPS ALU, implementation of a register file, testbench creation for verifying the ALU and register file operations.

Module III: Instruction Fetch and Decode Stage Implementation(07 Hrs)
The instruction fetch stage, design of program counter (PC) and instruction memory components, instruction decode stage, generation of control signals, Integration of the instruction memory with the register file.

Module IV: Execution, Memory, and Writeback Stages(07 Hrs)
Implementation of the execution stage, ALU operation and branching logic, memory access

stage, design of control logic for load store instructions, implementation of write back stage, testing of R-type, I-type, and J-type instructions using data and control path.

Module V: Integration and Testing of Single-Cycle MIPS Processor, Multiprocessor Architectures (07 Hrs)

Assembling the complete single-cycle MIPS processor, develop a complete Verilog testbench for the entire processor, run test programs, and debug issues that arise during simulation, analyze timing diagrams and waveforms using simulation tools. Overview of Pipelining and Parallel processing concepts, multiprocessors and its characteristics, Input/Output Subsystem:- Interfaces and buses, I/O Operations, Designing I/O Systems, Overview of Domain-Specific Architectures

Text Books:

1. Computer Organization and Design Edition - The Hardware/Software Interface, David A. Patterson, John L. Hennessy, 5th Edition, 2014.

Reference Books

1. Computer Architecture and Organization; J. P. Hayes; Third Edition (Fifth Reprint), McGraw Hill, 2012.
2. Computer Architecture And Parallel Processing; Kai Hawang, Faye A. Briggs, McGraw Hill, 2012

Course Code	25EE01TH0405-2			
Category	Multidisciplinary Minor -2			
Course Title	Programming for Environment IOT			
Scheme& Credits	L	P	Credits	Semester
	3	0	3	IV

Course Outcomes:

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

1. Explain architecture and components of Environmental IoT systems.
2. Develop embedded programs for environmental sensor interfacing.
3. Build cloud-connected monitoring and alert systems.
4. Design smart IoT solutions for sustainability and resource management.

Syllabus

Module I: (6 Hrs)

Introduction to Environmental IoT:

Concept of IoT, Environmental monitoring systems, Smart cities and sustainability, IoT architecture, Sensors–Gateway–Cloud model, Applications in air, water, agriculture and waste management.

Module II: (6 Hrs)

Programming Fundamentals for IoT Devices:

Embedded C / Python basics, Variables, loops, functions, Conditional statements, Interrupts, Real-time programming concepts, IDE setup, Introduction to Arduino, ESP32 and Raspberry Pi programming.

Module III: (6 Hrs)

Environmental Sensors and Interfacing:

Temperature and humidity sensors, Gas sensors (CO_2, CO, NH_3), pH sensor, Turbidity sensor, Soil moisture sensor, Light and UV sensors, ADC/DAC interfacing, I2C, SPI, UART communication.

Module IV: (6 Hrs)

Wireless Communication and Networking:

Wi-Fi, Bluetooth Low Energy, Zigbee, LoRaWAN, NB-IoT basics, MQTT protocol, HTTP/REST APIs, Edge gateways, Power management for remote nodes.

Module V: (6 Hrs)

Cloud Programming and Data Analytics:

IoT cloud platforms, Data logging, Dashboards, Real-time alerts, Mobile notifications, Data visualization, Predictive environmental analytics.

Module VI: (6 Hrs)

Applications of Environmental IoT:

Air quality monitoring station, Smart irrigation system, Water quality monitoring, Smart waste management, Flood warning system, Forest fire detection, Energy monitoring, Green campus solutions.

Text Books:

1. Adrian McEwen, *Designing the Internet of Things*, Wiley.
2. Raj Kamal, *Internet of Things: Architecture and Design Principles*, McGraw Hill.
3. Arshdeep Bahga & Vijay Madisetti, *Internet of Things – A Hands-on Approach*, Universities Press.

Reference Books:

1. David Hanes et al., *IoT Fundamentals*, Cisco Press.
2. Olivier Hersent, *The Internet of Things*, Wiley.
3. Vijay Madisetti, *IoT for Smart Cities and Environment*, CRC Press.

