VARDHAMAN
COLLEGE OF ENGINEERING

CURRICULUM
For
Bachelor of Technology
Computer Science and Engineering

Under
Choice Based Credit System (CBCS)

B. Tech. - Regular Four-Year Degree Program
(For batches admitted from the Academic Year 2025 - 2026)

&

B. Tech. - Lateral Entry Scheme
(For batches admitted from the Academic Year 2026 - 2027)

August 2025

VARDHAMAN COLLEGE OF ENGINEERING
(Autonomous)
Affiliated to **JNTUH,** Approved by **AICTE,** Accredited by **NAAC** with **A++** Grade
Kacharam, Shamshabad, Hyderabad- 501 218, Telangana, India
www.vardhaman.org, info@vardhaman.org

Department Vision

To be a leading source of competent computer engineers and meeting the needs of industry and society at large.

Department Mission

M1: Facilitate learning in advanced technologies adopting innovative methods.

M2: Associate continuously with industry to design and implement experiential curriculum.

M3: Promote Research and Development through Special Interest Groups (SIGs).

M4: Provide a platform for harnessing entrepreneurial and leadership qualities.

Program Educational Objectives (PEOs)

PEO1: Graduate will establish himself/herself as effective professionals by solving real-world problems using investigative and analytical skills along with the knowledge acquired in the field of Computer Science and Engineering.

PEO2: Graduate will demonstrate his/her ability to adapt to a rapidly changing environment in advanced areas of Computer Science and scale new heights in their profession through lifelong learning.

PEO3: Graduate will prove his/her ability to work and communicate effectively as a team member and /or leader to complete the task with minimal resources, meeting deadlines.

PEO4: Graduate will embrace the professional code of ethics in the profession while deliberately being part of projects, which contributes to the society at large, without disturbing the ecological balance.

Knowledge and Attitude Profile (WK)

WK1: A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.

WK2: Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.

WK3: A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.

WK4: Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline; much is at the forefront of the discipline.

WK5: Knowledge, including efficient resource use, environmental impacts, whole-life cost,reuse of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area.

WK6: Knowledge of engineering practice (technology) in the practice areas in the engineering discipline.

WK7: Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.

WK8: Engagement with selected knowledge in the current research literature of the discipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.

WK9: Ethics, inclusive behavior and conduct. Knowledge of professional ethics, responsibilities, and norms of engineering practice. Awareness of the need for diversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

Program Outcomes (POs)

Engineering Graduates will be able to:

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4).

PO3: Design/ Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5).

PO4: Conduct investigations of complex problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Modern Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6).

PO6: The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Team work: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9: Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences.

PO10: Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to one's own work, as a member and leader in a team, and to manage projects and in multidisciplinary environments.

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8)

Program Specific Outcomes (PSOs)

PSO1: To collect requirements, analyze, design, implement and test software systems.

PSO2: To analyze the errors and debug them accordingly.

United Nations Sustainable Development Goals (SDGs)

SDG1: No Poverty – End poverty in all its forms everywhere.

SDG2: Zero Hunger – End hunger, achieve food security and improved nutrition and promote sustainable agriculture.

SDG3: Good Health and Well-Being – Ensure healthy lives and promote well-being for all at all ages.

SDG4: Quality Education – Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.

SDG5: Gender Equality – Achieve gender equality and empower all women and girls.

SDG6: Clean Water and Sanitation – Ensure availability and sustainable management of water and sanitation for all.

SDG7: Affordable and Clean Energy – Ensure access to affordable, reliable, sustainable and modern energy for all.

SDG8: Decent Work and Economic Growth – Promote sustained, inclusive and sustainable economic growth, full and productive employment and decent work for all.

SDG9: Industry, Innovation and Infrastructure – Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation.

SDG10: Reduced Inequalities – Reduce inequality within and among countries.

SDG11: Sustainable Cities and Communities – Make cities and human settlements inclusive, safe, resilient and sustainable.

SDG12: Responsible Consumption and Production – Ensure sustainable consumption and production patterns.

SDG13: Climate Action – Take urgent action to combat climate change and its impacts.

SDG14: Life Below Water – Conserve and sustainably use the oceans, seas and marine resources for sustainable development.

SDG15: Life on Land – Protect, restore and promote sustainable use of terrestrial ecosystems, sustainably manage forests, combat desertification, and halt and reverse land degradation and halt biodiversity loss.

SDG16: Peace, Justice and Strong Institutions – Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels.

SDG17: Partnerships for the Goals – Strengthen the means of implementation and revitalize the global partnership for sustainable development.

I B.Tech. I Semester												
#	Course Code	Title of the Course	Category	Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
				CI		LI	TW + SL	H	C	CIE	SEE	Total
				L	T	P	SL					
Theory Courses												
1	A9001	Matrices and Calculus	BS	45	15	-	60	120	4	40	60	100
2	A9009	Engineering Chemistry	BS	45	-	-	45	90	3	40	60	100
3	A9501	Programming for Problem Solving	ES	45	-	-	45	90	3	40	60	100
4	A9011	English for Skill Enhancement	HS	30	-	-	30	60	2	40	60	100
5	A9402	Digital Electronics	ES	45	-	-	45	90	3	40	60	100
Practical Courses												
6	A9010	Engineering Chemistry Laboratory	BS	-	-	30	-	30	1	40	60	100
7	A9502	Programming for Problem Solving Laboratory	ES	-	-	30	-	30	1	40	60	100
8	A9012	English Language and Communiaction Skills Laboratory	HS	-	-	30	-	30	1	40	60	100
9	A9304	Computer Aided Engineering Graphics	ES	-	-	30	-	30	1	40	60	100
Community Related Project Work												
10	A9021	Community Centered Design Thinking	PW	-	-	-	45	45	1	40	60	100
Total				210	15	120	270	615	20	400	600	1000

I B.Tech. II Semester												
#	Course Code	Title of the Course	Category	Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
				CI		LI	TW + SL	H	C	CIE	SEE	Total
				L	T	P	SL					
Theory Courses												
1	A9002	Ordinary Differential Equations and Vector Caluculus	BS	45	15	-	60	120	4	40	60	100
2	A9007	Engineering Physics	BS	45	-	-	45	90	3	40	60	100
3	A9204	Basic Electrical Engineering	ES	30	-	-	30	60	2	40	60	100
4	A9503	Data Structures	ES	45	-	-	45	90	3	40	60	100
5	A9505	Computer Organization	ES	45	-	-	45	90	3	40	60	100
Practical Courses												
6	A9008	Engineering Physics Laboratory	BS	-	-	30	-	30	1	40	60	100
7	A9205	Basic Electrical Engineering Laboratory	ES	-	-	30	-	30	1	40	60	100
8	A9504	Data Structures Laboratory	ES	-	-	30	-	30	1	40	60	100
9	A9302	Engineering Workshop	ES	-	-	30	-	30	1	40	60	100
Community Related Project Work												
10	A9022	Product Design and Develpment	PW	-	-	-	45	45	1	40	60	100
Total				210	15	120	270	615	20	400	600	1000

II B.Tech. I Semester													
#	Course Code	Title of the Course	Category	Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
				CI		LI	TW + SL	H	C	CIE	SEE	Total
				L	T	P	SL					
Theory Courses												
1	A9014	Buisness Economics and Financial Analysis	HS	45	-	-	45	90	3	40	60	100
2	A9506	Discrete Mathemetical Structures	ES	45	-	-	45	90	3	40	60	100
3	A9507	Operating Systems	PC	45	-	-	45	90	3	40	60	100
4	A9509	Data Base Management System	PC	45	-	-	45	90	3	40	60	100
5	A9601	Object Oriented Programming through Java	PC	45	-	-	45	90	3	40	60	100
Practical Courses												
6	A9508	Operating Systems Laboratory	PC	-	-	30	-	30	1	40	60	100
7	A9510	Data Base Management System Laboratory	PC	-	-	30	-	30	1	40	60	100
8	A9602	Object Oriented Programming through Java Laboratory	PC	-	-	30	-	30	1	40	60	100
9	A9511	Python Programming Laboratory	PC	-	-	30	-	30	1	40	60	100
Skill Development Course												
10	A9802	Data Analysis and Visualization using Power BI	PC	-	-	30	-	30	1	40	60	100
Community Related Project Work												
11	A9023	Technology Entrepreneurship	PW	-	-	-	45	45	1	40	60	100
Total				225	0	150	270	645	21	440	660	1100

II B.Tech. II Semester													
#	Course Code	Title of the Course	Category	Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
				CI		LI	TW + SL	H	C	CIE	SEE	Total
				L	T	P	SL					
Theory Courses												
1	A9005	Probability Distributions and Applied Statistics	BS	45	-	-	45	90	3	40	60	100
2	A9513	Computer Networks	PC	45	-	-	45	90	3	40	60	100
3	A9604	Web Application Engineering	PC	45	-	-	45	90	3	40	60	100
4	A9515	Software Engineering	PC	45	-	-	45	90	3	40	60	100
5	A9516	Design and Analysis of Algorithms	PC	45	-	-	45	90	3	40	60	100
Practical Courses												
6	A9514	Computer Networks Laboratory	PC	-	-	30	-	30	1	40	60	100
7	A9605	Web Application Engineering Laboratory	PC	-	-	30	-	30	1	40	60	100
8	A9606	IoT Laboratory	PC	-	-	30	-	30	1	40	60	100
9	A9006	Computational Mathematics Laboratory	BS	-	-	30	-	30	1	40	60	100
Skill Development Course												
10	A9517	User Experience Design	PC	-	-	30	-	30	1	40	60	100
Community Related Project Work												
11	A9024	Community Driven Product Evaluation	PW	-	-	-	45	45	1	40	60	100
Total				225	0	150	270	645	21	440	660	1100
12		Exit Optional: Work Based Vocational Course / Internship or Apprenticeship	PW	-	-	-	90	90	2	40	60	100

III B.Tech. I Semester													
#	Course Code	Title of the Course	Category	Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
				CI		LI	TW + SL	H	C	CIE	SEE	Total
				L	T	P	SL					
Theory Courses												
1	A9702	Machine Learning	PC	45	-	-	45	90	3	40	60	100
2	A9518	Cloud Computing and Virtualization	PC	45	-	-	45	90	3	40	60	100
3	A9519	Automata Theory and Compiler Design	PC	45	-	-	45	90	3	40	60	100
4		Professional Elective - I	PE	45	-	-	45	90	3	40	60	100
5		Open Elective - I	OE	45	-	-	45	90	3	40	60	100
Practical Courses												
6	A9703	Machine Learning Laboratory	PC	-	-	30	-	30	1	40	60	100
7	A9013	English for Employability Skills Laboratory	HS	-	-	30	-	30	1	40	60	100
Skill Development Course												
8	A9520	Cloud Services using AWS and Azure	PC	-	-	30	-	30	1	40	60	100
Experiential Learning Course												
9	A9041	Internship/Industrial Training	PW	-	-	-	90	90	2	40	60	100
Value Added Course												
10	A9016	Gender Sensitization, Human Values and Professional Ethics	VA	15	-	-	15	30	1	40	60	100
Total				240	0	90	330	660	21	400	600	1000

III B.Tech. II Semester													
#	Course Code	Title of the Course	Category	Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
				CI		LI	TW + SL	H	C	CIE	SEE	Total
				L	T	P	SL					
Theory Courses												
1	A9610	DevOps	PC	45	-	-	45	90	3	40	60	100
2	A9705	Deep Learning	PC	45	-	-	45	90	3	40	60	100
3	A9612	Information Security	PC	45	-	-	45	90	3	40	60	100
4		Professional Elective - II	PE	45	-	-	45	90	3	40	60	100
5		Open Elective - II	OE	45	-	-	45	90	3	40	60	100
Practical Courses												
6	A9611	DevOps Laboratory	PC	-	-	30	-	30	1	40	60	100
7	A9706	Deep Learning Laboratory	PC	-	-	30	-	30	1	40	60	100
Skill Development Course												
8	A9613	Android App Development	PC	-	-	30	-	30	1	40	60	100
Experiential Learning Course												
9	A9042	Mini Project	PW	-	-	-	90	90	2	40	60	100
Value Added Course												
10	A9015	Environmental Science	VA	15	-	-	15	30	1	40	60	100
Total				240	0	90	330	660	21	400	600	1000

IV B.Tech. I Semester												
#	Course Code	Title of the Course	Category	Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
				CI		LI	TW + SL	H	C	CIE	SEE	Total
				L	T	P	SL					
Theory Courses												
1	A9814	Big Data Analytics	PC	45	-	-	45	90	3	40	60	100
2	A9707	Natural Language Processing	PC	45	-	-	45	90	3	40	60	100
3		Professional Elective - III	PE	45	-	-	45	90	3	40	60	100
4		Professional Elective - IV	PE	45	-	-	45	90	3	40	60	100
5		Open Elective - III	OE	45	-	-	45	90	3	40	60	100
Practical Courses												
6	A9815	Big Data Analytics Laboratory	PC	-	-	30	-	30	1	40	60	100
7	A9708	Natural Language Processing Laboratory	PC	-	-	30	-	30	1	40	60	100
Experiential Learning Course												
8	A9043	Major Project – Phase I	PW	-	-	-	90	90	2	100	-	100
Value Added Course												
9	A9017	Indian Knowledge System	VA	15	-	-	15	30	1	40	60	100
Total				240	0	60	330	630	20	420	480	900

IV B.Tech. II Semester												
#	Course Code	Title of the Course	Category	Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
				CI		LI	TW + SL	H	C	CIE	SEE	Total
				L	T	P	SL					
Theory Courses												
1		Professional Elective – V	PE	45	-	-	45	90	3	40	60	100
2		Professional Elective – VI	PE	45	-	-	45	90	3	40	60	100
Experiential Learning Course												
3	A9044	Major Project – Phase II	PW	-	-	-	630	630	14	40	60	100
Total				90	0	0	720	810	20	120	180	300

Common Abbreviations Used in the Curriculum

BS – Basic Sciences
HS – Humanities & Social Sciences
ES – Engineering Sciences
PC – Professional Core
PE – Professional Elective
OE – Open Elective
PW – Project Work
VA – Value Added Course
CI – Classroom Instruction
LI – Laboratory Instruction
L – Lecture Hours
T – Tutorial Hours
P – Practical Hours
TW – Team Work
SL – Self Learning
H – Hours
C – Credits
CIE – Continuous Internal Evaluation
SEE – Semester End Examination
SDG – Sustainable Development Goals

List of Professional Electives

Domain: Data Science			
III B.Tech. I Semester	Professional Elective – I	A9851	Data Science for Engineers
III B.Tech. II Semester	Professional Elective – II	A9852	Data Handling and Visualization
IV B.Tech. I Semester	Professional Elective – III	A9853	Data Analytics
IV B.Tech. I Semester	Professional Elective – IV	A9854	Social Media Mining
IV B.Tech. II Semester	Professional Elective – V	A9855	Cyber Forensics
IV B.Tech. II Semester	Professional Elective – VI	A9856	Optimization Techniques for Analytics

Domain: Network Security			
Year & Semester	**Professional Elective #**	**Course Code**	**Title of the Course**
III B.Tech. I Semester	Professional Elective – I	A9651	Ethical Cyber Security
III B.Tech. II Semester	Professional Elective – II	A9652	Web and Database Security
IV B.Tech. I Semester	Professional Elective – III	A9653	Cloud Security
IV B.Tech. I Semester	Professional Elective – IV	A9654	Wireless Mobile Security
IV B.Tech. II Semester	Professional Elective – V	A9655	IoT Security
IV B.Tech. II Semester	Professional Elective – VI	A9656	Blockchain Technology

Domain: Emerging Technologies			
Year & Semester	**Professional Elective #**	**Course Code**	**Title of the Course**
III B.Tech. I Semester	Professional Elective – I	A9551	Agile Methodology
III B.Tech. II Semester	Professional Elective – II	A9552	Augmented Reality and Virtual Reality
IV B.Tech. I Semester	Professional Elective – III	A9553	Robotic Process Automation
IV B.Tech. I Semester	Professional Elective – IV	A9554	Augmented Data Analytics
IV B.Tech. II Semester	Professional Elective – V	A9555	Quantum Computing
IV B.Tech. II Semester	Professional Elective – VI	A9556	Edge Computing

Domain: Emerging Trends in Machine Learning			
Year & Semester	**Professional Elective #**	**Course Code**	**Title of the Course**
III B.Tech. I Semester	Professional Elective – I	A9751	Explainable AI
III B.Tech. II Semester	Professional Elective – II	A9752	Federated Machine Learning
IV B.Tech. I Semester	Professional Elective – III	A9753	Reinforcement Learning
IV B.Tech. I Semester	Professional Elective – IV	A9754	Cognitive Computing
IV B.Tech. II Semester	Professional Elective – V	A9755	Agentic AI
IV B.Tech. II Semester	Professional Elective – VI	A9756	Quantum Computing for Machine Learning

List of Open Electives

Industry Skills		
#	Course Code	Title of the Course
1	A9505	Computer Organization
2	A9507	Operating Systems
3	A9509	Database Management Systems
4	A9515	Software Engineering
5	A9604	Web Application Engineering
6	A9612	Information Security
7	A9681	Cyber Security
8	A9682	Java Programming
9	A9683	Prompt Engineering
10	A9701	Artificial Intelligence
11	A9702	Machine Learning
12	A9705	Deep Learning
13	A9707	Natural Language Processing
14	A9710	Generative AI
15	A9803	Data Mining
16	A9851	Data Science for Engineers
Emerging Technologies		
#	Course Code	Title of the Course
17	A9381	Fundamentals of Robotics
18	A9382	Introduction to 3D Printing
19	A9383	Hybrid Vehicles
20	A9481	Internet of Things (IoT)
21	A9482	Consumer Electronics
22	A9483	VLSI Design Fundamentals
23	A9484	PCB Design and Fabrication
24	A9656	Blockchain Technology
Sustainability		
#	Course Code	Title of the Course
25	A9166	Smart Cities
26	A9181	Disaster Management
27	A9182	Road Safety Engineering
28	A9183	Building Science and Technology
29	A9281	Renewable Energy Systems
30	A9282	Smart Grid Technologies
31	A9283	Electrical Safety and Sustainable Engineering Practices
32	A9284	Smart Power Systems for Data Centers
33	A9285	E-Waste Management

List of Open Electives (Continued...)

Entrepreneurship		
#	Course Code	Title of the Course
34	A9081	Entrepreneurship Development
35	A9082	Research Methodology and IPR
36	A9083	Principles of Management
37	A9084	Organizational Behavior
38	A9355	Operations Research
39	A9684	E-Commerce
Life Skills and Holistic Development		
#	Course Code	Title of the Course
40	A9085	Emotional Intelligence and Leadership
41	A9086	Yoga and Wellness
42	A9087	National Cadet Corps (NCC)

I B.Tech. I Semester

A9001 – Matrices and Calculus

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	15	0	60	120	4	40	60	100

Course Description

Course Overview

This course provides mathematical knowledge required to analyze problems encountered in engineering. In this course, the students are acquainted with the solution of system of linear equations, eigen values and eigen vectors, functions of several variables, multiple integrals. The course is designed to build conceptual clarity and problem-solving skills, with emphasis on both theoretical understanding and practical applications.