Course Code	25EE01PR0407			
Category	Vocational and Skill Enhancement Course			
Course Title	Algorithmic Problem Solving -I			
Scheme & Credits	L	P	Credits	Semester
	0	2	1	IV

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

1. Apply fundamental algorithms, complexity concepts, and basic I/O techniques to solve computational problems.
2. Implement data structures such as arrays, strings, stacks, and queues for problem solving.
3. Analyze problems and apply appropriate searching, sorting, and recursion techniques.
4. Develop efficient programs to solve entry-level competitive coding problems with optimized logic.

Syllabus

Module	Topics	Platform Mapping
Module 1: Introduction to Problem Solving & Complexity	Basics of algorithms, flowcharts, pseudocode, introduction to online judges, time & space complexity (Big O, Ω, Θ), basic I/O optimization	HackerRank – "Solve Me First", "Time Conversion" LeetCode – "Two Sum"
Module 2: Arrays & Strings (Foundation)	1D & 2D arrays, basic string manipulation, prefix sum, basic sliding window	HackerEarth – Basic Array Problems LeetCode – "Valid Palindrome", "Best Time to Buy and Sell Stock"
Module 3: Recursion Basics	Recursion fundamentals, factorial, Fibonacci, recursion tree, simple problems	HackerRank – Recursion Challenges LeetCode – "Climbing Stairs"
Module 4: Searching & Sorting (Basic)	Linear search, binary search, bubble sort, selection sort, insertion sort	HackerRank – "Binary Search" LeetCode – "Search Insert Position"
Module 5: Basic Data Structures	Stack (basic operations), Queue, simple Linked List (insertion, deletion)	HackerRank – Stack & Queue Problems LeetCode – "Implement Stack using Queues"
Module 6: Problem Solving Techniques	Brute force approach, basic optimization, pattern-based problems	HackerRank – Easy Problem Solving
Module 7: Practice for Placements (Level 1)	Easy-level coding problems, debugging techniques, code efficiency basics	LeetCode – Easy Problems Set

Module-wise Coding Practice Framework for Problem Solving

Module	Topics	Problems to Solve (Outcome)	Platform Choice (Level-wise)	Platform + Problem Count + Difficulty
Module 1 – Intro to Problem Solving & Complexity	Basics of algorithms, pseudocode, flowcharts, time & space complexity	15–20 problems	Beginners: HackerRank Intermediate: HackerEarth Professional: LeetCode	HackerRank (10 Easy) HackerEarth (5 Easy) LeetCode (5 Easy) **Total: 20 (Easy)**
Module 2 – Arrays & Strings	1D/2D arrays, string manipulation, prefix sum, basic sliding window	40–50 problems	Beginners: HackerEarth Intermediate: HackerRank Professional: LeetCode	HackerEarth (15 Easy) HackerRank (15 Easy/Medium) LeetCode (20 Medium) **Total: 50 (Easy–Medium)**
Module 3 – Recursion Basics	Factorial, Fibonacci, recursion tree, simple recursion problems	20–25 problems	Beginners: HackerRank Intermediate: HackerEarth Professional: LeetCode	HackerRank (10 Easy) HackerEarth (5 Easy/Medium) LeetCode (10 Medium) **Total: 25 (Easy–Medium)**
Module 4 – Searching & Sorting	Linear search, binary search, basic sorting algorithms	25–30 problems	Beginners: HackerRank Intermediate: HackerEarth Professional: LeetCode	HackerRank (10 Easy) HackerEarth (10 Easy/Medium) LeetCode (10 Medium) **Total: 30 (Easy–Medium)**
Module 5 – Basic Data Structures	Stack, Queue, Linked List basics	25–30 problems	Beginners: HackerRank Intermediate: HackerEarth Professional: LeetCode	HackerRank (10 Easy) HackerEarth (10 Medium) LeetCode (10 Medium) **Total: 30 (Easy–Medium)**
Module 6 – Problem Solving Techniques	Brute force, pattern-based problems, basic optimization	20–25 problems	Beginners: HackerRank Intermediate: HackerEarth Professional: LeetCode	HackerRank (10 Easy) HackerEarth (5 Medium) LeetCode (10 Medium) **Total: 25 (Easy–Medium)**
Module 7 – Placement Practice (Level 1)	Mixed problems (all topics)	30–40 problems	Intermediate: HackerRank Professional: LeetCode	HackerRank (15 Medium) LeetCode (20 Medium) **Total: 35 (Medium)**

Course Code	25HS04PR0401			
Category	Ability Enhancement Course			
Course Title	Self Defence and Indian Marital Art			
Scheme & Credits	L	P	Credits	Semester
	0	2	0	IV

Course Type: Mandatory Audit Course (Practical)

Semester: All Ug Program during III and IV Semesters

Class Duration: 14 weeks (2 hours/week)

Mode of Instruction: In-person, practical, Practice and interactive sessions

Assessment: Satisfactory/Unsatisfactory based on participation, skill demonstration, and adherence to safety protocols.

Course Objective

This course aims to equip students—particularly female students—with the foundational skills and knowledge of essential self-defence techniques and Bhartiya Yudhkala (Indian Martial Arts). By the end of the program, students will possess increased awareness, confidence, and practical ability to protect themselves in potentially dangerous situations.

Course Description

"**Self-Défense & Indian Martial Arts**" is a life skills training program aimed at enhancing personal safety, especially for young women. It combines traditional Indian martial arts with modern self-defence techniques, focusing on physical fitness, mental resilience, and situational awareness.

Key Features:

- Real-life scenario-based training and escape techniques
- Use of everyday objects for defence
- Cultural insights into Indian martial arts
- Safe, structured learning environment

Course Outcomes

1. Demonstrate improved physical fitness, including enhanced strength, endurance, and agility.
2. Understand key self-defence laws and individual rights under Indian legal frameworks.
3. Apply basic defence techniques from Indian martial arts, including escape strategies and improvised weapon use.
4. Follow safety protocols and utilize resources such as protective gear and safe environments to manage emergencies and prevent injuries.

Weekly Schedule & Modules

Week	Module	Content
Week 1	Module 1 – Introduction	Overview of self-defense, importance, mindset training, and history of Indian martial arts.
Week 2	Module 1 – Introduction (contd.)	Basics of situational awareness and psychological preparedness.
Week 3	Module 2 – Targeting Weak Points	Anatomical knowledge for self-defense: eyes, nose, throat, groin, knees.
Week 4	Module 2 – Targeting Weak Points (contd.)	Practice and roleplay scenarios.
Week 5	Module 3 – Ground Défense	Techniques to defend and escape when on the ground.
Week 6	Module 3 – Ground Défense (contd.)	Partner practice and resistance drills.
Week 7	Module 4 – 20 Escape Techniques Through Indian Culture	Practical escape methods inspired by traditional Indian martial forms.
Week 8	Module 4 – 20 Escape Techniques (contd.)	Application in staged confrontation settings.
Week 9	Module 5 – Weapon of Opportunity	Training with improvised weapons: pen, keys, dupatta, bags, etc.
Week 10	Module 5 – Weapon of Opportunity (contd.)	Timing, precision, and legal considerations.
Week 11	Module 6 – Stick Rotation	Basics of stick (lathi/danda) handling and defence.
Week 12	Module 6 – Stick Rotation (contd.)	Practice with focus on flow, blocking, and disarming.

Note:

1.Students of Higher Semester classes have to Complete this 30 hrs Audit Course during any one of the (III/IV) semesters. Preferably Similar distribution of Students as Sports Classes (Now Newly Named: Health-Fitness and Wellbeing) in First Year of UG program.
2.A minimum of 60% attendance and satisfactory performance is necessary for passing the courses.
3.The students who are Suffering from Serious Disease, injured, specially abled and medically Critical condition shall be exempted from the exam.
4.Requirement: A multipurposed Indoor Ac Hall with capacity of Minimum 200 Person for conducting Various, Self-Défense classes and Yoga & Meditation**. If there is enough space to conduct at least 03 Sections (200 students) of students simultaneously then it is feasible to start the course.**

1. Requirement of sufficient number of Expert Trainers of self-defense Course

Course Code	25EE01MT0401				
Category	Minor Course				
Course Title	Sensor Interfacing with Arduino and ESP8266				
Scheme & Credits	L	T	P	Credits	Semester
	3	0	0	3	IV

Course Outcomes:

After learning the course, the student will be able to:

1. Know about the open source hardware platforms like Arduio and ESP
2. Learn the working of sensors and actuators
3. Understand the working of serial communication protocols

Syllabus

Module I: (6Hrs)

Introduction to Arduino Prototyping Platform: Arduino IDE, Arduino C, Setting up the Arduino board, creating sketches,using Libraries, using example codes, Debugging Using the Serial Monitor.