Course Pre/Co-requisites

This course has no specific prerequisite and co-requisite.

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

Course Outcomes

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

A9001.1. Solve system of equations using rank of a matrix.

A9001.2. Construct the canonical form of a quadratic form using orthogonal transformations.

A9001.3. Express a function in series by mean value theorems and evaluate improper integrals using Beta and Gamma functions.

A9001.4. Examine the extremum of a function of several variables.

A9001.5. Apply multiple integrals to find the areas and volumes.

Course Syllabus

Unit-I:

Matrices: Rank of a matrix by Echelon form and Normal form, Inverse of Non- singular matrices by Gauss-Jordan method, System of linear equations: Solving system of Homogeneous and Non-Homogeneous equations by Gauss elimination method, Gauss Jacobi and Gauss Seidel Iteration Method.

Unit-II:

Eigen Values and Eigen Vectors: Linear Transformation and Orthogonal transformation: Eigenvalues, Eigenvectors and their properties, Diagonalization of a matrix, Cayley- Hamilton Theorem (without proof), finding inverse and power of a matrix by Cayley- Hamilton Theorem. Quadratic forms and Nature of the Quadratic Forms, Reduction of Quadratic form to canonical forms by Orthogonal Transformation.

Unit-III:

Single Variable Calculus: Limit and Continuous of functions and its properties. Mean value theorems: Rolle's theorem, Lagrange's Mean value theorem with their Geometrical Interpretation and applications, Cauchy's Mean value Theorem, Taylor's Series (All the theorems without proof), Definition of Improper Integral, Beta and Gamma functions and their applications.

Unit-IV:

Multivariable Calculus (Partial Differentiation and applications): Definitions of Limit and Continuity, Partial Differentiation: Euler's Theorem, Total derivative, Jacobian, Functional dependence & independence. Applications: Maxima and minima of functions of two variables and three variables using method of Lagrange multipliers.

Unit-V:

Multivariable Calculus (Integration): Evaluation of Double Integrals (Cartesian and polar coordinates), change of order of integration (only Cartesian form), Change of variables for double integrals (Cartesian to polar), Evaluation of Triple Integrals. Applications: Areas by double integrals and volumes by triple integrals.

Books and Materials

Text Books:

1. Grewal, B. S. *Higher Engineering Mathematics*, 43rd ed., Khanna Publications, 2015.
2. Jain, R. K., Iyengar, S. R. K. *Advanced Engineering Mathematics*, 5th ed., Narosa Publishing House, 2016.

Reference Books:

1. Seymour Lipschutz and Marc Lars Lipson *Schaum's Outline of Linear Algebra* , 6th ed., McGraw-Hill Education, 2018.
2. Greenberg Michael D.*Advanced Engineering Mathematics*, 2nd ed., Upper Saddle River, N.J. Prentice Hall, 1998.
3. Kreyszig, E. *Advanced Engineering Mathematics*, 9th ed., John Wiley & Sons, 2006.
4. Ramana, B. V. *Higher Engineering Mathematics*, 32nd reprint, McGraw Hill Education (India), 2018.

A9009 – Engineering Chemistry

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course emphasizes the application of chemical principles to analyse and address engineering problems, including water and its treatment for diverse purposes, the study of engineering materials such as plastics, fibres, elastomers, and composites, as well as non-conventional energy sources, batteries, and fuel cells. The course aims to integrate theoretical knowledge with practical applications, preparing students to evaluate and implement chemical solutions in engineering contexts.

Course Pre/Co-requisites

This course has no specific prerequisites and co requisites.

Relevant Sustainable Development Goals (SDGs)

SDG 6: Clean Water and Sanitation
SDG 12: Responsible Consumption and Production
SDG 13: Climate Action

Course Outcomes

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

A9009.1. Analyse the hardness and other impurities present in the water for industrial and domestic applications.

A9009.2. Apply electrochemical principles to protect the metals from corrosion.

A9009.3. Illustrate the types of energy sources along with their characteristics and applications.

A9009.4. Differentiate the properties of various polymeric materials based on their structure and engineering applications.

A9009.5. Compare the materials to study various physical and chemical properties.

Course Syllabus

Unit-I:

Water and its treatment: Introduction – hardness of water – causes of hardness – types of hardness: temporary and permanent – expression and units of hardness, numerical problems. Steps involved in the treatment of potable water - disinfection of potable water by chlorination and break-point chlorination. Boiler troubles: sludges, scales and caustic embrittlement. Internal treatment of boiler feed water – Calgon conditioning – Phosphate conditioning – Colloidal conditioning – softening of water by ion exchange processes. Desalination of water – Reverse osmosis.

Unit-II:

Electrochemistry and Corrosion: Introduction- Electrode potential, standard electrode potential, Nernst equation (no derivation), electrochemical cell - Galvanic cell, cell representation, EMF of cell - numerical problems. Types of electrodes, reference electrodes - primary reference electrode - standard Hydrogen Electrode (SHE), Secondary reference electrode - Calomel electrode. Construction, working and determination of pH of an unknown solution using SHE and Calomel electrode.

Corrosion: Introduction- definition, causes and effects of corrosion – theories of corrosion, chemical and electrochemical theories of corrosion, Types of corrosion: galvanic, water-line and pitting corrosion. Factors affecting rate of corrosion - nature of the metal, nature of the corroding environment. Corrosion control methods - electroplating, electroless plating and metal cladding.

Unit-III:

Energy Sources:

Batteries: Introduction – Classification of batteries - Primary, secondary and reserve batteries with examples. Construction, working and applications of Lead acid battery and Lithium ion battery. Fuel Cells – differences between a battery and a fuel cell, construction and applications of Hydrogen-Oxygen fuel cell.

Fuels: Introduction and characteristics of a good fuel, Calorific value – Units - HCV, LCV- Dulongs formula - Numerical problems. *Fossil fuels:* Introduction, Classification, Petroleum - Refining of Crude oil, Cracking - Types of cracking - Moving bed catalytic cracking. LPG and CNG composition and use. *Synthetic Fuels:* Fischer-Tropsch process, Introduction and applications of Hythane and Green Hydrogen.

Unit-IV:

Polymeric Materials: Terminology, types of polymerization – addition and condensation polymerization with examples. Plastics: Thermoplastic resins & Thermosetting resins. Preparation, properties and engineering applications of Polyvinyl chloride and Teflon. Fibers: Preparation, properties and engineering applications of Nylon-6,6 and Dacron. Elastomers: Natural rubber and its vulcanization, artificial rubbers - Buna-S and Butyl rubber. Conducting Polymers: classification, mechanism of conduction in trans - polyacetylene – applications. Biodegradable polymers: Polylactic acid and its applications.

Unit-V:

Advanced Functional Materials:

Graphene: Isolation, Structure and strength, applications in Computer, Electrical and Electronic Devices.

Smart materials: Introduction, Classification with examples - Shape Memory Alloys – Nitinol, Piezoelectric materials – quartz and their engineering applications. Biosensor - Definition, Amperometric Glucose monitor sensor.

Portland cement: Chemical constituents, Setting and Hardening and applications of cement.

Books and Materials

Text Books:

1. Rama Devi, B., Aparna, P., and Prasanta Rath. *Engineering Chemistry.* 2nd ed., Cengage Publications, 2025.
2. Jain, Jain. *Engineering Chemistry.* 16th ed., Dhanpat Rai Publication Company, 2015.

Reference Books:

1. Agarwal, Shikha. *Engineering Chemistry.* Cambridge University Press, Delhi, 2015.
2. Chawla, Shashi. *Engineering Chemistry.* Dhanpat Rai and Company (P) Ltd., Delhi, 2011.
3. Thirumala Chary, M., E. Laxminarayana, and K. Shashikala. *A Textbook of Engineering Chemistry.* Pearson Publications, 2021.
4. Singh, Paramvir, Avinash Kumar Agarwal, Anupma Thakur, and R. K. Sinha. *Challenges and Opportunities in Green Hydrogen.* Springer, 2024.
5. Leo, Donald J. *Engineering Analysis of Smart Material Systems.* John Wiley & Sons, 2007.
6. *E-book:* "Engineering Chemistry by Shashi Chawla." Internet Archive, https://archive.org/details/EngineeringChemistryByShashiChawla/page/n11/mode/2up.

A9501 – Programming for Problem Solving

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course introduces the principles of problem-solving and programming through C language. It begins with the basics of algorithms, flowcharts, and structured program design, enabling students to develop logical thinking skills. Core programming concepts such as variables, operators, control statements, arrays, and strings are covered to build a strong foundation. The course further explores modular programming using functions and recursion, along with structures and unions for handling complex data. Advanced concepts like pointers and dynamic memory management are introduced to enhance program efficiency. File handling techniques are discussed for effective data storage and retrieval. Fundamental searching and sorting algorithms are included to improve problem-solving efficiency and performance analysis. By the end of the course, students will be able to design, implement, and evaluate C programs that solve real-world computational problems systematically and efficiently.

Course Pre/Co-requisites

This Course has no specific Pre/Co requisites

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 8: Decent Work and Economic Growth
SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9501.1. Use basic programming constructs and control statements to design solutions for computational problems.

A9501.2. Develop programs using arrays and strings to store and manipulate sequential data.

A9501.3. Implement modular programming using functions, structures, and unions to manage complex problems and data.

A9501.4. Make use of pointers and file handling to effectively manage and process data.

A9501.5. Choose appropriate searching and sorting technique to organize and retrieve data efficiently.

Course Syllabus

Unit-I:

Problem Solving Techniques: Algorithms- Algorithmic approach, characteristics of algorithm and Examples, Flowcharts- Definition, Symbols and examples, Problem solving strategies: Top-down approach and Bottom-up approach.

Introduction to C: Structure of a C Program, Identifiers, Variables, Constants and Data Types. Operators and Expressions. Precedence of operators and Evaluation of Expressions, Type conversions, Formatted input and output. Control Statements: Conditional Statements- if, if else, nested if, else if ladder and switch statements. Iterative or Loop statements- while, do while and for statements. Jump statements- break, continue and goto statements.

Unit-II:

Arrays and Strings: Arrays: Introduction, One Dimensional Arrays - Declaration and initialization, Reading and Writing. Two Dimensional Arrays - Declaration and initialization, Reading and Writing. Manipulating elements of Arrays. Strings: Introduction, Declaration, and initialization, Reading and writing, string handling functions, handling two dimensional strings.

Unit-III:

Functions, Structures and Unions: Functions- Introduction, Function definition and Function call, Categories of functions, Recursion, Limitations of recursive functions, storage classes, Passing Arrays to functions, Common Pre-processor Directives. Structures- Definition, Declaration, and Initialization, accessing structure members, Array of Structures, Arrays within structures, Structures and functions, size of structures, Unions- Definition, Declaration, and Initialization, accessing Union member.

Unit-IV:

Pointers and Files: Pointers-Declaration, Initialization, Pointer to Pointer, Pointer Arithmetic, Parameter Passing Techniques, Pointer to Arrays, Pointers to Structures. Files- Introduction, defining, opening, and closing a File, Input / Output operations on Files, Random Access in files, Command line arguments.

Unit-V:

Searching and Sorting: Time and Space Complexity, Searching- Linear Search and Binary Search, Sorting- Bubble Sort, Selection Sort, Insertion Sort and Quick Sort.

Books and Materials

Text Books:

1. Thareja, Reema. *Programming in C.* AICTE ed., 2nd rev. ed., Oxford University Press, 2018.
2. Forouzan, Behrouz A., and Richard F. Gilberg. *Computer Science: A Structured Programming Approach Using C*, 3rd ed., reprint, Cengage Learning (formerly Course Technology), 2007.

Reference Books:

1. Kanetkar, Yashavant P. *Let Us C: Authentic Guide to C Programming Language.*, 20th ed., reprint, BPB Publications, 2024.
2. Gottfried, Byron S. *Programming with C.*, 4th ed., reprint, McGraw-Hill Education (India), 2018.
3. Padmanabham, P. *C & Data Structures.*, 3rd ed., B.S. Publications, 2016.
4. Hanly, Jeri R., and Elliot B. Koffman. *Problem Solving and Program Design in C.*, 8th ed., reprint, Pearson, 2015.
5. Balagurusamy, E. *Programming in ANSI C.*, 9th ed., reprint, McGraw-Hill Education India, 2024.

A9011 – English for Skill Enhancement

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
30	0	0	30	60	2	40	60	100

Course Description

Course Overview

This course has been designed to develop linguistic and communicative competencies among engineering students. The Reading and Writing skills of the students are honed during the sessions using the prescribed textbook. Additionally, students are trained on effective usage of grammar and vocabulary. Further, they are encouraged to read texts which are aimed at developing their comprehension skills.

Course Pre/Co-requisites

This course has no specific prerequisite and co-requisite.

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 8: Decent Work and Economic Growth
SDG 17: Partnerships for the Goals

Course Outcomes

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

A9011.1. Identify and use appropriate vocabulary to compose and deliver clear oral and written communication
A9011.2. Practice adept usage of grammar for effective communication
A9011.3. Interpret and summarize known and unknown passages
A9011.4. Develop proficiency in writing for academic purposes
A9011.5. Demonstrate basic proficiency in professional correspondence

Course Syllabus

Unit-I:

Theme: Perspectives
Text: Lesson on 'The Generation Gap' by Benjamin M. Spock
Vocabulary: Word Formation - Prefixes and Suffixes; Synonyms and Antonyms
Grammar: Identifying Common Errors in Writing with reference to Articles and Prepositions, Conjunctions
Reading: Reading and its importance - Sub skills of Reading – Skimming and Scanning
Writing: Sentence Structures - Use of Phrases and Clauses in Sentences - Types of Sentences; Punctuation; Techniques for Writing precisely – Paragraph Writing – Types, Structures and Features of a Paragraph – Creating Coherence - Organizing Principles of Paragraphs in Documents

Unit-II:

Theme: Digital Transformation
Text: Lesson on 'Emerging Technologies'
Vocabulary: Homophones, Homonyms and Homographs
Grammar: Identifying Common Errors in Writing with reference to Tenses, Noun-Pronoun Agreement and Subject-Verb Agreement
Reading: Reading Strategies - Guessing Meaning from Context – Identifying Main Ideas - Exercises for Practice
Writing: Essay writing.

Unit-III:
Theme: Attitude and Gratitude
Text: Poems on 'Leisure' by William Henry Davies and 'Be Thankful' – Unknown Author
Vocabulary: Words often Confused; Phrasal Verbs
Grammar: Misplaced Modifiers
Reading: Sub-Skills of Reading – Identifying Topic Sentence and Providing Supporting Ideas - Exercises for Practice
Writing: Letter Writing: Letter of Request, Letter of Inquiry, Letter of Apology, Letter of Complaint, Email writing - Format, Style and Etiquette.

Unit-IV:
Theme: Entrepreneurship
Text: Lesson on 'Why a Start-up Needs to Find its Customers First' by Pranav Jain
Vocabulary: Standard Abbreviations in English, Idioms
Grammar: Redundancies in Oral and Written Communication, Transformation of sentences - Active and Passive Voice
Reading: Prompt Engineering Techniques – Comprehending and Generating Appropriate Prompts - Exercises for Practice
Writing: Precis Writing; Writing a Letter of Application and Resume/CV.

Unit-V:
Theme: Integrity and Professionalism
Text: Lesson on 'Professional Ethics'
Vocabulary: Technical Vocabulary and its Usage, Collocations
Grammar: Transformation of sentences - Reported Speech, Common Errors covering all other aspects of grammar
Reading: Survey, Question, Read, Recite and Review (SQ3R Method) – Inferring the Meaning and Evaluating a Text - Exercises for Practice
Writing: Technical Reports - Introduction – Characteristics of a Report - Structure of Report (Manuscript Format)

Books and Materials

Text Books:

1. Board of Editors, *English for the Young in the Digital World.* Orient Black Swan Pvt. Ltd. 2025.

Reference Books:

1. Swan, Michael, *Practical English Usage.* Oxford University Press. New Edition, 2016.
2. Karal, Rajeevan, *English Grammar Just for You.* Oxford University Press. New Delhi, 2023.
3. Cengage India, *Empowering with Language: Communicative English for Undergraduates.* Cengage Learning India Pvt. Ltd. New Delhi, 2024.
4. Sanjay Kumar & Pushp Lata, *Communication Skills – A Workbook.* Oxford University Press. New Delhi, 2022.
5. Wood, F.T., *Remedial English Grammar.* Macmillan, 2007.
6. Vishwamohan, Aysha. *English for Technical Communication for Engineering Students.* McGraw-Hill Education India Pvt. Ltd, 2013.

A9402 – Digital Electronics

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course provides a foundational understanding of digital systems and logic design, essential for modern electronic and computing applications. It covers number systems, Boolean algebra, and logic gates, progressing into the design and analysis of combinational and sequential circuits. Emphasis is placed on logic simplification using Karnaugh Maps, as well as the implementation of logic functions using NAND and NOR gates. The course also introduces memory elements such as RAM and ROM, and provides exposure to Programmable Logic Devices, enabling custom logic design.

Course Pre/Co-requisites

This course has no specific prerequisites and co-requisites

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 9: Industry, Innovation, and Infrastructure
SDG 11: Sustainable Cities and Communities

Course Outcomes

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

A9402.1. Apply Boolean algebra and minimization techniques to simplify a Boolean function.

A9402.2. Construct combinational logic circuits using logic gates to implement specified Boolean functions.

A9402.3. Analyze the behavior of latches and flip-flops for designing sequential logic.

A9402.4. Develop synchronous sequential circuits using flip-flops and combinational logic.

A9402.5. Implement digital systems using Programmable Logic Devices for flexible and efficient circuit design.

Course Syllabus

Unit-I:

Number Systems: Binary, Octal, Decimal, Hexadecimal, Complements of Numbers: 1's and 2's Complement. **Boolean Algebra and Logic Gates:** Introduction, basic definitions, axiomatic definition of Boolean algebra, basic theorems and properties of Boolean algebra, Boolean functions, canonical and standard forms (SOP, POS), Digital logic gates, NAND-NOR realization.

Unit-II:

Gate-Level Minimization Techniques: Introduction, The Map method (Karnaugh Map) - 2, 3, and 4 variable K-map, Sum-of-Products and Product-of-Sums Simplification, Don't care conditions.

Unit-III:

Combinational Logic: Introduction, combinational circuits, analysis of combinational circuits, design procedure, binary adder-subtractor, magnitude comparator, decoders, encoders, multiplexers.

Unit-IV:

Sequential Logic: Introduction, sequential circuits, storage elements – latches and flip-flops, flip-flop conversions (JK to SR, JK to D, D to T), Design of counters - Synchronous and asynchronous counters, Shift registers – register with parallel load and serial-in serial-out shift register.

Unit-V:

Memory and Programmable Logic Devices: Random Access Memory (RAM), Read-Only Memory (ROM), Programmable ROM (PROM), Programmable Logic Array (PLA), Programmable Array Logic (PAL).

Books and Materials

Text Books:

1. Mano, M. Morris, and Michael D. Ciletti. Digital Design with an Introduction to the Verilog HDL. 6th ed., Pearson Education/PHI, 2017.

Reference Books:

1. Tocci, Ronald J., Neal S. Widmer, and Gregory L. Moss. Digital Systems: Principles and Applications. 10th ed., Pearson Education International, 2009.
2. Roth, Charles H., Jr., and Larry L. Kinney. Fundamentals of Logic Design. 6th ed., Cengage Learning, 2009.

A9010 – Engineering Chemistry Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

The Engineering Chemistry Laboratory equips students with practical skills essential for understanding the chemical principles behind engineering materials and processes. It bridges theoretical knowledge with real-world applications, fostering analytical thinking and precision. Students learn to handle instruments, analyze data, and interpret results relevant to industrial and environmental contexts. The course emphasizes the role of chemistry in addressing engineering challenges and societal needs. Overall, it builds a strong foundation for innovation and responsible technological development.

Course Pre/Co-requisites

This course has no specific prerequisites and co requisites.

Relevant Sustainable Development Goals (SDGs)

SDG 6: Clean Water and Sanitation
SDG 12: Responsible Consumption and Production
SDG 13: Climate Action

Course Outcomes

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

A9010.1. Apply the instrumental techniques to find out the strength of solutions.
A9010.2. Analyze the impurities present in the water using volumetric analysis.
A9010.3. Make use of different titrimetric methods to measure chemical species.
A9010.4. Analyze the importance of temperature and pressure on physical properties of liquids.
A9010.5. Calculate the yield of synthesized compounds by maintaining appropriate reaction conditions.

Course Syllabus

List of Experiments:

1. Estimation of amount of ferrous ion in the given solution by permanganometry.
2. Estimation of hardness of water by complexometry using EDTA.
3. Estimation of amount of hydrochloric acid in the given solution by conductometry.
4. Estimation of amount of strong and weak acid in the given solution by conductometry.
5. Estimation of amount of hydrochloric acid in the given solution by potentiometry.
6. Estimation of amount of ferrous ion in the given solution using potassium permanganate by potentiometry.
7. Estimation of manganese ion in the given solution by colorimetry.
8. Estimation of Copper ion in the given solution by colorimetry.
9. Determination of viscosity of the given liquid by Ostwald's viscometer.
10. Determination of surface tension of the given liquid by using stalagmometer.

11. Preparation of Bakelite.
12. Preparation of Nylon 6,6.

Laboratory Equipment/Software/Tools Required:

1. Digital Conductometer
2. Digital Potentiometer
3. Digital Colorimeter
4. Electrical Water Heater
5. Wall Mount Distillation Plant
6. Analytical/Digital Weighing Balance
7. Ostwald's Viscometer
8. Stalagmometer
9. Stopwatch
10. Thermometer
11. RB Flask condenser
12. Magnetic Stirrer
13. Pipette
14. Burette
15. Beaker

Books and Materials

Text Books:

1. Rama Devi, B., Aparna, P., and Prasanta Rath. *Engineering Chemistry.* 2nd ed., Cengage Publications, 2025.

Reference Books:

1. Vogel, A. I. *Inorganic Quantitative Analysis.* ELBS Publications.
2. Ahluwalia, V. K. *College Practical Chemistry.* Narosa Publications Ltd., 2007

A9502 – Programming for Problem Solving Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This course aims to build practical programming skills using the C language. Students learn to approach problems logically and implement solutions efficiently. Emphasis is given to writing clear and structured programs using control statements and modular design. They gain hands on experience with data handling, including arrays, strings, and user-defined data types. Pointers are introduced to manage memory and work with complex data efficiently. File operations are covered to handle data storage and retrieval. Students practice implementing algorithms for sorting, searching, and numerical computations. The course develops debugging and problem-solving abilities through practical exercises. Focus is placed on optimizing code for better performance and readability. By the end, learners can design and implement robust C programs for a variety of computational problems.

Course Pre/Co-requisites

This course has no specific pre-requisites and co-requisites.

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 8: Decent Work and Economic Growth
SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9502.1. Make use of fundamental programming constructs to develop solutions for computational problems.

A9502.2. Perform various operations on arrays and strings to effectively organize, process, and manipulate sequential data in programs.

A9502.3. Develop programs with functions and structures to design modular programs that efficiently handle and process data.

A9502.4. Apply pointers and file handling techniques to implement programs for storing and managing data effectively.

A9502.5. Implement searching and sorting algorithms to efficiently organize and access data.

Course Syllabus

List of Experiments:

1. Variables and Expressions
   a. Write a C program for Swapping of two numbers using a third variable.
   b. Write a C program for the simple and compound interest.
   c. Write a C program to evaluate the expressions (Finding $y = m \cdot x + c$, displacement).
2. Conditional Statements–I
   a. Write a C program for finding the max and min from the three numbers.
   b. Write a C program to check the given year is leap year or not.

c. Write a C program to find the roots of a quadratic equation.

3. Conditional Statements–II
   a. Write a C program to check the given number is power of 2 or not using bitwise operators.
   b. Write a C program to read 3 subject marks. Calculate and display the grade of a student based on the percentages.
   c. Write a C program to perform Arithmetic Operations using switch statement.
4. Iterative Statements–I
   a. Write a C program to find sum of n natural numbers $(1+2+3+\cdots+n)$.
   b. Write a C program to find factorial of a given number.
   c. Write a C program to print Fibonacci numbers.
   d. Write a C program to find reverse of the given number.
   e. Write a C program to check if the binary representation of a positive number is palindrome or not. (Examples: 101, 11, 11011, 1001001 are palindromes. 100, 110, 1011 are not).
5. Iterative Statements–II
   a. Write a C program to read a password until it is correct. For wrong password print "Incorrect password" and for correct password print "Correct password" and quit the program. (The correct password is 1234).
   b. Write a C program to check the given number is prime or not.
   c. Write a C program to find the GCD of given two numbers.
   d. Write a C program to print the output in various triangle patterns using nested `for` loops.
   e. Write a C program to find the sum of the series Geometric Progression.
6. Arrays
   a. Write a C program to find the largest and smallest number among a list of integers.
   b. Write a C program to read an array of n elements and find the mean, variance, and standard deviation.
   c. Write a C program to find addition of two matrices.
   d. Write a C program to find multiplication of two matrices.
7. Strings
   a. Write a C program to demonstrate the string handling functions.
   b. Write a C program to check whether a given string is palindrome or not.
   c. Write a C program to concatenate three strings.
   d. Write a C program to count the lines, words and characters in a given text.
   e. Write a C program that displays the position of a character ch in the string S or -1 if S doesn't contain ch.
8. Functions
   a. Write a C program to find the factorial of a given number using non-recursive and recursive function.
   b. Write a C program to find the nth term of a Fibonacci series using recursive function.
   c. Write a C program to compute x^y.
9. Structures
   a. Write a C program to create a Student structure containing name, rollNo and grade as structure members. Display the name, rollNo and grade of a student.

b. Write a C program to create a Student structure containing name, rollNo and grade as structure members. Display the name, rollNo and grade of n students by using array of structures concept.

10. Structures and functions
    a. Write a C program to add two complex numbers by passing structure to a function.
    b. Write a C program to add two distances (in inch-feet system) using structures.
11. Pointers
    a. Write a C program to swap two integers using the following methods:
        i. Call by Value
        ii. Call by Reference
    b. Write a C program to demonstrate pointer arithmetic.
    c. Write a program to display values in reverse order from an array using a pointer.
    d. Write a program through a pointer variable to find sum of n elements from an array.
    e. Write a C program to check the given string is palindrome or not using pointer.
    f. Write a C program to print n city names using pointers and strings.
12. Files
    a. Write a C program to merge two files into a third file.
    b. Write a C program to reverse the contents of a file.
    c. Write a C program to use random access functions in files.
    d. Write a C program to count the number of times a character occurs in a text file (file name and character are supplied as command line arguments).
13. Searching
    a. Write a C program that uses a non-recursive function to search for a key value in a list of integers using linear search.
    b. Write a C program that uses a non-recursive function to search for a key value in a sorted list of integers using binary search.
14. Sorting
    a. Write a C program that implements the Bubble sort method to sort a given list of integers in ascending order.
    b. Write a C program that sorts the given array of integers using selection sort in descending order.
    c. Write a C program that sorts the given array of integers using quick sort in ascending order.
    d. Write a C program that sorts the given array of integers using insertion sort in ascending order.
15. Miscellaneous
    1. Write a program that shows the binary equivalent of a given positive number between 0 to 255.
    2. Write a C program to calculate the following, where x is a fractional value:

$$1 - \frac{x}{2} + \frac{x^2}{4} - \frac{x^3}{6}$$

    3. Write a C program to read in two numbers, `x` and `n`, and then compute the sum of this geometric progression:

$$1 + x + x^2 + x^3 + \cdots + x^n$$

    For example: if $n = 3$ and $x = 5$, the program computes $1 + 5 + 25 + 125$.

4. Write a C program to convert a Roman numeral ranging from I to L to its decimal equivalent.
5. Write a C program that converts a number ranging from 1 to 50 to its Roman equivalent.
6. Write a C program that uses functions to perform the following operations:
   i. To insert a sub-string into a given main string from a given position.
   ii. To delete n characters from a given position in a given string.
7. Write a C program to merge two files into a third file (i.e., the contents of the first file followed by those of the second are put in the third file).
8. Write a program for display values reverse order from an array using a pointer.
9. Write a program through a pointer variable to sum of n elements from an array.
10. Write a C program that sorts the given array of integers using insertion sort in ascending order

Laboratory Equipment/Software/Tools Required:

1. Computer Systems (PCs) installed with Ubuntu OS (Open source/ Freeware)
2. GCC Compiler (Open source/ Freeware)

Books and Materials

Text Books:

1. Thareja, Reema. *Programming in C.* AICTE ed., 2nd rev. ed., Oxford University Press, 2018.
2. Forouzan, Behrouz A., and Richard F. Gilberg. *Computer Science: A Structured Programming Approach Using C.*, 3rd ed., reprint, Cengage Learning (formerly Course Technology), 2007.

Reference Books:

1. Kanetkar, Yashavant P. *Let Us C: Authentic Guide to C Programming Language*, 20th ed., reprint, BPB Publications, 2024.
2. Gottfried, Byron S. *Programming with C*, 4th ed., reprint, McGraw-Hill Education (India), 2018.
3. Padmanabham, P. *C & Data Structures*, 3rd ed., B.S. Publications, 2016.
4. Hanly, Jeri R., and Elliot B. Koffman. *Problem Solving and Program Design in C*, 8th ed., reprint, Pearson, 2015.
5. Balagurusamy, E. *Programming in ANSI C*, 9th ed., reprint, McGraw-Hill Education India, 2024.

A9012 – English Language and Communication Skills Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This course is designed to cater to the needs of students in developing their oral communication skills. It begins with an introduction to Phonetics to make them understand the received pronunciation and to help them speak with a neutral accent. This course incorporates listening skills and draws exercises of listening comprehension from various general and business contexts. The speaking exercises in this course will help the students to present their ideas in different situations, besides helping them to develop team spirit by participating in pair/group activities.

Course Pre/Co-requisites

This course has no specific prerequisite and co-requisite.

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 8: Decent Work and Economic Growth
SDG 17: Partnerships for the Goals

Course Outcomes

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

A9012.1. Acquire the received pronunciation and speak in a neutral accent
A9012.2. Use language effectively in real-life situations
A9012.3. Demonstrate effective use of non verbal communication
A9012.4. Interpret visual data for oral communication
A9012.5. Develop the ability to enhance listening skills

Course Syllabus

List of Experiments:

1. CALL Lab:

Instruction: Speech Sounds - Listening Skills - Listening vs. Hearing - Importance – Purpose - Types
Practice: Listening to Distinguish Speech Sounds (Minimal Pairs) - Testing Exercises

2. ICS Lab:

Diagnostic Test: Activity titled 'Express Your View'
Instruction: Spoken and Written language - Formal and Informal English - Greetings – Introducing Oneself and Others
Practice: Any Ice-Breaking Activity

3. CALL Lab

Instruction: Barriers to Listening - Active Listening
Practice: Listening for General Information - Multiple Choice Questions - Listening Comprehension - Exercises for practice

4. ICS Lab:
Instruction: Features of Good Conversation – Strategies for Effective Communication
Practice: Role Play Activity - Situational Dialogues – Expressions used in Various Situations – Making Requests and Seeking Permissions – Taking Leave - Telephone Etiquette

5. CALL Lab
Instruction: Minimizing Errors in Pronunciation (MTI)
Practice: Differences between British and American Pronunciation – Listening Comprehension – Exercises for practice

6. ICS Lab:
Instruction: Describing Objects, Situations, Places, People and Events
Practice: Picture Description Activity – Looking at a Picture and Describing Objects, Situations, Places, People and Events

7. CALL Lab:
Instruction: Techniques for Effective Listening
Practice: Listening for Specific Details - Listening - Gap Fill Exercises - Listening Comprehension – Exercises for practice

8. ICS Lab:
Instruction: Information transfer - oral interpretation of graphical data
Practice: Activity on oral interpretation of graphical data

9. CALL Lab:
Instruction: Identifying the literal and implied meaning
Practice: Listening for Evaluation - Write the Summary – Listening Comprehension – Exercises for practice

10. ICS Lab:
Instruction: Understanding Non-Verbal Communication
Practice: Dumb Charades Activity
Post-Assessment Test: 'Express Your View'

Laboratory Equipment/Software/Tools Required:

1. Computers with internet
2. K VAN Solutions Software
3. Headphones
4. Audio Visual Equipment
5. Camcorder

Books and Materials

Lab Manual:

1. Laboratory Handbook on English Language and Communication Skills Lab.

Reference Books:

1. Brook-Hart, Guy. *Cambridge English Business Benchmark - Upper Intermediate Business Vantage (with CD)*, 2nd Edition, South Asian Edition, Cambridge University Press, 2019.
2. Shobha, KN & Rayen, J. Lourdes. *Communicative English – A workbook.* Cambridge University Press, 2019.
3. Board of Editors. *ELCS Lab Manual: A Workbook for CALL and ICS Lab Activities.* Orient BlackSwan Pvt. Ltd. , 2016.
4. Mishra, Veerendra et al. *English Language Skills: A Practical Approach.* Cambridge University Press, 2020.
5. *English Language Communication Skills – Lab Manual cum Workbook.* Cengage Learning India Pvt. Ltd.
6. Ur, Penny and Wright, Andrew. *Five Minute Activities – A Resource Book for Language Teachers.* Cambridge University Press, 2022.
7. *TOEFL & GRE (KAPLAN, AARCO & BARRONS, USA, Cracking GRE by CLIFFS).*

A9304 - Computer Aided Engineering Graphics

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This course provides students from diverse engineering disciplines with essential skills in computer-aided engineering drawing using AutoCAD. It focuses on the fundamentals of constructing two-dimensional geometric objects, understanding orthographic projections of points, lines, planes, and solids, and applying isometric projections. Students will learn to use AutoCAD tools such as Draw, Modify, Layers, and Dimensioning to create technically accurate drawings aligned with engineering standards. The course emphasizes practical applications relevant to civil, electrical, electronics, and other engineering fields, enabling students to visualize, interpret, and communicate design concepts effectively.

Course Pre/Co-requisites

This course has no specific pre-requisites and co-requisites.

Relevant Sustainable Development Goals (SDG(s))

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

SDG 12: Responsible Consumption and Production

Course Outcomes

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

A9304.1. Analyze the basic drawing and editing tools to create and modify 2D sketches.

A9304.2. Interpret the projection principles to draw points and lines in different quadrants.

A9304.3. Compare the projected views of planes to identify their true shape and inclination.

A9304.4. Apply the orthographic projection principles to construct two-dimensional views of solids.

A9304.5. Construct isometric views by applying principles derived from orthographic drawings.

Course Syllabus

Unit-I:

Introduction to AutoCAD: User Interface and Workspace Customization, Basic Drawing Tools, Modify and Editing Tools, Properties and Object Management, Layer Management, Dimensioning and Annotation, Layouts and Plotting, Geometrical construction of two-dimensional objects.

Unit-II:

Orthographic Projections: Reference plane, importance of reference lines or Plane, Projections of a point situated in any one of the four quadrants. Projections of Straight Lines: Projections of straight lines parallel to both reference planes, perpendicular to one reference plane and parallel to other reference plane, inclined to one reference plane and parallel to the other reference plane.

Unit-III:

Projections of Planes: Regular planes Perpendicular to both reference planes, parallel to one reference plane and inclined to the other reference plane; plane inclined to both the reference planes.

Unit-IV:

Projections of Solids: Types of solids: Polyhedra and Solids of revolution. Projections of solids in simple positions: Axis perpendicular to horizontal plane, Axis perpendicular to vertical plane and Axis parallel to both the reference planes, Projection of Solids with axis inclined to one reference plane and parallel to another plane.

Unit-V:

Isometric Projections: Isometric coordinates, Isometric Scale, Isometric Views of Lines, Planes and solids. Conversion of Isometric View to Orthographic View and Vice-versa.

Books and Materials

Text Books:

1. Bhatt N.D., Panchal V.M. & Ingle P.R., *Engineering Drawing*, 53rd Edition, Charotar Publishing House, 2019.
2. K. Balaveera Reddy et al, *Computer Aided Engineering Drawing*, 2nd Edition, CBS Publications, 2015.

Reference Books:

1. Narayana, K.L. & P Kannaiah, *Text book on Engineering Drawing*, 3rd Edition, Sci-Tech Publishers, 2020.
2. Basant Agrawal B. and Agrawal C. M., *Engineering Graphics*, 3rd Edition, TMH Publication, 2020.
3. Shah, M.B., Rana B.C., *Engineering Drawing and Computer Graphics*, 2nd Edition, Pearson Education, 2009.

A9021 - Community Centered Design Thinking

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	0	45	45	1	40	60	100

Course Description

Course Overview

The "Community-Centered Design Thinking" course aims to enable students to identify and address unique needs and challenges within local communities. Through the application of design thinking principles, students will develop creative problem-solving mindsets and the ability to collaborate effectively in multidisciplinary teams. The course emphasizes integrating moral code, professional standards, and sustainability principles into design solutions.

Course Pre/Co-requisites

This course has no specific prerequisite and co-requisite.