Module II: (7Hrs)

Sensor interfacing with Arduino: Analog and digital sensors, Temperature sensors, Humidity sensors, Obstacle sensors, Ultrasonic sensor, Accelerometer and gyro, etc.

Module III: (7Hrs)

Serial Communication in Arduino: Serial and parallel communication, Serial communication protocols, UART, I2C, SPI, Wired and Wireless communication, Interfacing Communication Modules with Arduino.

Module IV: (7 Hrs)

Interfacing Displays and Actuators: 16x2 LCD, Graphical LCD, Graphical OLED, Arduino Tone functions, Melody generation on a piezo buzzer, Speed and direction control of DC, Servo and, Stepper Motor.

Module V: (6Hrs)

Introduction to ESP8266: ESP8266 development board, Programming ESP8266 through Arduino IDE, connecting to the internet, sending and receiving data on internet.

Module VI: (7Hrs)

Interfacing sensors and actuators with ESP8266: LDR, Temperature sensor, Humidity sensor, IR sensor, OLED, RGB LED, Servo motor, etc.

Text Books:

1. Arduino Cookbook by Michael Margolis, O'Reilly Media, Inc., 1st edition
2. Beginning C for Arduino by Jack Purdum (ebook)
3. Arduino for Beginners: Essential Skills Every Maker Needs, John Baichtal, Pearson Education, Inc., 1st edition

Course Code	24EE01TH0505-1			
Category	Multidisciplinary Minor -3			
Course Title	Chip Verification Using System Verilog			
Scheme & Credits	L	P	Credits	Semester
	3	0	3	V

Course Outcomes:

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

1. **CO1 (Understand/Apply):** Develop comprehensive Verification Plans and Self-Checking Testbenches for combinational and sequential modules using reference models and scoreboards.
2. **CO2 (Evaluate):** Design and implement Constrained Random Stimulus and Object-Oriented transactions to automate the exploration of complex hardware state spaces.
3. **CO3 (Analyze/Design):** Construct SystemVerilog Assertions (SVA) to monitor and validate temporal logic, timing requirements, and protocol compliance in real-time.
4. **CO4 (Analyze/Evaluate):** Quantify verification quality using Functional and Code Coverage metrics to determine "Verification Sign-off" and tape-out readiness.
5. **CO5 (Create/Modernize):** Architect a complete System-Level Environment for processors (MIPS) and protocols, integrating AI and Machine Learning for intelligent test generation and coverage closure.

Syllabus:

Module I: Verification Planning and SV Essentials

Verification vs. Validation, The Verification Lifecycle, Verification Planning, Defining Features vs. Test Cases, SystemVerilog Data Types, Logic vs. Bit, Interfaces, Modports, Program Blocks vs. Modules, Avoiding Race Conditions, Tasks and Functions for Verification.

Module II: Verifying Combinational Logic & Data Integrity

Directed vs. Random Testing, Reference (Golden) Models, Self-checking Testbench Architecture, Scoreboard Logic, Verifying 32-bit ALUs, Priority Encoders, Barrel Shifters, Data Comparison logic, Predictor Functions, Error Injection techniques.

Module III: Verifying Sequential Logic & Assertions

Timing-accurate Stimulus, Clocking Blocks, Input/Output Skews, Synchronous and Asynchronous Reset Verification, FIFO Depth/Overflow/Underflow Testing, FSM State Transition Verification, Introduction to SystemVerilog Assertions (SVA), Immediate vs. Concurrent Assertions, Temporal Logic for Timing Checks.

Module IV: Transaction-Level Modeling & Randomization

Object-Oriented Programming (OOP) for Verification, Transactions and Data Packets, Class-based Testbench Components, Dynamic Arrays and Queues, Constrained Randomization, rand vs. randc, Weighted Constraints, Conditional Constraints (if-else, implication), Handling Corner Cases.

Module V: Protocol Verification & Coverage Sign-off
Verification of Bus Protocols (UART, SPI, AMBA-APB), Master and Slave Bus Functional Models (BFMs), Transaction-level checking, Functional Coverage, Covergroups, Coverpoints, Cross-coverage, Binning, Code Coverage (Statement, Branch, Toggle), Coverage-driven Verification (CDV), Defining Verification Sign-off Criteria, Determining "When is Verification Done?"