Relevant Sustainable Development Goals (SDGs)

SDG 3: Good Health and Well-being
SDG 4: Quality Education
SDG 11: Sustainable Cities and Communities
SDG 17: Partnerships for the Goals

Course Outcomes

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

A9021.1. Apply the principles of design thinking, empathy, and sustainability to identify and understand real-world community challenges.

A9021.2. Conduct field research surveys and observation to define community-based problem statements.

A9021.3. Ideate creative solutions using appropriate tools and techniques to meet the identified community needs.

A9021.4. Collaborate with community members, NGOs, and peers to test, refine, and validate design solutions through feedback and co-design processes.

A9021.5. Communicate design outcomes effectively through documentation, storytelling, and ethical reflection considering accessibility, inclusivity, and life-cycle impact.

Course Syllabus

Unit-I:

Community-Centered Design Thinking: Understanding the significance of community-centered approaches, Overview of Design Thinking principles for community engagement.

Unit-II:

Needs and Challenges Assessment: Techniques for identifying and analyzing unique needs within local communities, SDGs Alignment, Case studies illustrating successful community-centered design projects.

Unit-III:

Research and Comparative Analysis for Innovation: Investigating existing solutions to community challenges, comparing their effectiveness, and identifying opportunities to create improved, innovative approaches.

Unit-IV:

Ethical Design and Sustainability: Integrating moral code and professional standards into the design process. Incorporating sustainability principles in design to define socially responsible solutions.

Unit-V:

Refine Problem Statement: study existing solutions, and generate creative, community-focused ideas, with all findings documented for the next stage of development.

Activity Plan

Week	Unit	Objective	Teaching Method	In-Class Activities	Assignments / Assessments	CO Mapping
1	Unit-I: CCDT	Understand significance of community centered approaches	Concept Briefing + Discussion	Icebreaker: list visible community challenges	Reflection notes: "Why is community engagement critical?"	CO1
2	Unit-I: CCDT	Learn design thinking principles	Concept Briefing + Activity	Mini design thinking cycle for sample issue	Mind-map of 5 stages applied to local issue	CO1
3	Unit-II: Needs Assessment	Acquire techniques for identifying needs	Concept Briefing + Hands on Session	Practice mock interviews, empathy mapping	Conduct mini survey (3–5 people)	CO2
4	Unit-II: Needs Assessment	Connect needs to SDGs	Concept Briefing + Case Study	Group analysis: link issues to SDGs	Case study report (2–3 pages)	CO2
5	Unit-III: Research	Investigate existing solutions	Guided Research	Group research on 2–3 existing interventions	Summary table of solutions	CO3
6	Unit-III: Comparative Analysis	Compare effectiveness of solutions	Presentations	Present comparison of solutions (pros/cons)	Comparative chart submission	CO3
7	Unit-IV: Ethics	Integrate ethics into design	Debate	Debate: "Should cost outweigh ethics?"	Short essay on ethical dilemma	CO4
8	Unit-IV: Sustainability	Apply sustainability principles	Hands on Session	Create sustainability checklist for ideas	Submit checklist + reflection	CO4
9	Unit-V: Refinement	Refine problem statements	Guided Group Work	Rewrite into "How Might We..." questions	Final problem statement submission	CO2, CO5

Week	Unit	Objective	Teaching Method	In-Class Activities	Assignments / Assessments	CO Mapping
10	Unit-V: Refinement	Study existing solutions deeply	Problem Statement Review	Critique and identify gaps in solutions	Gap Analysis Report	CO3, CO5
11	Unit-V: Ideation	Generate creative ideas	Ideation Hands-on Session	Brainstorming, SCAMPER, Story boarding	Sketches/story boards of top 3 ideas	CO3, CO5
12	Unit-V: Documentation & Presentation	Present and document findings	Final Showcase	Group presentations + reflection sharing	Final report + individual reflection essay	CO5

Books and Materials

Text Books:

1. Pavan Soni. *Design Your Thinking*, Penguin Random House India, New Delhi, 2020.
2. Anuja Agarwal. *Design Thinking: A Framework for Applying Design Thinking in Problem Solving*, Cengage India, 2024.

Reference Books:

1. Srinivasan R., Mohammed Ismail, Arulmozhi Srinivasan. *Design Thinking: Principles, Processes and Applications*, S Chand Publishing, 2025.

I B.Tech. II Semester

A9002 – Ordinary Differential Equations and Vector Calculus

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	15	0	60	120	4	40	60	100

Course Description

Course Overview

This course provides the fundamental mathematical concepts and techniques essential for engineering applications. In this course, the students are acquainted with ordinary differential equations of first and higher order and Laplace transforms, vector calculus. The course is designed to build conceptual clarity and problem-solving skills, with emphasis on both theoretical understanding and practical applications.

Course Pre/Co-requisites

This course has no specific prerequisite and co-requisite.

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9002.1. Make use of first order differential equations to solve real world problems.

A9002.2. Solve ordinary differential equations of higher order.

A9002.3. Apply Laplace transforms to solve ordinary differential equations.

A9002.4. Determine divergence and curl of a vector point function.

A9002.5. Compute line, surface, and volume integrals and convert them into one another using appropriate theorems.

Course Syllabus

Unit-I:

First Order Ordinary Differential Equations: Exact differential equations, Equations reducible to exact differential equations, linear and Bernoulli's equations, Orthogonal Trajectories (only in Cartesian Coordinates). Applications: Newton's law of cooling, Law of natural growth and decay.

Unit-II:

Ordinary Differential Equations of Higher Order: Higher order linear differential equations with constant coefficients: Non-homogeneous terms of the type e^{ax}, $\sin ax$, $\cos ax$, polynomials in x, $e^{ax}V(x)$ and $xV(x)$, method of variation of parameters.

Unit-III:

Laplace Transforms: Laplace Transform of standard functions, First shifting theorem, Laplace transforms of functions multiplied by 't' and divided by 't', Laplace transforms of derivatives and integrals of function, Evaluation of integrals by Laplace transforms, Laplace transform of periodic functions, Inverse Laplace transform by different methods, convolution theorem (without proof). Applications: solving Initial value problems by Laplace Transform method.

Unit-IV:

Vector Differentiation: Vector point functions and scalar point functions, Gradient, Directional derivatives, Divergence and Curl, Vector Identities, Scalar potential functions, Solenoidal and Irrotational vectors.

Unit-V:

Vector Integration: Line integral, Surface integral and Volume Integral. Theorems of Green, Gauss and Stokes (without proofs) and their applications.

Books and Materials

Text Books:

1. Grewal, B. S. *Higher Engineering Mathematics*, 43rd ed., Khanna Publications, 2015.
2. Jain, R. K., Iyengar, S. R. K. *Advanced Engineering Mathematics*, 5th ed., Narosa Publishing House, 2016.

Reference Books:

1. Raisinghania M.D. *Ordinary and Partial Differential Equations* , 20th ed., S. Chand Publishing, 2024.
2. Greenberg Michael D.*Advanced Engineering Mathematics*, 2nd ed., Upper Saddle River, N.J. Prentice Hall, 1998.
3. Kreyszig, E. *Advanced Engineering Mathematics*, 9th ed., John Wiley & Sons, 2006.
4. Ramana, B. V. *Higher Engineering Mathematics*, 32nd edition reprint, McGraw Hill Education (India), 2018.

A9007 – Engineering Physics

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

The Engineering Physics course introduces the fundamental principles of quantum mechanics, semiconductor physics, quantum computing, magnetic and dielectric materials, as well as lasers and fibre optics. Students will explore theoretical foundations, material properties, and device concepts alongside their practical applications in modern technologies such as electronics, communication, sensing, and computing systems. This course bridges core physics concepts with real-world innovations, preparing learners for advanced studies and research in emerging technologies.

Course Pre/Co-requisites

This course has no specific prerequisites and co requisites.

Relevant Sustainable Development Goals (SDGs)

SDG 7: Affordable and Clean Energy
SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9007.1. Apply quantum mechanical principles to understand the particle behavior and formation of energy bands in solids.

A9007.2. Analyze semiconductor properties and explain the operation of P–N junction diode and their applications.

A9007.3. Apply quantum gates to design quantum circuits and implement fundamental quantum algorithms.

A9007.4. Analyze magnetic and dielectric properties relevant to modern technological applications.

A9007.5. Apply laser and fibre optic principles to communication and sensing technologies.

Course Syllabus

Unit-I:

Quantum Mechanics: Introduction, de-Broglie hypothesis, Heisenberg uncertainty principle, physical significance of wave function, postulates of quantum mechanics: operators in quantum mechanics, eigen values and eigen functions, Schrödinger's time independent wave equation, particle in a 1D box, Bloch's theorem (qualitative), Kronig-Penney model (qualitative): E-k diagram, effective mass of electron, formation of energy bands, origin of band gap, classification of solids, concept of discrete energy levels.

Unit-II:

Semiconductor Physics: Intrinsic semiconductors, density of states, Fermi-Dirac distribution function, carrier concentration in intrinsic semiconductors, direct and indirect band gap semiconductors, extrinsic semiconductors, characteristics of P-N junction diode, applications: Light Emitting Diode (LED), solar cell, Hall effect.

Unit-III:

Quantum Computing: Introduction, linear algebra for quantum computation, Dirac's Bra and Ket notation and their properties, Hilbert space, Bloch's sphere, concept of quantum computer, classical bits, Qubits, multiple Qubit system, quantum computing system for information processing, evolution of quantum systems, quantum measurements, entanglement, quantum gates (Hadamard, CNOT, Toffoli), challenges and advantages of quantum computing over classical computation, Introduction to quantum algorithms: Deutsch-Jozsa, Shor, Grover (Qualitative).

Unit-IV:

Magnetic and Dielectric Properties: Introduction to magnetic materials, origin of magnetic moment, classification of magnetic materials (dia, para, ferro), Weiss domain theory of ferromagnetism, hysteresis, soft and hard magnetic materials, applications: magnets for EV, Giant Magneto Resistance (GMR) device. Introduction to dielectric materials, types of polarization (qualitative): electronic, ionic & orientational; ferroelectric, piezoelectric, pyroelectric materials and their applications: Ferroelectric Random-Access Memory (Fe-RAM), load cell and fire sensor.

Unit-V:

Lasers and Fibre Optics: Introduction to laser, characteristics of laser, Einstein coefficients and their relations, metastable state, population inversion, pumping, lasing action, Ruby laser, He-Ne laser, semiconductor diode laser, applications: Bar code scanner, LIDAR for autonomous vehicle. Introduction to fibre optics, total internal reflection, construction of optical fibre, acceptance angle, numerical aperture, classification of optical fibres, losses in optical fibres, applications: optical fibre for communication system, sensor for structural health monitoring.

Books and Materials

Text Books:

1. T. Vijaya Krishna, T. Madhu Mohan, B.K. Pandey, Manoj K. Harbola, and S. Chaturvedi. *Physics for Engineers.* 2nd ed., Cengage, 2024.
2. M. N. Avadhanulu, P. G. Kshirsagar, and T. V. S. Arun Murthy. *A Textbook of Engineering Physics.* 13th ed., S. Chand & Company Pvt. Ltd., 2023.
3. Thomas G. Wong. *Introduction to Classical and Quantum Computing.* Rooted Grove.

Reference Books:

1. Jozef Gruska. *Quantum Computing.* McGraw Hill Education, 1999.
2. Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2010.
3. John M. Senior. *Optical Fiber Communications: Principles and Practice.* Pearson Education Limited, 2009.

A9204 – Basic Electrical Engineering

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
30	0	0	30	60	2	40	60	100

Course Description

Course Overview

Basic Electrical Engineering course serves as a theoretical foundation aimed at enriching students' understanding of electric circuits, DC and AC machines, while fostering analytical abilities. This course delves into the foundational concepts and methodologies integral to Electrical Engineering, covering various aspects such as electrical circuits, network theorems, and operational principles of key components including DC machines (motors and generators), Transformers, Induction motors, and Synchronous generators.

Course Pre/Co-requisites

This course has no specific pre requisites or co-requisites.

Relevant Sustainable Development Goals (SDGs)

SDG 7: Affordable and Clean Energy
SDG 9: Industry, Innovation, and Infrastructure
SDG 11: Sustainable Cities and Communities

Course Outcomes

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

A9204.1. Apply network reduction techniques and theorems to solve DC circuits.

A9204.2. Apply AC circuit concepts to solve and analyze single-phase series R, L, and C circuits

A9204.3. Apply the principles of transformer to determine EMF, efficiency, and performance under no-load and load conditions.

A9204.4. Apply the concepts of DC machines to determine their performance under different operating conditions.

A9204.5. Apply AC machine principles to determine the characteristics and behavior of three phase induction and synchronous machines.

Course Syllabus

Unit-I:

DC Circuits: Electrical circuit elements (R, L and C), Ohm's Law, KVL and KCL, Types of sources, Source transformation, Network reduction techniques (Series and Parallel), Mesh and Nodal analysis, Superposition theorem, Thevenin's and Norton's theorems(DC Excitation only) - Numerical problems.

Unit-II:

AC Circuits: Representation of sinusoidal waveforms, Average & RMS value, Peak factor, Form factor, j-notation, Analysis of single-phase AC circuits consisting of R, L, C, RL, RC, RLC combinations (series circuits only) - Numerical problems.

Unit-III:

Single Phase Transformers: Working principle and constructional details, Types-Core and Shell type transformers, EMF equation, Transformer operation on NO load and ON load Conditions, OC and SC tests on Transformer, Losses and Efficiency - Numerical problems on EMF equation.

Unit-IV:

DC Machines: D.C. Generators - Construction, Principle of operation, E.M.F. equation, Methods of excitation - Separately excited and Self-excited generators- Numerical problems on EMF equation. D.C Motors - Principle of operation, Concept of Back E.M.F., Torque equation, Load test on DC Shunt motor - Conceptual description only.

Unit-V:

AC Machines: Generation of rotating magnetic fields, Construction and working of a three-phase Induction motor, Concept of slip, Torque production- Starting and Running torques, Torque-Slip characteristics - Numerical problems on slip. Construction of Synchronous generator-Salient pole and Non-salient pole generators, working principle of synchronous generator, No-Load Characteristics - Conceptual description only.

Books and Materials

Text Books: Text Books:

1. Hayt, William H., Jack E. Kemmerly, and Steven M. Durbin. *Engineering Circuit Analysis.* 9th ed., McGraw-Hill Higher Education, 2007.
2. Theraja, B. L., and A. K. Theraja. *A Textbook of Electrical Technology.* Vols. 1–2, S. Chand Publishers, New Delhi.

Reference Books:

1. Kothari, D. P., and I. J. Nagrath. *Basic Electrical Engineering.* 3rd ed., Tata McGraw-Hill, 2010.
2. Kulshreshtha, D. C. *Basic Electrical Engineering.* McGraw-Hill, 2009.
3. Bobrow, L. S. *Fundamentals of Electrical Engineering.* Oxford University Press, 2011.

A9503 – Data Structures

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course introduces C++ programming with a focus on designing and implementing data structures. It covers object-oriented concepts such as classes, encapsulation, and abstraction for modular programming. Students learn to use pointers and dynamic memory for efficient data management. Templates and the Standard Template Library (STL) are taught for generic and reusable code. Linked lists are explored for dynamic data storage and sequential operations. Stacks and queues are implemented for practical applications like expression evaluation and task management. Tree structures are studied to represent hierarchical data with traversal and search methods. Binary search trees and AVL trees are covered to ensure organized and balanced data storage. Graph concepts and algorithms are introduced to model and navigate complex relationships. Hashing and dictionary structures provide fast data retrieval and efficient storage solutions. The course emphasizes hands-on programming to reinforce theoretical concepts. By the end, students can design, implement, and analyze core data structures effectively in C++.

Course Pre/Co-requisites

A9501 – Programming for Problem Solving

A9502 – Programming for Problem Solving Laboratory

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

- A9503.1. Make use of programming constructs, Templates and STL components to implement data structures and solve computational problems effectively.
- A9503.2. Implement linked list data structures using abstract data types to perform various operations.
- A9503.3. Implement stack and queue abstract data types for solving applications on linear data.
- A9503.4. Choose an appropriate nonlinear data structure for representing and solving real world problems.
- A9503.5. Examine hashing and dictionary structures for insertion, deletion, and searching in computational problems.

Course Syllabus

Unit-I:

Introduction to C++ and Data Structures: Structure of a C++ Program, Classes and objects, Dynamic memory allocation (new and delete), Constructors and destructors, Access specifiers (public, private), Encapsulation and abstraction, Pointers and references, Functions, Inline functions, and function overloading. Template Definition, Function Templates, class Templates, Generic Function, Template function Overloading. Standard Template Library (STL) Overview- Containers: vector, list, map, set. Iterators and algorithms, Using STL templates in practical applications.

Unit-II:

Linked Lists: Classification of Data Structures, Abstract Data Types, Introduction to Linked Lists, Applications of Linked Lists, Node Structure and Memory Allocation. Linked List ADT, Singly Linked List Operations-Insert, Delete, Search and Traverse. Doubly Linked List Operations – Insert, Delete, Search and Traverse and Circular Linked Lists.

Unit-III:

Stacks and Queues: Stack ADT – Concepts and Applications. Array and Linked List implementation of Stacks. Applications – Conversion of Expression from infix to postfix, Evaluation of Postfix Expression. Queue ADT – Concepts & Applications. Array and Linked List implementation of Linear Queues, Array Implementation of Circular Queue and Double Ended Queue.

Unit-IV:

Trees and Graphs: Tree ADT, Binary Tree Terminology and Properties, Binary Tree Traversals – Inorder, Preorder and Postorder. Binary Search Tree Operations – Insertion, Deletion and Searching. AVL Trees, Definition, Height of an AVL Tree, Operations – Insertion, Deletion and Searching. Graph Definitions and Terminology. Graph Representations – Adjacency Matrix and Adjacency List. Graph Traversals – Depth-First Search (DFS) and Breadth-First Search (BFS).

Unit-V:

Dictionaries and Hashing: Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing, linear probing, quadratic probing, double hashing, rehashing, extendible hashing. Dictionaries-linear list representation, skip list representation, operations - insertion, deletion and searching.

Books and Materials

Text Books:

1. Horowitz, Ellis, Sartaj Sahni, and Dinesh Mehta. *Fundamentals of Data Structures in C++*, 2nd ed., Universities Press, 2019.
2. Horowitz, *Data Structures Using C++*, 3rd edition, Course Technology, 2010.

Reference Books:

1. Drozdek, Adam. *Data Structures and Algorithms in C++*, 5th ed., Cengage Learning, 2025.
2. Dale, Nell, Chip Weems, and Tim Richards. *C++ Plus Data Structures*, 6th ed., Jones & Bartlett Learning, 2018.

A9505 – Computer Organization

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course offers a detailed study of how a computer system is structured and operates, focusing on the flow of data and control between its components. It explains how instructions are represented, executed, and managed within a processor, along with the mechanisms that support data transfer and storage. Students will understand how information is encoded, processed, and manipulated to perform various operations. The course highlights the roles of processing units, memory subsystems, and input-output interfaces in delivering efficient performance. It emphasizes both the functional and operational aspects of computer hardware. Learners will explore performance enhancement techniques such as pipelining, parallelism, and advanced processor organization. The concepts of interrupts, memory access methods, and system interconnections are discussed to build a complete systems view. Practical insights into how hardware components interact in real-time operations are provided. By the end, students will have the knowledge to interpret, analyze, and optimize computer systems for different computing needs.

Course Pre/Co-requisites

A9402 – Digital Electronics

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9505.1. Demonstrate the architecture and operations of digital computers using register transfer language and micro-operations.

A9505.2. Apply foundational knowledge to analyze and execute basic computer operations, instruction processing, and data handling within a computer system.

A9505.3. Use knowledge of number representations and arithmetic algorithms to perform computations in digital systems.

A9505.4. Apply knowledge of input-output interfaces and memory hierarchy to efficiently manage data flow and storage in computer systems.

A9505.5. Examine the design and performance of pipelined and multiprocessor systems for efficient use of system resources.

Course Syllabus

Unit-I:

Introduction: Block diagram of Digital Computer, Von Neumann architecture, Harward architecture. Register Transfer Language and Micro operations - Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Unit-II:

Central Processing Unit: Instruction codes, Computer Registers, Computer instructions, Instruction cycle, Memory Reference Instructions, Register Reference Instructions, Input – Output Instructions. General Register Organization, Stack Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Interrupt types, interrupt cycle.

Unit-III:

Computer Arithmetic: Data Representation: Fixed Point Representation, Floating Point Representation. Addition and subtraction, multiplication Algorithms, Division Algorithms, floating – point arithmetic operations.

Unit-IV:

Input-Output and Memory Organization: Input-Output Interface, Modes of Transfer, Direct memory Access. Memory Hierarchy, Main Memory, RAM and ROM chips, Auxiliary memory, Cache Memory, Cache mapping techniques.

Unit-V:

Pipeline and Multi Processors: Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics, Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, and Vector Processing. Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Inter-processor arbitration and Cache Coherence.

Books and Materials

Text Books:

1. Mano, M. Morris. *Computer System Architecture.*, 3rd ed., Pearson Education, 1992.

Reference Books:

1. Hamacher, Carl, Zvonko Vranesic, Safwat Zaky, and Naraig Manjikian. *Computer Organization and Embedded Systems.*, 9th 6th ed., McGraw-Hill Education, 2012.
2. Stallings, William. *Computer Organization and Architecture: Designing for Performance.*, 32nd 10th ed., Pearson Education, 2016.
3. Tanenbaum, Andrew S., and Todd Austin. *Structured Computer Organization.* 6th ed., Pearson, 2013.

A9008 – Engineering Physics Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

The Engineering Physics laboratory course provides hands-on experience with fundamental concepts in semiconductors, magnetism, optics, and wave phenomena. Students will conduct practical experiments including the I-V characteristics of LEDs and solar cells, Hall effect measurements, and determination of energy band gaps. The course also covers the analysis of magnetic and dielectric properties, measurement of laser wavelength, characterization of optical fibre parameters, and AC frequency determination using a sonometer. These experiments are designed to enhance conceptual understanding and develop experimental skills relevant to modern physics and engineering applications.

Course Pre/Co-requisites

This course has no specific pre-requisites and co-requisites.

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 7: Affordable and Clean Energy
SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9008.1. Determine key parameters of LEDs and solar cells from their I–V characteristics.

A9008.2. Apply the Hall Effect to determine the type of semiconductor and estimate the density of majority charge carriers.

A9008.3. Evaluate material properties including energy band gap, magnetic moment, dielectric constant, and magnetic hysteresis behavior.

A9008.4. Apply the principles of lasers and optical fibres to determine laser wavelength and Numerical Aperture.

A9008.5. Apply principles of mechanical waves to determine AC supply frequency.

Course Syllabus

List of Experiments:

1. Determination of threshold voltage of LED from its I-V characteristics.
2. Study the I-V characteristics of Solar cell and find the fill factor.
3. Verification of the type of semiconductor material by estimating the density of majority carriers using the Hall Effect.
4. Determination of the energy band gap of a given semiconductor.
5. Determine the Magnetic moment of a given magnet and Horizontal component of earth's magnetic field.
6. Study of B-H curve of a ferromagnetic material.
7. Determination of dielectric constant of a given material.
8. Determination of the wavelength of a laser source using a plane transmission grating.

9. Evaluation of the numerical aperture (NA) and acceptance angle of a given optical fibre.
10. Determination of the frequency of an AC supply using a sonometer.

Laboratory Equipment/Software/Tools Required:

1. Light Emitting Diode Kit
2. Solar Cell Kit
3. Hall Effect Setup
4. Energy Gap of a Semiconductor Kit
5. Magnetic Moment Setup
6. B-H Curve Kit
7. Dielectric Constant Setup
8. Semiconductor Diode Laser
9. Plane Diffraction Grating
10. Optical Fibre Trainer Kit
11. Sonometer Setup
12. Meters – Ammeter, Voltmeter, Digital Multimeter, Deflecting Magnetometer, Thermometers

Books and Materials

Text Books:

1. Jain, Sushil Kumar, and Manjeet Singh *Applied Physics Experiments*, JBC Press, 2013.

Reference Books:

1. Mal, S. B., and Er. Ashish Jesuja *Practical Physics for Engineering Students of B.Tech*, JBC Press, 2020.

A9205 – Basic Electrical Engineering Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

The Basic Electrical Engineering Laboratory provides students with practical hands-on experience to deepen their understanding of electric circuits, DC and AC machines, and to foster the development of analytical skills. This course delves into the fundamental concepts and methodologies underpinning Electrical Engineering, covering various aspects such as electrical circuits, network theorems, and the principles and operating conditions of DC machines (motors and generators), Transformers, Induction motors, and Alternators. Through practical applications and experimentation, students engage in active learning to solidify their comprehension and proficiency in the field of Electrical Engineering.

Course Pre/Co-requisites

This course has no specific pre requisites or co-requisites.

Relevant Sustainable Development Goals (SDGs)

SDG 7: Affordable and Clean Energy

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9205.1. Analyze and demonstrate the application of Ohm's Law and Kirchhoff's Laws in DC and AC circuits.

A9205.2. Demonstrate practical application and evaluation skills by analyzing and verifying Superposition, Thevenin's and Norton's theorems in provided circuits.

A9205.3. Determine and verify impedance, voltage, and current in series RL and RC circuits using AC circuit concepts.

A9205.4. Analyze and interpret the torque-speed and performance characteristics of DC motors to evaluate their operational efficiency.

A9205.5. Analyze the open-circuit, short-circuit, and performance test results of transformers and AC machines to assess their key parameters and operational characteristics.

Course Syllabus

List of Experiments:

1. Verification of Ohm's Law.
2. Verification of KVL and KCL.
3. Verification of Super position theorem.
4. Verification of Thevenin's theorem.
5. Verification of Norton's theorem.
6. Calculations and Verification of Impedance, Voltage and Current of series RL and series RC circuits.
7. OC and SC Tests on Single Phase Transformer.
8. Measurement of transformation ratio of Single Phase Transformer.

9. Torque-Speed Characteristics of a DC Shunt Motor.
10. Torque-Speed Characteristics of 3-phase Induction Motor.
11. Performance Characteristics of a Separately Excited DC Motor.
12. No-Load Characteristics of a Three-phase Alternator.

Laboratory Equipment/Software/Tools Required:

1. Bread Boards, Resistors of different values, Regulated Power Supply.
2. 1-Phase Transformer, DC Machines, 3-phase Induction Motor, Alternator.
3. Voltmeter, Ammeter, Tachometer, Rheostats and Watt meters (LPF and UPF).

Books and Materials

Text Books:

1. Hayt, William H., Jack E. Kemmerly, and Steven M. Durbin. *Engineering Circuit Analysis.* 9th ed., McGraw-Hill Higher Education, 2007.
2. Theraja, B. L., and A. K. Theraja. *A Textbook of Electrical Technology.* Vols. 1–2, S. Chand Publishers, New Delhi.

Reference Books:

1. Kothari, D. P., and I. J. Nagrath. *Basic Electrical Engineering.* 3rd ed., Tata McGraw-Hill, 2010.
2. Kulshreshtha, D. C. *Basic Electrical Engineering.* McGraw-Hill, 2009.
3. Bobrow, L. S. *Fundamentals of Electrical Engineering.* Oxford University Press, 2011.

A9504 – Data Structures Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This course enables to design and implement efficient C++ programs. Students learn object-oriented concepts such as classes, objects, and dynamic memory. It covers data structures including arrays, linked lists, stacks, queues, trees, and graphs. Hands-on exercises emphasize insertion, deletion, traversal, and searching operations. Students explore algorithms to improve performance and optimize data handling. Practical work with STL containers, iterators, and templates is included. Hashing and dictionary implementations demonstrate efficient data retrieval techniques. The course develops analytical thinking and systematic problem-solving skills. Students gain experience in building reliable, maintainable, and scalable software. By the end, students can apply programming concepts to solve real-world computational problems.

Course Pre/Co-requisites

A9501 – Programming for Problem Solving

A9502 – Programming for Problem Solving Laboratory

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9504.1. Implement programs that efficiently manage and manipulate data using dynamic programming techniques in C++.

A9504.2. Apply various linked list techniques to perform insertion, deletion and traversal on given data.

A9504.3. Develop programs using linear data structures stack and queue to handle data processing tasks efficiently.

A9504.4. Implement nonlinear data structures to solve real time aplications.

A9504.5. Choose appropriate hashing and dictionary methods to efficiently store, retrieve, and manipulate data.

Course Syllabus

List of Experiments:

1. Implementing Classes, Objects, and Dynamic Memory Allocation
   a. Define a Student class with attributes: rollNumber, name, marks
   b. Implement a constructor to initialize objects and a destructor to display a message when an object is deleted
   c. Dynamically allocate an array of Student objects using new
   d. Input details of n students and display them
   e. Release allocated memory using delete

2. Function Overloading and Templates
    a. Implement two overloaded functions add() that can add:Two integers, Two floating-point numbers
    b. Define a function template swapValues() that swaps two variables of any type.
    c. Test swapValues() with integer, float, and string types
3. Using STL Containers and Iterators
    a. Create a vector of integers, insert elements, and display them using an iterator
    b. Create a list of strings, perform insertion and deletion, and traverse using an iterator
    c. Use a map to store StudentID -> Name pairs and display all elements
    d. Use a set to store unique integers and print them in sorted order
    e. Apply STL algorithms like sort(), find(), and count() on the containers
4. Implementing Singly Linked List (ADT) Operations
    a. Define a Node structure containing data and a next pointer Implement functions to:
    b. Insert a node at the beginning, end, and at a given position
    c. Delete a node from the beginning, end, and a specified position..
    d. Traverse the linked list and display all elements
    e. Search for an element in the list and return its position.
    f. Demonstrate all operations with sample inputs
5. Implementing Doubly Linked List (ADT) Operations
    a. Define a DoublyNode structure containing data, prev, and next pointers. Implement functions to:
    b. Insert a node at the beginning, end, and any position
    c. Delete a node from the beginning, end, and a specified position
    d. Traverse the list forward and backward
    e. Search for an element in the list.
    f. Demonstrate all operations with sample inputs
6. Circular Linked Lists (ADT) Operations
    a. Define a Node structure for circular singly linked lists with a next pointer pointing to the first node. Implement functions to:
    b. Insert a node at the beginning and end
    c. Delete a node from the beginning and end
    d. Traverse the list starting from any node and print all elements.
    e. Extend the above to circular doubly linked lists with prev and next pointers
    f. Demonstrate operations with sample inputs.
7. Implementing Stack (ADT) Using Array and Linked Lists Implement the following operations:
    a. push() – insert an element onto the stack
    b. pop() – remove the top element from the stack
    c. peek() – view the top element without removing it
    d. isEmpty() and isFull() – check stack status
8. Expression Conversion and Evaluation Using Stack
    a. Implement infix to postfix conversion using a stack
    b. Implement evaluation of postfix expressions using a stack
    c. Test with different arithmetic expressions (including parentheses)

9. Implementing Queues (ADT) Using Array and Linked Lists Implement a linear queue using:
    a. Array with enqueue() and dequeue() operations
    b. Linked list dynamically allocating nodes for each element
    c. Display or traverse list using array and linked list
10. Implementing Queues (ADT) Using Circular Linear List Implement a linear queue using:
    a. Array with enqueue() and dequeue() operations.
    b. Display or traverse list using array.
11. Binary Tree (ADT) Implementation and Traversals
    a. Define a Node structure with data, left, and right pointers. Implement functions to:
    b. Insert nodes into a binary tree
    c. Traverse the tree using:
    d. Inorder Traversal
    e. Preorder Traversal
    f. Postorder Traversal
    g. Demonstrate traversal operations with a sample binary tree
12. Binary Search Tree (BST) and AVL Tree Operations Implement BST operations:
    a. Insert a node
    b. Delete a node
    c. Search for a value
    d. Display the tree using Inorder traversal to verify correctness
13. AVL Tree Operations Implement BST operations:
    a. Insert a node
    b. Delete a node
    c. Search for a value
    d. Implement AVL tree insertion and deletion with rotations to maintain balance
    e. Display the tree using Inorder traversal to verify correctness
14. Hash Table Implementation and Collision Handling
    a. Implement a hash table using an array
    b. Design and apply a simple hash function
    c. Implement collision resolution techniques: Separate Chaining using linked lists Open Addressing: linear probing, quadratic probing, and double hashing
    d. Perform insertion, deletion, and searching operations.
    e. Demonstrate handling of collisions with sample inputs.
15. Dictionary Implementation Using Linear List and Skip List
    a. Implement a dictionary using a linear list: Perform insertion, deletion, and search operations.
    b. Implement a dictionary using a skip list for faster search: Include multiple levels with forward pointers. Implement insertion, deletion, and search.
    c. Compare the efficiency of linear list and skip list implementations with sample data order

Laboratory Equipment/Software/Tools Required:

1. Computer Systems (PCs) installed with Ubuntu OS (Open source/ Freeware)
2. GCC Compiler (Open source/ Freeware)

Books and Materials

Text Books:

1. Horowitz, Ellis, Sartaj Sahni, and Dinesh Mehta. *Fundamentals of Data Structures in C++*, 2nd ed., Schaum's Outlines, Universities Press, 2019.
2. Malik D.S.*Data Structures and Algorithms in C++.* , 5th ed., Course Technology, 2010.

Reference Books:

1. Drozdek, Adam. *Data Structures and Algorithms in C++.* , 5th ed., Cengage Learning, 2025.
2. Dale, Nell, Chip Weems, and Tim Richards. *C++ Plus Data Structures.*, 6th ed., Jones & Bartlett Learning, 2018.

A9302 – Engineering Workshop

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

The Engineering Workshop course is designed to introduce students to basic and advanced manufacturing processes, workshop trades, and hands-on practical skills essential for engineering practice. The course provides experiential learning on a variety of trade skills including fitting, carpentry, welding, foundry, plumbing, electrical house wiring, and fabrication techniques such as 3D Printing. Students will gain practical familiarity with common tools, machines, and manufacturing methods, along with safety and quality management practices.

Course Pre/Co-requisites

This course has no specific pre-requisites and co-requisites.

Relevant Sustainable Development Goals (SDG(s))

SDG 4: Quality Education
SDG 9: Industry, Innovation, and Infrastructure
SDG 12: Responsible Consumption and Production

Course Outcomes

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

A9302.1. Demonstrate the ability to perform fundamental workshop trades, including fitting, carpentry, welding, and plumbing, by completing a variety of hands-on tasks.

A9302.2. Demonstrate safe and effective usage of fabrication tools and digital equipment.

A9302.3. Identify and operate common workshop machines and tools while strictly adhering to safety protocols and quality management practices.

A9302.4. Recognize the properties of different materials and select appropriate tools and processes for specific manufacturing applications.

A9302.5. Fabricate a complete, functional assembly by integrating multiple skills learned across different workshop trades.

Course Syllabus

Part - A (Practical)

1. Fitting: L - Fit / V - Fit / Square - Fit / Semi Circular - Fit.
2. Carpentry: Lap Joint/ T- Bridle Joint / Mortise & Tenon Joint.
3. House wiring: Series / Parallel / One Bulb by One Switch / Tube Light / One Bulb by Two way Switch.
4. Welding: Butt Joint / Lap Joint / T Joint.
5. Foundry: Single Piece Pattern/ Split Piece Pattern / Multi Piece Pattern.
6. Tin Smithy: Open Scoop / Funnel / Rectangular Tray / Cylindrical.
7. Plumbing: Pipe Threading / Pipe Joints.
8. 3D Printing: Prepare a 3D Printing Model.

Part - B (Demonstration)

1. CNC Machining & Power Tools.
2. Casting & Plastic Moulding.
3. Welding (TIG/MIG, Gas Welding), Brazing.
4. Blacksmithy.

Laboratory Equipment/Software/Tools Required:

1. Fitting: Bench vise, Hacksaw frame, Calipers, Files, Try Square.
2. Carpentry: Carpentry vise, Chisels, Saws, Wooden Hammer, Try Square.
3. House Wiring: Voltage Tester, Wire Cutter, Wire Stripper, Cutting Plier, Nose Plier, Wire Gauge.
4. Welding: Welding M/c, Safeguards, Chipping Hammer, Electrode Holder.
5. Foundry: Wooden patterns, Riddle, Riser, Runner, Gate cutter, Rammers.
6. Tin Smithy: Wire Gauge, Snips, Pliers, Steel rule, Soldering kit, Spot Welding, Nylon Hammers.
7. Plumbing: Pipe Wrench, Pipe Cutter, Pliers, Pipe Die Set.
8. 3D Printing, 3D Modeling & Slicer Software.
9. Furnace, tongs, Swage Block.
10. Additional: Model Joints, Craft Knives and Electric Boards.

Books and Materials

Text Books:

1. Hajra Choudhury, S.K., and Nirjhar Rao. *Elements of Workshop Technology [Vol. 1, Manufacturing Processes]*, Revised and Enlarged 7th ed., Media Prompters & Publishers, 2023.
2. Singh, Devendra, et al. *Workshop Technology: Crafting Innovation for Engineering Students*, 1st ed., Redshine Publication, 2025.
3. Rosenberg, Neil. *Designing 3D Printers: Essential Knowledge*, 3rd ed., Independently published, 2023.