Module VI: AI-Augmented System Verification (MIPS Case Study)
Layered Testbench Architecture for MIPS, Verifying Pipeline Hazards and Memory Coherency, AI-driven Stimulus Generation, Using LLMs for Assertion (SVA) Synthesis, Machine Learning for Coverage Closure (Intelligent steering), AI-based Bug Localization and Failure Triage, Pattern Recognition in Regression results, Introduction to Autonomous Verification Agents, The Future of AI in Chip Validation.

Textbook:

1. SystemVerilog for Verification: A Guide to Learning the Testbench Language; Chris Spear & Greg Tumbush, 3rd edition, Springer Nature, 2012
2. Writing Testbenches using SystemVerilog; Anick Bergeron Springer Nature, 2006

Reference Books:

1. SystemVerilog Assertions Handbook, Ben Cohen, VDHLCohen Publishing, 2023

Course Code	24EE01TH0505-2				
Category	Multidisciplinary Minor -3				
Course Title	IoT Privacy and Security				
Scheme & Credits	L	T	P	Credits	Semester
	3	0	0	3	V

Course Outcomes

1. Understand the principles of security C.I.A
2. Apply the principals to identify the vulnerabilities at different levels
3. Analyse the security required at different layers in IoT models
4. Investigate the attacks using detection tools
5. Evaluate the attacks, patterns and install the security provisons

Module 1: Principles of security: C.I.A, in Data Network and IoT network
Module 2: security policies Encryption and PGP
Module 3: Vulnerability at various levels in IoT, Security at different levels in IoT: perception, Network,
Transport, Application
Module 4: Secure Elements in IoT, Hardware Authenticators, Trusted Platform Modules(TPM),
Transport Layer Shell etc
Module 5: Use cases of deployment different security scenarios Hardware Authenticators, Trusted
Platform Modules(TPM), SSH etc.

Text Book:

1. Internet of Things Principles and Paradigms, Rajkumar Buyya Amir Vahid Dastjerdi, Morgan
Kaufman, Elsevier 2016 1 st Edition

Reference Books:

1. Internet of Things Principles, Paradigms and Application of IoT, Joseph Kofi Wireko, Kaml
Hiran, BPB Publications 2020 1 st Edition
2. IoT Protocols and IoT Standards by Jari Haiston 2023, Symmetry Electronics
3. Cisco Application Note on Network Security
4. Microchip Application note on Trusted Platform Module(TPM)

Course Code	24EE01TH0605-1				
Category	Multidisciplinary Minor -4				
Course Title	VLSI Physical Design				
Scheme & Credits	L	T	P	Credits	Semester
	3	0	0	3	VI

Course Outcomes:

On successful completion of the course, students will be able to:

1. Interpret the VLSI design flow and various VLSI design styles in detail
2. Apply algorithmic graph theory and combinatorial optimization techniques, as per requirement, to correctly formulate and solve VLSI design problems
3. Apply the algorithms for partitioning, floor planning, placement and routing of VLSI circuits and use them to solve simple VLSI design problems.
4. Analyze Static Timing Analysis of VLSI circuits.

Syllabus:

Module 1: (06 Hrs)

Introduction to VLSI CAD : VLSI design methodologies, use of VLSI CAD tools, Algorithmic Graph Theory and computational Complexity.

Module 2: (08 Hrs)

High-level Synthesis : Hardware Models for High-level Synthesis, Internal Representation of the Input Algorithm, and Understanding RTL to Gate Level design mapping. Basic concept of Static Timing Analysis (STA).

Module 3: (08 Hrs)

Partitioning: Introduction, Types of Partitioning, Classification of partitioning Algorithm, KL algorithm.

Floor - planning: Introduction, Sliced and non-sliced planning, Polish expression, Power planning, IO Planning.

Placement: Introduction, classification of placement algorithms, partition based placement, timing / congestion aware Placement.

Module 4: (08 Hrs)

Clock Tree Synthesis: Different topologies of Clock Tree Structure. Overview on Clock Mesh implementation for High Performance designs.

Routing : Fundamental Concepts of Steiner trees, Two phases of Routing: Global routing & detailed routing, Routing Algorithms.