Reference Books:

1. Reddy, K. Venkata. *Workshop Practice Manual*, Reprint, 6th ed., BSP Books Private Ltd, 2025.
2. Gupta, Ram K. *3D Printing: Fundamentals to Emerging Applications*, 1st ed., CRC Press, 2024.
3. Devi, V. Lakshmi, and Kumar K. *Battery Technology Handbook: Classification, Control, and System Integration: Comprehensive Guide to EV Battery Design and Management Systems*, 1st ed., Notation Press, 2024.

A9022 - Product Design and Development

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	0	45	45	1	40	60	100

Course Description

Course Overview

This course equips students with a community-focused approach to product design, taking them from initial concept to a refined, practical solution. They will learn core design principles, understand the product development life cycle, and explore essential hardware and software tools through curated resources. Students will engage in prototyping, testing, and iterative refinement using feedback from community partners, ensuring sustainability and user-centered results. The course also develops their ability to document and communicate designs effectively, including preparing detailed specifications and user manuals.

Course Pre/Co-requisites

A9021 - Community Centered Design Thinking

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 9: Industry, Innovation, and Infrastructure
SDG 11: Sustainable Cities and Communities
SDG 12: Responsible Consumption and Production

Course Outcomes

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

A9022.1. Explain the principles of product design and the product development life cycle, with an emphasis on addressing real-world community needs

A9022.2. Generate and evaluate innovative product concepts using relevant Hardware and Software design tools

A9022.3. Develop functional prototypes using appropriate prototyping tools, and perform initial testing and validation

A9022.4. Refine prototypes through iterative feedback loops, integrating sustainability and user-centered design principles

A9022.5. Document and communicate product designs effectively with comprehensive specifications and user manuals tailored for community stakeholders

Course Syllabus

Unit-I:

Introduction to Product Design for Community Need: Understanding the principles and significance of product design, product development life cycle. Communicating design concepts to community partner. Refining designs based on feedback.

Unit-II:

Product Development Skills: Identify & Develop proficiency in using relevant Hardware & Software design tools. Equip with curated resources on tools essential for managing and scaling products effectively.

Unit-III:

Prototype & Testing: Introducing the concepts and purpose of prototyping. Creating functional prototypes to represent product designs using appropriate tools and techniques. Testing prototypes for performance, usability, and alignment with design goals.

Unit-IV:

Iterative Refinement: Refinement of prototypes based on community partner feedback and verification of product sustainability, with integration of user-centered design principles to align with community needs and expectations.

Unit-V:

Documentation and Communication Strategies: Documenting product designs with detailed specifications, Effective communication strategies for conveying designs to community partners, Preparation of user manuals and documentation for community partners.

Activity Plan

Week	Unit	Objective	Teaching Method	In-Class Activities	Assignments / Assessments	CO Mapping
1	Unit-I: Intro to Product Design	Understand product design principles & life cycle	Concept briefing	Case study analysis of successful community-based product	Short reflection: "Importance of product design for communities"	CO1
2	Unit-I: Intro to Product Design	Identify community needs & build empathy	Concept briefing + Fieldwork preparation	Practice empathy mapping, mock survey in class	Conduct 3–5 interviews/surveys with stakeholders	CO1, CO2
3	Unit-I: Intro to Product Design	Define problem statements	Guided teamwork + Brainstorm session	Develop "How Might We" questions, prioritize opportunities	Submit refined problem statement document	CO1, CO2
4	Unit-II: Product Development Skills	Generate diverse concepts	Creativity Hands-on Session (SCAMPER, Role-storming)	Group ideation, sketching concepts	Sketchbook submission (min. 10 ideas)	CO2
5	Unit-II: Product Development Skills	Apply digital design tools	Hands-on training + Peer support	Practice in Figma / SolidWorks / TinkerCAD	Submit wireframes / 3D sketches	CO2
6	Unit-II: Product Development Skills	Evaluate concepts systematically	Evaluation + User testing demo	Apply Pugh method to concepts, gather peer feedback	Submit evaluation matrix + selected final concept	CO2, CO3

Week	Unit	Objective	Teaching Method	In-Class Activities	Assignments / Assessments	CO Mapping
7	Unit-III: Prototype & Testing	Build low-fidelity prototype	Practical lab session + Peer feedback	Teams construct basic prototypes from cardboard/foam	Submit prototype photos + reflection	CO3
8	Unit-III: Prototype & Testing	Develop high-fidelity prototype	Prototype building - Hands-on Session	Create working model with core functionality	Submit tested prototype (video evidence optional)	CO3
9	Unit-IV: Iterative Refinement	Collect & apply user/community feedback	User feedback roundtable	Usability testing with peers/partners	Submit iteration log with design changes	CO4
10	Unit-IV: Iterative Refinement	Refine based on sustainability & ergonomics	Fine-tuning activity	Refine materials, safety, visual design, ergonomics	Submit refined prototype design brief	CO4
11	Unit-V: Documentation	Document & communicate design	Documentation + Visual design session	Create instruction guides, packaging design, visuals	Draft user manual (Canva/InDesign optional)	CO5
12	Unit-V: Communication Strategies	Present & reflect on outcomes	Final Showcase	Final presentations: video demos, posters, product showcase	Final report, user manual, and presentation	CO5

Books and Materials

Text Books:

1. Pavan Soni. *Design Your Thinking: The Mindsets, Toolsets and Skill Sets for Creative Problem-solving*, Penguin Random House India, 2024.
2. Anuja Agarwal. *Design Thinking: A Framework for Applying Design Thinking in Problem Solving*, Cengage India, 2024.

Reference Books:

1. Shalini Rahul Tiwari, Rohit Rajendra Swarup. *Design Thinking: A Comprehensive Textbook*, Wiley India, 2023.
2. Srinivasan R., Mohammed Ismail, Arulmozhi Srinivasan. *A Textbook on Design Thinking: Principles, Processes and Applications*, reprint, S. Chand Publishing, 2025.

II B.Tech. I Semester

A9014 - Business Economics and Financial Analysis

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course introduces the fundamentals of Business Economics and Financial Analysis, covering business structures, economic concepts, demand and supply analysis, production and cost, market structures, and pricing. It also focuses on accounting principles, preparation of financial statements, ratio analysis, and capital budgeting methods to support effective financial decision-making.

Course Pre/Co-requisites

This course has no specific prerequisite and co-requisite.

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 8: Decent Work and Economic Growth
SDG 9: Industry, Innovation, and Infrastructure
SDG 17: Partnerships for the Goals

Course Outcomes

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

A9014.1. Analyze business and economic concepts to assess their impact on the overall economic environment.

A9014.2. Examine the relationship between demand, supply, and elasticity in understanding market behavior.

A9014.3. Apply production, cost, market structure, and pricing concepts to interpret business operations and competitive strategies.

A9014.4. Apply accounting principles and rules for preparing financial statements.

A9014.5. Analyze financial statements and capital budgeting techniques to evaluate the financial health of a business.

Course Syllabus

Unit-I:

Introduction to Business and Economics: Business-Structure of Business Firm, Types of Business Entities, Limited Liability Companies, Sources of Capital for company – Conventional and Non-Conventional. Economics-Significance of Economics, Micro and Macro Economic Concepts. Business Economics- Nature and Scope, Role of Business Economist. National Income-Concepts and Importance, Inflation and Money Supply. Business Cycle -Features and Phases.

Unit-II:

Demand and Supply Analysis: Demand-Function, Determinants and types. Law of Demand-Assumption and Exceptions. Elasticity of Demand- Types, Measurement and Significance of Elasticity of Demand, Factors affecting Elasticity of Demand. Demand Forecasting- Methods of Demand Forecasting. Supply Analysis-Functions, Determinants and Law of Supply.

Unit-III:

Production, Cost, Market Structures & Pricing: Production Analysis- Factors of Production, Production Function, Production Function with one variable input, two variable inputs, Returns to Scale. Cost analysis: Types of Costs, Short run and Long run Costs Break Even Analysis (simple problems). Market Structure: Nature of Competition, Features of Perfect competition, Monopoly, Oligopoly, Monopolistic Competition. Pricing-Types of Pricing, Product Life Cycle based Pricing,

Unit-IV:

Financial Accounting: Accounting concepts and Conventions, Accounting Equation, Double Entry system of Accounting, Rules for maintaining Books of Accounts, Journal, Posting to Ledger, Preparation of Trial Balance, Elements of Financial Statements, Preparation of Final Accounts (Simple Problems).

Unit-V:

Ratios Analysis and Capital Budgeting: Concept of Ratio Analysis, Importance and Types of Ratios- Liquidity Ratios, Turnover Ratios, Profitability Ratios, Proprietary Ratios, Solvency, Leverage Ratios– Analysis and Interpretation. Capital Budgeting – Capital, Types of capital, Capital Budgeting Methods (Simple Problems).

Books and Materials

Text Books:

1. D. Chaturvedi, S. L. Gupta. *Business Economics Theory and Applications* 4th ed., Interna tional Book House Pvt. Ltd. 2013.
2. Dhanesh K Khatri. *Financial Accounting*, 3rd ed., Tata Mc–Graw Hill, 2011.

Reference Books:

1. Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury. *Managerial Economics*, 2nd ed., Tata Mc Graw Hill Education Pvt. Ltd. 2012.
2. A.R. Aryasri, *Managerial Economics and Financial Analysis*, 9th ed., TMH, India, 2011.
3. Geethika Ghosh, Piyali Gosh, Purba Roy Choudhury. *Managerial Economics*, 2nd ed., Tata Mc Graw Hill Education Pvt. Ltd. 2012.

A9506 – Discrete Mathematical Structures

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course will provide the mathematical fundamentals needed to understand computer applications. This course will begin by covering the mathematical concepts necessary in the study of propositional and predicate logic. Next, it covers the concepts of relations and ordering to study and construct the lattices. Further, it discusses the concepts of algebraic systems like semi groups and groups. Then move on to the recurrence relations, which help in writing efficient code. Finally, it covers the topics of graph theory to analyze complex structures using the concepts of planar, Euler graphs, and chromatic numbers.

Course Pre/Co-requisites

This course has no specific prerequisite or co-requisite.

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9506.1. Identify the importance of statements and predicate calculus in deriving valid inferences.

A9506.2. Use relations and ordering methods to identify the relationship among the elements in the system.

A9506.3. Select suitable algebraic systems to find solutions for real-time problems.

A9506.4. Apply the concepts of counting, inclusion, and exclusion principles to solve portioning problems in computer algorithms.

A9506.5. Make use of graph concepts and computing methods to solve complex problems, and also examine the recurrence relations to improve the code efficiency.

Course Syllabus

Unit-I:

Propositional Logic: Statements and Notation, Connectives, Well-formed formulas, Tautologies, Equivalence of formulas and Tautological implications, Normal Forms, Rules of Inference, Consistency of Premises, Direct and Indirect method of proof, Predicates, the statement functions, Variables and Quantifiers (Single and Multiple), Free and Bound Variables.

Unit-II:

Relations and Ordering: Basics of Relations, Relation Matrix and Digraphs, Properties of Binary Relations on a Set, Equivalence Relations, Transitive closure (Warshall's algorithm), Partial Ordering Relations, Hasse diagrams. Lattices as Partially Ordered Sets, Definitions and Examples, Properties of Lattices, Some Special Lattices.

Unit-III:

Algebraic Structures: Algebraic Systems: Definitions and Examples, Simple algebraic systems and General properties. Semigroups and Monoids: Definitions and Examples. Groups and Subgroups: Definitions and Examples, Order of an element, Cyclic group: Definition and Examples.

Unit-IV:

Elementary Combinatorics: Basics of Counting, Combinations, and Permutations with Repetitions and Constrained Repetitions, Binomial Coefficients, The Binomial and Multinomial Theorems (Without Proofs), The Principle of Inclusion-Exclusion (for two and three-sets), Pigeonhole principle, The Generalized Pigeonhole Principle.

Unit-V:

Recurrence Relations and Graph Theory: Solving Recurrence Relations by Substitution, The method of characteristic Roots, Solutions of Linear Homogeneous and Inhomogeneous recurrence relations of second and third order. Basic Concepts of Graphs, Adjacency Matrix, Isomorphism and Subgraphs, Planar Graphs, Euler Circuits, Hamiltonian graphs, Graph Coloring, Chromatic Numbers, Trees and their properties, Spanning trees (BFS and DFS algorithms).

Books and Materials

Text Books:

1. J. P. Trembly, R. Manohar., *Discrete Mathematical Structures with Applications to Computer Science*, 43rd Tata McGraw Hill, India,2008.
2. Joe L. Mott, Abraham Kandel, Theodore P. Baker. *Discrete Mathematics for Computer Scientists and Mathematicians*, 3rd 2nd Edition, Prentice Hall of India Learning Private Limited, New Delhi, India,2009.

Reference Books:

1. Kenneth H. Rosen., *Discrete Mathematics and its Applications*, 9th 7th Edition, Tata McGraw Hill, India,2017.
2. Ralph P. Grimaldi. *Discrete and Combinatorial Mathematics - an applied introduction*, 32nd Pearson Education, 5th edition.

A9507 – Operating Systems

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course provides a thorough understanding of how operating systems function as a vital component of computing systems. It begins by exploring the services of an operating system in managing hardware and software resources. Students will learn how systems handle multiple tasks and processes efficiently, ensuring smooth execution and coordination. The course also examines how multiple processes communicate and synchronize to avoid conflicts and ensure consistency. Techniques for managing memory effectively, including strategies for allocation and optimization, are explored. Learners gain insight into how files and storage devices are organized, accessed, and maintained by the system. Efficient storage and retrieval methods, along with space management, are discussed in practical contexts. The challenges of resource contention and how systems deal with issues like deadlocks are also analyzed. By the end, students will understand how an operating system ensures reliability, performance, and user interaction.

Course Pre/Co-requisites

A9501 – Programming for Problem Solving

A9505 – Computer Organization

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 8: Decent Work and Economic Growth

SDG 9: Industry, Innovation, and Infrastructure

SDG 11: Sustainable Cities and Communities

Course Outcomes

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

A9507.1. Demonstrate various services provided by operating system for both users and system components.

A9507.2. Examine the core concepts of process management in coordinating the concurrent execution of multiple processes.

A9507.3. Utilize memory management techniques to optimize the efficient utilization of main memory.

A9507.4. Choose appropriate file and disk management techniques to ensure efficient data storage and quick access.

A9507.5. Employ a suitable deadlock handling method to manage the allocation of resources among processes.

Course Syllabus

Unit-I:

Operating Systems Overview and Process Management: Definition, Operating System Types, Operating System operations, Operating system services, System calls and System Programs. Process concepts-Process, Process State Diagram, PCB and Operations on processes, Process Scheduling- Scheduling Criteria, Scheduler Types and Scheduling Algorithms.

Unit-II:

Process Synchronization: Inter Procee Communication- Pipes, Message Passing and Shared Memory. Concept of Synchronization, Critical section problem, Peterson's solution, Semaphores, Classic problems of Synchronization- The Bounded Buffer Problem, The Readers –Writers Problem, Dining - Philosophers Problem.

Unit-III:

Memory Management: Introduction to Memory Management, Swapping, Contiguous Memory Allocation, paging, segmentation, virtual memory, demand paging, Page-replacement algorithms, allocation of frames, thrashing.

Unit-IV:

File and Disk Management: Concept of a file – File Attributes, File Types, Access Methods, Directory Structures, File System Implementation, Directory Implementation, File Allocation methods, and Free-Space management. Introduction to Magnetic Disks, Disk Structures, Disk Scheduling, Swap Space Management.

Unit-V:

Deadlocks: System Model, Deadlock Characterization-Necessary Conditions, Resource Allocation Graph, Deadlock Prevention, Deadlock Avoidance - RAG Algorithm, Banker's Algorithm, Detection- Single Instance of a Resource type, Multiple Instances of a resource type, recovery from deadlock.

Books and Materials

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. *Operating System Concepts* , 9th Edition, Wiley India Private Limited, New Delhi,2019.

Reference Books:

1. Stallings, William. *Operating Systems: Internals and Design Principles.* 9th ed., Pearson Education, 2017.
2. Das, Sumitabha. *Your UNIX: The Ultimate Guide.* 3 rd ed., McGraw-Hill, 2013.
3. Chan, Terrence. *UNIX System Programming Using C++.* 1st ed., Prentice Hall PTR, 1997.

A9509 – Database Management System

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course introduces the core principles and techniques required in the design and implementation of database systems. This course focus on relational database management systems, including database design theory: E-R modeling, query languages like relational algebra, relational calculus and SQL. It also covers essential DBMS concepts such as: Normalization, Transaction Processing, Concurrency Control, Recovery, and tree based indexing techniques like ISAM, B+ trees etc which are required for designing an effective database. Students can undertake a semester project to design, build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS .

Course Pre/Co-requisites

A9506 – Discrete Mathematical Structures

A9601 – Object Oriented Programming

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 8: Decent Work and Economic Growth

SDG 11: Sustainable Cities and Communities

SDG 16: Peace, Justice, and Strong Institutions

Course Outcomes

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

A9509.1. Design a database for a given problem using E-R diagrams and Relational Model.

A9509.2. Make use of relational algebra and SQL to construct queries for solving case study-based database problems.

A9509.3. Employ normalization techniques to organize database structures efficiently and eliminate data redundancy.

A9509.4. Select appropriate transaction control techniques to maintain database consistency.

A9509.5. Examine various concurrency control, recovery, and indexing techniques for improving performance.

Course Syllabus

Unit-I:

Introduction and Data Base Design: Introduction to DBMS, applications of DBMS, database systems versus file systems, view of data, Database users and administrators, database system structure. Database Design: Introduction to ER model, entities, attributes and entity sets, relationships and relationship sets, specialization, generalization, ER model sailor's database. Logical database design: E-R to relational, Relational algebra operations. Introduction to Relational database model, SQL data types and database languages.

Unit-II:

SQL Programming: SQL basic operators, SQL set operators-union, intersect and except operators, Integrity constraints in SQL. Aggregate operators, GROUP BY, ORDER BY and HAVING Clause, Writing queries for sailor's database. Null values, views in SQL, nested queries, SQL joins-inner join, outer join, left outer join, right outer join, storing and retrieving images, storing and retrieving files, PL/SQL basics.

Unit-III:

Schema Refinement and Normal Forms: Introduction to schema refinement & Normalization, Decomposition and properties of decompositions, functional dependencies, Closure of Attributes set. Normal forms: 1NF, 2NF, 3NF, BCNF, 4NF,5NF. Problems on normalization, Schema refinement in database design.

Unit-IV:

Transaction Management: Transaction concept, transaction states, ACID properties, schedules-serial and non-serial schedules, Serializability-Conflict serializability, View serializability, recoverability and types of recoverability. Triggers, Cursors, SQL stored procedures.

Unit-V:

Concurrency control, Recovery, and Indexing: Concurrency control- lock based protocols, timestamp-based protocols, deadlocks handling. Recovery-ARIES recovery algorithm, Log based recovery. File organization techniques, Tree index structures: ISAM and B+ trees.

Books and Materials

Text Books:

1. Raghurama Krishnan, Johannes Gehrke. *Database Management Systems*, 3rd ed., Tata McGraw-Hill, 2014.
2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan. *Database System Concepts*, 7th ed., McGraw- Hill, 2019.

Reference Books:

1. Elmasri Navate *Fundamentals of Database Systems*, 7th ed., Pearson Education, 2016.
2. Swamynathan C. J. Date, A. Kannan and S *An Introduction to Database Systems*, 8th ed., Pearson Education India, 2015.

A9601 – Object Oriented Programming through Java

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course offers a comprehensive understanding of OOP concepts using Java. It emphasizes various aspect of core Java Environment suitable to write efficient, maintainable and portable code for real world applications. The course builds a strong foundation in OOP principles, packages, interfaces, exception handling and multithreading. It also provides in depth knowledge to implement Collection framework to manage and organize the data efficiently. The course introduces GUI programming using Swings and event handling mechanisms to create interactive applications. It further covers JDBC concepts to connect Java applications with databases and perform CRUD operations using both console and GUI interfaces

Course Pre/Co-requisites

A9501 – Programming for Problem Solving

A9503 – Data structures

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9601.1. Make use of OOP principles and programming constructs to build console-based real-time applications.

A9601.2. Examine the applications for Exception Handling and Multithreading.

A9601.3. Implement Collection Framework and Streams to organize and manipulate data efficiently.

A9601.4. Build GUI applications using Swings and event handling.

A9601.5. Analyze and implement JDBC concepts to perform database operations using console and GUI-based applications.

Course Syllabus

Unit-I:

Introduction to OOP and Java: Need for OOP paradigm-Difference between Procedural and Object-Oriented Programming, OOP Concepts, Java Buzz Words, Arrays, Encapsulation-Class, Objects, Methods and Constructors, this keyword, static keyword, Overloading-Methods and Constructors. Inheritance Basics, super keyword, Inheritance types, Method Overriding, Dynamic Method Dispatch, Abstract classes, final keyword. Defining and implementing interfaces, Extending interfaces. Defining a Package, Finding Packages and Class path, Access Protection and Importing Packages.

Unit-II:

Exception Handling and Multithreading: Exception-Handling Fundamentals, Exception Types, Using try catch, throw throws and finally keywords, Built-in Exceptions, Creating own exception subclasses. Multithreading: Life cycle of a thread, creating threads, thread priorities, Synchronizing threads, Inter thread Communication. String class and StringBuffer.

Unit-III:

Collections Framework and Streams: Collections Hierarchy, Collection classes-ArrayList, LinkedList, HashSet, TreeSet, HashMap and TreeMap, StringTokenizer, Arrays class. Streams Hierarchy, File Streams FileInputStream and FileOutputStream, FileReader and FileWriter, Performing read, write operations on Files and Serialization.

Unit-IV:

GUI Programming with Java and Event Handling: Delegation Event Model, Event Sources, Event Classes,Event Listener Interfaces, Handling Mouse and Keyboard Events, Layout Managers-FlowLayout, BorderLayout, GridLayout and CardLayout. Swings: Introduction to swing, swings Vs AWT, hierarchy for swing components. JFrame, JPanel, JComponent, JLabel, JTextField, JTabbedPane, Swing Buttons, JScrollPane, JComboBox, JTable.

Unit-V:

JDBC: Architecture, Types of JDBC Drivers, JDBC Components: DriverManager, Connection, Statement, Prepared Statement, Callable Statement ResultSet, Establishing Connection with Data Base, Statement, PreparedStatement and Callable Statement. Performing CRUD operations (Create, Retrieve, Update and Delete) using console, Connecting and retrieving data using with GUI (Swing).

Books and Materials

Text Books:

1. Herbert Schildt, and Danny Coeard. *Java: The Complete Reference*, 13 th Tata McGraw-Hill Education, 2024.

Reference Books:

1. Y. Daniel Liang, *Introduction to Java Programming-Comprehensive Version*, 10th Pearson Education, 2018.
2. Kathy Sierra, Bert Bates *OCA Java SE 8 Programmer*, 1st reprint, McGraw-Hill Education, 2017

A9508 – Operating Systems Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This course provides hands-on experience with UNIX/Linux operating system commands, shell scripting, and system programming. Students will practice fundamental file and process management commands and explore inter-process communication (IPC) mechanisms such as pipes, FIFOs, shared memory, message queues, and semaphores. The course also includes simulation of essential commands and implementation of system-level programming using UNIX I/O system calls. Additionally, students will develop and analyze memory management techniques, page replacement policies, disk scheduling algorithms, and deadlock avoidance strategies. Through practical programming exercises, the course aims to strengthen problem-solving skills and deepen understanding of operating system concepts and internal mechanisms.

Course Pre/Co-requisites

A9501 - Programming for Problem Solving

A9502 - Programming for Problem Solving Laboratory

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 7: Affordable and Clean Energy

SDG 8: Decent Work and Economic Growth

SDG 11: Sustainable Cities and Communities

Course Outcomes

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

A9508.1. Utilize Unix utilities and system calls to implement and manage processes, memory, and file systems effectively in a UNIX/Linux environment.

A9508.2. Apply inter-process communication (IPC) mechanism to establish and manage communication among processes.

A9508.3. Employ synchronization using semaphores to coordinate concurrent tasks in operating systems.

A9508.4. Develop shell scripts and simulate file handling operations to access and manage data from disk storage efficiently.

A9508.5. Implement memory management techniques, disk scheduling algorithms, and deadlock handling mechanisms to demonstrate key operating system services.

Course Syllabus

List of Experiments:

1. Practice commands – like `mkdir`, `rmdir`, `cat`, `nl`, `ls`, `cp`, `mv`, `rm`, `man`.
2. Practice commands – like `wc`, `uniq`, `comm`, `cmp`, `diff`, `ln`, `unlink`, `chmod`, `du`, `df`.
3. Practice commands – `head`, `tail`, `sort`, `grep`, `egrep`, `fgrep`, `cut`, `paste`, `join`.
4. Process Management System calls `fork()`, `exec()` and `wait()`.
5. a. Two-way Communication using Pipes.

b. Process Communication using FIFOs.

6. Implement Shared Memory form of IPC.
7. Implement Message Queue form of IPC.
8. Implement Semaphore form of IPC to implement Producer Consumer Problem.
9. Simulate `cp`, `head` and `tail` commands.
10. Shell Script programs using Conditional and Iterative statements.
11. Write C programs to simulate the following memory management techniques
    a. Paging
    b. Segmentation
12. Program to implement FIFO and LRU Page replacement algorithms.
13. Program to implement FCFS and SSTF Disk Scheduling algorithm.
14. Program to implement Banker's algorithm for Deadlock avoidance.

Laboratory Equipment/Software/Tools Required:

1. Computer Systems (PCs) installed with Ubuntu OS (Open source/ Freeware)
2. GCC Compiler (Open source/ Freeware)

Books and Materials

Text Books:

1. Sumitabha Das. *Your Unix: The Ultimate Guide*, Tata McGraw-Hill, New Delhi, India.
2. T. Chan. *Unix System Programming using C++*, PHI, India, 1996.

Reference Books:

1. Randal K. Michael. *Mastering Unix Shell Scripting*, 2nd Edition, Wiley, India, 2008.

A9510 – Database Management System Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This practical course introduces the fundamental principles and techniques of database system design and implementation. It emphasizes relational database management systems (RDBMS) and covers key concepts like E-R modeling and SQL. Students will gain hands-on experience with SQL commands, including joins, aggregate functions, and nested queries. The course also introduces PL/SQL features such as triggers, cursors, procedures, and functions for advanced data handling. Learners will practice creating and querying databases effectively using structured programming techniques. Through case studies and exercises, students develop skills in writing efficient and optimized database queries. The course encourages analytical thinking through query formulation and problem-solving in real-world scenarios. Python integration with databases will also be introduced for data access and manipulation. Students will undertake a mini-project to design and implement a functional database system. By the end of the course, students will demonstrate competence in modeling, designing, and managing relational databases.

Course Pre/Co-requisites

A9501 - Programming for Problem Solving

A9503 - Data Structures

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 8: Decent Work and Economic Growth

SDG 11: Sustainable Cities and Communities

SDG 16: Peace, Justice, and Strong Institutions

Course Outcomes

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

A9510.1. Develop a database by creating E-R diagrams and transforming them into a relational model for a given problem.

A9510.2. Formulate and execute SQL queries to retrieve, modify, and manage data for a given case study.

A9510.3. Utilize PL/SQL programming constructs, cursors and triggers to interact with databases effectively.

A9510.4 Implement stored procedures and functions in PL/SQL to develop reusable solutions for database problems.

A9510.5 Build applications that interact with relational databases to handle real-time data operations.

Course Syllabus

List of Experiments:

1. a. Practice on SQL data definition language (DDL) commands and data manipulation language (DML) commands to retrieve and modify data.
   b. Practice on SQL DCL and TCL commands.
2. a. Case Study on designing ER diagrams for university database.
   b. Case Study on designing ER diagrams for company database to store information about employees.

3. Practice on different types of SQL operators and aggregate operations.
4. a. Practice on queries using `GROUP BY`, `ORDER BY`, and `HAVING` clauses.
   b. Practice on queries involving different types of joins.
5. a. Case Study on sailors database queries.
   b. Case Study on employee database queries.
6. Practice on different types of SQL built-in functions like date functions, string functions, numeric and conversion functions, analytic functions, storing and retrieving images, files.
7. Practice on queries using co-related sub-queries and nested queries.
8. Practice on PL/SQL basics for writing programs using programming constructs like variables, operators, conditional and control statements.
9. Practice on PL/SQL programs using cursors.
10. Practice on PL/SQL programs using triggers.
11. Practice on PL/SQL programs using stored procedures.
12. Practice on PL/SQL programs using functions.
13. Practice on performing basic database operations by connecting to database using Python.
14. Case Study in developing a database following all steps in the design of databases elaborating normalization and de-normalization.

Laboratory Equipment/Software/Tools Required:

1. Computer Systems (PCs) installed with Ubuntu OS (Open source/ Freeware)
2. MySql (Open Source/ Freeware)
3. Python and Python IDE (Open Source/ Freeware)

Books and Materials

Text Books:

1. Raghurama Krishnan, Johannes Gehrke. *Database Management Systems*, 3rd Edition, Tata McGraw-Hill, New Delhi, India, 2014.
2. Abraham Silberschatz, Henry F. Korth, S. Sudarshan. *Database System Concepts*, 7th Edition, McGraw-Hill, New Delhi, India, 2019.

Reference Books:

1. Elmasri Navate. *Fundamentals of Database Systems*, 7th Edition, Pearson Education, India, 2016.

A9602 - Object Oriented Programming through Java Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This course provides a comprehensive coverage of theory and practice of OOP concepts using Java. The course focuses on different aspect of core Java Environment suitable to write efficient, maintainable, and portable code for real world application. It provides strong foundation on OOP Principles, Packages, and Interfaces and also illustrates Exception Handling and Multithreaded mechanisms. The course provides in depth knowledge to implement Collection framework. Emphasis on Swing concepts used for GUI applications is given with event handling. It also covers the implementation of JDBC to connect Java applications with databases. The course plays a vital role in developing front-end interface for Mini and Major Projects

Course Pre/Co-requisites

A9502 - Programming for Problem Solving Laboratory

A9503 - Data structures

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9602.1. Make use of various constructs to write a console application.

A9602.2. Use principles of OOP to develop real time applications.

A9602.3. Examine the applications for Exception Handling and Multithreading.

A9602.4. Implement Collection Framework to organize data efficiently.

A9602.5. Build GUI applications using Swings connecting JDBC.

Course Syllabus

List of Experiments:

1. a. Read the marks of a student in 4 subjects and find grade.
   b. Define a class Rectangle with data members length and width. Write methods to find perimeter and area of a rectangle (class and object).
   c. Create a class Account with data members name, acno and balance. Use appropriate methods to perform various operations like deposit, withdraw, and balance check.
2. a. Create a class Student with appropriate data and methods using constructor.
   b. Create overloaded methods to find volume of Sphere, Cylinder and Cone.
3. a. Read two matrices of size $m \times n$, $p \times q$, perform the multiplication of matrices.
   b. Declare a class called Employee having employee-id and employee-name as members. Extend class Employee to have a sub class called Salary having designation and monthly salary as members. Define the appropriate constructors and methods, use `this` and `super` keywords, display the employees drawing salary more than 30000.

4. Write a Java program that creates an abstract base class Shape with two members base and height, a member function for initialization and a method to compute shapeArea(). Derive two specific classes Triangle and Rectangle which override the method shapeArea(). Write a driver class (main) to display the area of the triangle and rectangle (Use `super` keyword).
5. Create a Package `Measure`; in which store a class named `Convertor` that contains methods to convert mm to cm, cm to m and m to km. Define a class `NeedConvertor` that imports the `Convertor` class, now store `NeedConvertor` outside the package `Measure`.
6. a. Read two integers as strings Num1 and Num2 to perform division. The program throws a NumberFormatException if Num1 or Num2 cannot be converted to integers, and if Num2 is Zero throw an ArithmeticException. Display the exception message.
   b. In the Custom Exception Test class, the age is expected to be a positive number. It would throw the user defined exception NegativeAgeException if the age is assigned a negative number.
7. a. Create a multithreaded java program by creating a subclass of Thread and then creating, initializing, and starting two Thread objects from your class. The threads will execute concurrently and display "Java is object oriented" in console window.
   b. Write a program to implement thread synchronization using synchronized block and synchronized method.
8. Use an ArrayList to manage Employee objects for insertion, display and remove.
9. a. Implement MouseListener and MouseMotionListener to handle various mouse events.
   b. Implement KeyListener to handle key events.
10. Create a Simple login window to validate a user with name and password.
11. Create a user interface to insert employee details, Display the data in TextArea.
12. Create a JTable to display various fields of Student data like RollNo, Name, Branch, Year, Percentage etc.
13. JDBC Program to create, retrieve, insert, delete and update student data.
14. JDBC program to create, retrieve, insert, delete and update student data using Prepared Statement.

Laboratory Equipment/Software/Tools Required:

1. Computer Systems (PCs) installed with Ubuntu OS (Open source/ Freeware
2. JDK (Open Source/Freeware)

Books and Materials

Text Books:

1. Herbert Schildt, and Danny Coeard. *Java: The Complete Reference*, 13 th Tata McGraw-Hill Education, 2024.

Reference Books:

1. Y. Daniel Liang, *Introduction to Java Programming-Comprehensive Version*, 10th Pearson Education, 2018.
2. Kathy Sierra, Bert Bates *OCA Java SE 8 Programmer*, 1st reprint, McGraw-Hill Education, 2017

A9511 – Python Programming Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This Python programming lab is designed to develop strong problem-solving and coding skills through hands-on practice. It starts with basic concepts like installation, simple output, and input handling to build a clear understanding of the language. Students learn to work with variables, operators, and expressions to perform computations and make logical decisions using conditional statements. Iteration techniques are practiced to solve problems like reversing numbers, factorials, and Fibonacci series. The lab strengthens skills in string handling, including slicing, searching, and transformations. Core Python data structures such as lists, dictionaries, tuples, and sets are explored to manage and process data effectively. Emphasis is placed on writing reusable code through user-defined functions, recursive calls, and lambda expressions. Students work with Python modules to handle tasks like date, time, and calendars. Error handling is introduced to ensure safe and reliable programs. The course concludes with file operations for creating, reading, appending, and summarizing data, preparing students for real-world applications.

Course Pre/Co-requisites

A9501 - Programming for Problem Solving

A9502 - Programming for Problem Solving Laboratory

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 8: Decent Work and Economic Growth

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9511.1. Interpret basic programming constructs and control statements to solve simple computational problems

A9511.2. Apply string operations and regular expressions to process and manipulate text.

A9511.3. Make use of data structures lists, tuples, sets, and dictionaries to store and organize data effectively.

A9511.4 Implement modular programming concepts using functions, modules, and file handling techniques.

A9511.5 Develop applications using exception handling constructs to handle runtime errors in software applications.

Course Syllabus

Theory:

1. **Python Fundamentals and Control Statements:** Introduction to Python, Features of Python, Identifiers, Reserved Words, Data Types, Variables and Constants, Input / Output Statements, Conditional Statements – if, if-else, if-elif-else; Iterative Statements – for, while; Jump / Transfer Statements – break, continue, pass.

2. **Strings and Regular Expressions:** Strings: String Definition, Indexing, Slicing, Multiline Strings, Escape Sequences, String Formatting, Mathematical Operations on Strings, Checking Membership, Comparison, String Manipulation Techniques, String Immutability, Built-in String Functions and Methods. Regular Expressions: Pattern Matching, Search, Replace, Match Objects, Grouping.
3. **Functions, Modules:** Functions: Introduction, Function Definition, Function Call, Types of Arguments, Return Statement, Recursive Functions, Anonymous Functions (Lambda). Modules: Importing, Creating and Using Modules, Standard Libraries.
4. **Exception Handling and File Handling:** Exception Handling – Errors in Python, types of exceptions, try, except, else, finally blocks, raise keyword, custom exception classes. File Handling – Introduction, File Types, Opening and Closing Files, Reading and Writing in Text and Binary Files, File Modes and Methods (read(), readline(), write(), etc.), File Path Operations, Copy and Merge.

List of Experiments:

1. Introduction to Python Lab: Installation and Simple Output Display.
   a. Write a python program to print your Name, Roll Number, and Branch.
   b. Write a python program to read a string "Python Programming" and display it on the screen.
   c. Write a python program to read integer, float & string values and display them on the screen.
2. Programs using Input Output Statements, Variables and Expressions.
   a. Write a python program to read a float value and convert Fahrenheit to Centigrade.
   b. Write a python program to find the area of triangle.
   c. Write a python program to read the marks in 5 subjects and display the average.
3. Programs using various operators in Python.
   a. Write a python program for demonstrating the usage of comparison operators.
   b. Write a python program to swap / interchange two numbers.
   c. Write a python program for demonstrating the usage of unary, shift, logical, membership and identity operators.
4. Programs using Conditional Statements.
   a. Write a python program to check a given number is Even or Odd.
   b. Write a python program to find the greatest of 3 integer numbers.
   c. Write a python program to demonstrate nested if statement.
5. Programs using Iterative Statements.
   a. Write a Python program to reverse the digits of a given number.
   b. Write a Python program to find the factorial of a given number.
   c. Write a python program to display factors of a given integer number.
   d. Write a python program to print Fibonacci numbers.
   e. Write a python program to display all prime numbers between 0 to n.
6. Programs using Strings and its Operations.
   a. Write a program that asks the user to enter a string and perform the following:
      i. Total number of characters in the string.
      ii. Repeat the string 10 times.

iii. The first character of the string.
iv. The first three characters of the string.
v. The last three characters of the string.
vi. The string in backwards.
vii. The seventh character of the string if it exists, otherwise display a message "Not exist".
viii. The string with its first and last characters removed.
ix. The string into capital case.
x. The string with every letter replaced by a space.

b. Write a python program to demonstrate string concatenation, repetition, and membership testing.
c. Write a python program to demonstrate built-in string functions: upper(), lower(), strip(), find().
d. Write a Python program to count the number of vowels in a given string.