Module 5: (08 Hrs)

Low Power Physical Design : Understanding Various Power Optimization algorithms (dynamic and Leakage). Overview on implementation and complexities involved in low power PD.

SOC Physical Design : Re-convergent model of VLSI SOC Design, SOC Physical design, advanced physical design of SOCs

Text Books:

1. VLSI Physical Design Automation: Theory and Practice: Sadiq M. Sait, Habib Youssef, McGraw- Hill 2004

2. VLSI Physical Design: From Graph Partitioning to Timing Closure: Jin Hu, Jens Lienig, Igor L. Markov, Andrew B. Kahng, Springer, Dordrecht 2011.

Reference Books:

1. Physical Design Essentials: An ASIC Design Implementation Perspective: Khosrow Golshan, Springer, (2007).

2. Static Timing Analysis for Nanometer Designs: A Practical Approach: J. Bhasker and Rakesh Chadha, Springer, (2009).

3. Practical Problems in VLSI Physical Design Automation, Sung Kyu Lim, Springer, (2008), ISBN 978-1402066269.

4. Algorithms for VLSI Design Automation: Sabih H. Gerez and John Wiley,(1998).

5. An Introduction to VLSI Physical Design: Majid Sarrafzadeh and C. K. Wong, McGraw Hill, (1996).

6. Algorithms for VLSI Physical Design Automation: Naveed Sherwani, Kluwer Academic Pub., (1999).

Course Code	24EE01TH0605-2				
Category	Multidisciplinary Minor -4				
Course Title	Use cases of Environmental IoT				
Scheme & Credits	L	T	P	Credits	Semester
	3	0	0	3	VI

Course Outcomes:

At the end of this course, students will be able to:

1) Select and integrate appropriate sensors and LPWAN protocols to monitor diverse environmental parameters such as air quality, water health, and soil conditions in remote or rugged settings.
2) Architect IoT-based solutions for water distribution, precision agriculture, and waste management that optimize resource consumption and improve operational efficiency.
3) Apply IoT data analytics to build real-time monitoring systems for environmental threats, including forest fires, industrial gas leaks, and urban pollution spikes.
4) Assess the sustainability of IoT deployments by implementing energy-harvesting methods (solar, thermal) and edge computing to minimize the carbon footprint of the network itself.
5) Interpret environmental data into actionable insights for Air Quality Indexing (AQI) and regulatory compliance, bridging the gap between raw sensor data and environmental governance.

Syllabus

Module 1: Environmental Sensing & Data Acquisition

Review of sensor nodes for environmental parameters (PM2.5/PM10, CO2, VOCs, dissolved oxygen, soil moisture, noise levels); Low-power wide-area networks (LPWAN) for remote monitoring (LoRaWAN, NB-IoT).

Module 2: Smart Water Management

Real-time water quality monitoring in rivers and lakes; IoT for wastewater treatment plants; Smart irrigation systems using soil moisture and weather forecasting; Detecting leakages in urban water distribution.

Module 3: Air Quality & Climate Monitoring

Urban air quality indexing (AQI) networks; Monitoring greenhouse gas emissions in industrial zones; IoT for forest fire detection and early warning systems; Wildlife tracking and biodiversity monitoring.

Module 4: Smart Agriculture & Waste Management

Precision farming (site-specific crop management); Livestock health monitoring; IoT-enabled smart bins and optimized waste collection routing; Circular economy and e-waste tracking.

Module 5: Edge Intelligence & Sustainability

Edge computing for reduced data latency in disaster management; Energy harvesting for self-powered IoT nodes (solar, thermal, vibration); Environmental impact of IoT (Green IoT)

Text Books:

1. Internet of Things for Sustainable Community Development-Wireless Communications, Sensing, and Systems by Abdul Salam, Springer Nature, 1st Edition, 2020
2. IoT and Low Power Wireless: Circuits, Systems, and Applications by Christopher Siu. CRC Press, 1st Edition, 2018.

Reference Books:

1. "Precision Agriculture Technologies for Food Security and Sustainability" by Abd El-Kader and El-Basioni.
2. "Environmental Management using the Internet of Things" (Springer Briefs) by various authors.
3. ITU-T Reports on "L.1300 series": Specifically focusing on Green Data Centres and Environmental Sustainability.
4. Case Studies from The LoRa Alliance: Real-world whitepapers on smart water and soil monitoring.