7. Programs using Python Data Structures (Lists).
   i. Print the total number of items in the list.
   ii. Print the last item in the list.
   iii. Print the list in reverse order.
   iv. Print Yes if the list contains a 5 and No otherwise.
   v. Print the number of occurrences of an element in the list.
   vi. Remove the first and last items from the list and sort the remaining items.
   vii. Print how many integers in the list are less than a given value.
   viii. Print the average of the elements in the list.
   ix. Print the largest and smallest value in the list.
   x. Program which applies list comprehensions to generate even numbers.
8. Programs using Python Data Structures (Dictionary).
   a. Write a python program for demonstrating the creation of dictionary, accessing dictionary elements, modifying dictionary elements, finding length and possible operations.
   b. Write a python program to create a dictionary of students with keys as roll numbers and values as names. Perform operations like insert, update and modify student data.
   c. Write a Python program with the following requirements: Create a dictionary that stores 10 usernames as keys and their corresponding passwords as values. Prompt the user to enter a username and a password. Check the entered credentials and:
   (i) If the username does not exist, display: "Not a valid user."
   (ii) If the username exists but the password is incorrect, display: "Invalid password."
   (iii) If both are correct, display: "Welcome".
9. Programs using Python Data Structures (Tuples and Sets).
   a. Write a python program to demonstrate various operations on tuples.
   b. Write a python program to demonstrate various operations on sets.
10. Programs using User Defined Functions.
   a. Define a lambda function to compute square and cube.
   b. Write a python program to find factorial of a given number using function.
   c. Write a python program to find factorial of a given number using recursive function.

11. Programs using Modules.
    a. Write a Python program to display the date and time using the time module.
    b. Write a Python program that prints the calendar of a particular month.
    c. Write a Python program to check whether a given year is a leap year or not using the calendar module.
12. Programs using Exception Handling.
    a. Write a Python program to read two integers from the user and perform division, handling ZeroDivisionError and ValueError using try-except-finally.
    b. Write a Python program that asks for user input and converts it to an integer, with exception handling for invalid data.
13. Programs on File Handling.
    a. Write a Python program to read a text file, handling File Not Found Error.
    b. Write a Python program to count the number of lines, words, and characters, and then write the summary of a file to a new output file with exception handling.
    c. Create a text file named `student_data.txt`.
    d. Accept student details (roll number, name, marks) from the user and write them to the file.
    e. Read and display the contents of the file.
    f. Append new records to the same file and display the updated contents.

Laboratory Equipment/Software/Tools Required:

1. Computer Systems (PCs) installed with Ubuntu OS (Open source/ Freeware)
2. Jupyter notebook or Pycharm IDE, Python Run Time System

Books and Materials

Text Books:

1. Reema Thareja. *Python Programming using Problem solving Approach*, 3rd Edition Oxford University Press, New Delhi India, 2017.
2. Gowrishankar S., Veena A.. *Introduction to Python Programming*, 1st Edition, CRC Press, Boca Raton, Florida, USA, 2018..

Reference Books:

1. Charles R. Severance.. *Python for Everybody: Exploring Data Using Python 3*, 1st Edition, , CreateSpace Independent Publishing Platform, United States, 2016.
2. Timothy A Budd. *Exploring Python*, 8th Edition, Tata McGraw Hill, New Delhi, 2019.
3. Allen B. Downey. *Think Python: How to Think Like a Computer Scientist*, 2nd Edition, O'Reilly Media, California, USA, 2016

A9802 - Data Analysis and Visualization Using Power BI

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

The Data Analysis and Visualization using Power BI Lab introduces students to the powerful business intelligence and visualization capabilities of Microsoft Power BI. The lab focuses on connecting to diverse data sources, cleaning and transforming data using Power Query, and modeling relationships between datasets. Students will gain hands-on experience in building reports and dashboards with a variety of visualization techniques, ranging from basic charts to advanced visuals such as maps, KPIs, decomposition trees, and AI-powered insights. The lab also covers Data Analysis Expressions (DAX) for calculated measures and time intelligence, as well as publishing, sharing, and maintaining reports on the Power BI Service. By the end of the course, students will be able to design interactive, data-driven dashboards that communicate insights effectively for decision-making in real-world business scenarios.

Course Pre/Co-requisites

This course has no specific pre-requisites and co-requisites.

Relevant Sustainable Development Goals (SDGs)

SDG 3: Good Health and Well-being
SDG 4: Quality Education
SDG 8: Decent Work and Economic Growth
SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9802.1 Install and operate WEKA software to load, explore, and preprocess datasets using various interfaces.

A9802.2 Apply data preprocessing techniques such as attribute selection, normalization, discretization, and outlier removal.

A9802.3 Implement and evaluate classification algorithms including Decision Tree, Naïve Bayes, Rule-based, and k-NN.

A9802.4 Perform association rule mining using Apriori and FP-Growth algorithms and interpret the results.

A9802.5 Apply clustering algorithms such as K-Means, Hierarchical, and Density-based to group data and interpret cluster outcomes.

Course Syllabus

Theory:

1. **Introduction to Data Visualization & Power BI:** Importance of data visualization in decision making, Principles of effective visualization (clarity, accuracy, storytelling), Overview of Business Intelligence (BI) and Self-Service BI, Introduction to Microsoft Power BI: Components, Architecture, and Workflow, Installing and exploring Power BI Desktop interface.
2. **Data Import, Cleaning, and Modeling:** Connecting to various data sources (Excel, CSV, databases, online sources), Power Query for data shaping: filtering, merging, transformations, Creating relationships between tables, Data modeling basics: Star schema, hierarchies, calculated columns, and measures.

3. **Basic and Advanced Visualizations:** Bar, Column, Line, Pie, and Donut charts, Hierarchies and drill-downs, Tree maps, Funnel charts, and Waterfall charts, Custom visuals from AppSource, Formatting visuals (titles, legends, data labels, tooltips).
4. **Interactive Dashboards and Advanced Features:** Slicers, filters, and cross-highlighting, Bookmarks and storytelling with reports, Time series visualization and forecasting, Geographic visualization: Maps, Filled maps, and ArcGIS maps, Key Performance Indicators (KPIs) and Cards.
5. **Report Design, Export, and Best Practices:** Designing professional dashboards, Applying themes, layouts, and color consistency, Exporting reports (PDF, PowerPoint, Excel), Sharing .pbix files (offline collaboration), Best practices in visualization design and dashboard performance optimization, Case study: Building an end-to-end dashboard using Power BI.

List of Experiments:

1. Introduction to Power BI Environment
   a. Power BI Desktop vs. Power BI Service
   b. Connecting to data sources (Excel, CSV, Databases)
   c. Overview of Power Query Editor
2. Data Import and Basic Transformations
   a. Loading multiple datasets
   b. Data type conversions
   c. Removing duplicates and missing values
3. Data Cleaning and Shaping
   a. Splitting and merging columns
   b. Filtering and sorting
   c. Creating calculated columns
4. Data Modeling Basics
   a. Understanding tables and relationships
   b. Creating and managing one-to-many relationships
   c. Star schema vs. snowflake schema
5. Introduction to DAX (Data Analysis Expressions)
   a. Creating measures and calculated fields
   b. Using basic DAX functions (SUM, AVERAGE, COUNT)
6. Advanced DAX for Analytics
   a. Time intelligence functions (YTD, MTD, QTD)
   b. Conditional measures with IF, SWITCH
7. Basic Visualizations
   a. Bar charts, Column charts, Pie charts
   b. Line and area charts
   c. Scatter plots with formatting
8. Advanced Visualizations
   a. Tree maps, Funnel charts, Waterfall charts

   b. KPIs and card visuals
   c. Slicers and filters
9. Geographic Data Visualization
   a. Maps, Filled maps, ArcGIS maps
   b. Latitude/longitude-based plotting
10. Hierarchies and Drill Features
   a. Creating hierarchies (Year → Month → Day)
   b. Drill-down and drill-through functionality
   c. Bookmarks for navigation
11. Dashboards and Report Design
   a. Designing interactive dashboards
   b. Report formatting and theming
12. AI Features in Power BI
   a. Using Q&A visuals (natural language)
   b. Decomposition tree
   c. Key Influencers visual
13. Exporting and Sharing Reports
   a. Exporting reports to PDF, PowerPoint, and Excel
   b. Saving and distributing .pbix files
   c. Printing and offline sharing
14. Performance Optimization and Best Practices
   a. Optimizing data models (query folding, star schema)
   b. Reducing file size and improving refresh times
   c. Best practices in visualization and report design

Laboratory Equipment/Software/Tools Required:

1. Computer System with Ubuntu Operating System.
2. Microsoft Power BI Desktop
3. Microsoft Excel (2016 or later)
4. Web Browser (Edge/Chrome/Firefox)

Books and Materials

Text Books:

1. Powell, Brett, et al. *Learning Path Microsoft Power BI Complete Reference.* 1st ed., Packt Publisher, 2018.

Reference Books:

1. Arnold, Jeremey. *Learning Microsoft Power BI: Transforming Data Into Insights.* 1st ed., Packt Publisher, 2022.

A9023 - Technology Entrepreneurship

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	0	45	45	1	40	60	100

Course Description

Course Overview

This course enables students to transform refined product designs into viable entrepreneurial ventures or patentable innovations. Building on skills from previous courses in design thinking and product development, students will explore opportunity identification, intellectual property protection, market research, sustainable business models, funding strategies, and go-to-market planning. Emphasis is placed on aligning innovations with community needs while preparing for startup creation, patent filing, or both.

Course Pre/Co-requisites

A9022 - Product Design and Development

textbfRelevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 8: Decent Work and Economic Growth

SDG 9: Industry, Innovation, and Infrastructure

SDG 17: Partnerships for the Goals

Course Outcomes

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

A9023.1. Identify and analyze market opportunities for community-driven technological innovations.

A9023.2. Apply intellectual property strategies for protecting product designs and innovations.

A9023.3. Develop sustainable and scalable business models for product commercialization.

A9023.4. Formulate funding, financial, and go-to-market strategies for product launch.

A9023.5. Prepare and deliver investor-ready pitches or patent documentation to relevant stakeholders.

Course Syllabus

Unit-I:

Entrepreneurial Mindset and Opportunity Identification: Understanding technology entrepreneurship in the community context. Startup ecosystem and innovation pathways. Market analysis and opportunity mapping for commercialization of Product.

Unit-II:

Intellectual Property and Innovation Protection: Overview of IP: patents, trademarks, copyrights, and trade secrets; patent search, drafting, filing, and grant procedures; leveraging IP for competitive advantage and innovation scaling.

Unit-III:

Market Research and Business Model Development: Defining target markets, customer segments, and value propositions. Competitive analysis and differentiation strategies. Business Model Canvas and Lean Startup principles.

Unit-IV:

Funding, Financial Planning and Sustainability: Study of funding options including grants, angel investors, venture capital, and crowd funding; budgeting, forecasting, and financial planning for startups; and integration of sustainability into long-term business growth strategies.

Unit-V:

Go-to-Market Strategy, Pitching and Documentation: Branding, marketing, and distribution planning; creating persuasive pitches for investors, partners, and stakeholders; preparing necessary documentation; final presentation of patent draft or startup business plan to an expert panel.

Activity Plan

Week	Unit	Objective	Teaching Method	In-Class Activities	Assignments / Assessments	CO Mapping
1	Unit-I: Entrepreneurial Mindset	Understand technology entrepreneurship & startup ecosystem	Concept briefing + Open forum	Discussion: examples of community-based startups	Reflection note: "Why entrepreneurship matters for communities"	CO1
2	Unit-I: Opportunity Identification	Identify & map opportunities	Hands on session + Case analysis	Opportunity mapping using local problems	Opportunity mapping chart submission	CO1
3	Unit-II: IP Basics	Learn types of IP (patents, trademarks, copyrights)	Concept briefing + Example-driven discussion	Analyze famous patents & trademarks	Short report: "One innovation and its IP protection strategy"	CO2
4	Unit-II: Patent Process	Apply patent search & filing basics	Demo session + Hands-on exercise	Perform a mock patent search online (guided)	Draft simple patent claim for a product	CO2
5	Unit-III: Market Research	Define target market & customer segments	Concept briefing + Team activity	Build customer personas for chosen product idea	Submit customer persona & value proposition canvas	CO3
6	Unit-III: Business Model Development	Apply BMC & Lean Startup	Business modeling – Hands on session	Teams fill out Business Model Canvas	Submit BMC with initial differentiation strategy	CO3
7	Unit-IV: Funding Sources	Understand startup funding landscape	Concept briefing + Case discussion	Funding source comparison (VC, grants, crowdfunding)	Assignment: Funding strategy document for idea	CO4

Week	Unit	Objective	Teaching Method	In-Class Activities	Assignments / Assessments	CO Mapping
8	Unit-IV: Financial Planning	Apply budgeting & forecasting	Hands on session	Build basic revenue/cost projection table	Submit 1-year financial projection	CO4
9	Unit-IV: Sustainability	Integrate sustainability in startups	Group reflection	Apply sustainability checklist to business model	Submit revised BMC integrating sustainability	CO4
10	Unit-V: Go-to-Market	Learn branding & marketing strategies	Hands on session + Peer feedback	Draft a marketing plan with target channels	Submit draft marketing & distribution strategy	CO4, CO5
11	Unit-V: Pitching Skills	Develop persuasive pitch	Startup pitch drill	Students deliver 3-min practice pitches with feedback	Submit pitch deck draft	CO5
12	Unit-V: Final Showcase	Present final startup plan/patent draft	Showcase + Expert review	Final presentations to panel (faculty/guests)	Final project submission: startup plan or patent draft	CO5

Books and Materials

Text Books:

1. Deependra Sharma. *Entrepreneurship in India* , Routledge, 2023.
2. Dr. S. Glory Swarupa & Ms. Swapna Vanamala. *Innovation, Incubation and Intellectual Property Rights*, 2023.

Reference Books:

1. Neck, Heidi M., Patricia G. Greene, and Candida G. Brush. *Teaching Entrepreneurship: A Practice-Based Approach*, Edward Elgar Publishing, 2014.
2. Drucker, Peter F. *Innovation and Entrepreneurship: Practice and Principles*, reprint, Harper & Row, 1985.

II B.Tech. II Semester

A9005 – Probability Distributions and Applied Statistics

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		PI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	15	0	60	120	4	40	60	100

Course Description

Course Overview

This course provides an undergraduate foundation in both probability distributions and mathematical statistics and at the same time provides an indication of the relevance and importance of the theory in solving practical problems in the field of multidisciplinary engineering applications. The mathematical skills sustained from this course form a suitable base to analytical and theoretical concepts encountered in engineering profession.

Course Pre/Co-requisites

This course has no specific prerequisite and co-requisite.

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

Course Outcomes

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

A9005.1. Identify an appropriate probability distribution for a given discrete or continuous random variable.

A9005.2. Make use of probability distributions to analyze and solve a given problem.

A9005.3. Interpret correlation coefficient, perform regression analysis to fit the best curve and make inferences about population parameters based on sample data.

A9005.4. Inspect scientific hypothesis and estimate confidence intervals at different levels.

A9005.5. Compute P-value of a test statistic using component of hypothesis test.

Course Syllabus

Unit-I:

Random Variables: Concept of a Random Variable, Discrete probability Distributions, Continuous Probability Distributions. Mean and Variance of Random Variables.

Unit-II:

Discrete and Continuous Distributions: Discrete Distributions: Binomial Distribution, Poisson Distribution, Continuous Distributions: Uniform Distribution, Normal Distribution, areas under the Normal Curve, applications of the Normal Distribution.

Unit-III:

Correlation and Regression: Scatter diagram, Positive and Negative correlation, limits for coefficient of Correlation, Karl Pearson's coefficient of correlation, Spearman's Rank correlation, Regression Analysis: Concept, two lines of regression, Properties of regression coefficients. **Fundamental Sampling Distributions:** Random Sampling, some Important Statistics, Sampling Distributions, Sampling Distribution of means and the Central Limit Theorem, t- Distribution, F-Distribution.

Unit-IV:

Estimation and Testing of Hypothesis for Large samples: Point estimation, Maximum error estimate, Interval Estimation, Introduction to Hypothesis, Level of significance, one tailed and two tailed test, Test concerning one mean and one proportion, Two means and two Proportions.

Unit-V:

Testing of Hypothesis for Small samples: Test for single mean, difference of means and paired t-test, Test for ratio of variances (F-test), Chi-square test for goodness of fit and independence of attributes.

Books and Materials

Text Books:

1. Gupta, S.C. and Kapoor, V. K. *Fundamentals of Mathematical statistics* , 10^{th} ed., S Chand & Sons, New Delhi, 2000.
2. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye *Probability and Statistics for Engineers & Scientists* , 9^{th} ed., Pearson Publishers, 2016.

Reference Books:

1. Grewal, B.S. *Higher Engineering Mathematics* , 43^{rd} ed., Khanna Publications, 2015.
2. T.T. Soong *Fundamentals of Probability and Statistics for Engineers* , John Wiley & Sons, 2004.
3. Miller and Freund's *Probability and Statistics for Engineers* , 8^{th} ed., Pearson Educations, 2011.

A9513 – Computer Networks

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course builds a solid understanding of computer networks and their working principles. It introduces how devices and systems connect and communicate using wired and wireless methods. Students learn how data is transmitted, verified, and corrected to maintain accuracy and reliability. The course covers techniques for sharing network resources and managing access among devices. It explains routing, addressing, and congestion control to ensure smooth data delivery. Learners study how networks are structured and standardized for global communication. Transport mechanisms and flow control are discussed to achieve reliable end-to-end communication. Common tools and services are introduced to show how applications work over networks. Key protocols for web, email, file transfer, and remote access are explored in detail. By completion, students can analyze, design, and troubleshoot modern networking systems effectively.

Course Pre/Co-requisites

A9505 – Computer Organization

A9507 – Operating Systems

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

SDG 11: Sustainable Cities and Communities

Course Outcomes

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

A9513.1. Demonstrate the reference models and physical transmission methods used to establish and support network communication.

A9513.2. Select appropriate access control and error handling techniques to ensure reliable communication at the data link layer.

A9513.3. Examine different routing and congestion control algorithms used to manage efficient packet transmission in networks.

A9513.4. Employ transport layer protocols to establish and manage reliable end-to-end communication with efficient flow and congestion control.

A9513.5. Implement an appropriate application layer protocol to enable effective user communication and data exchange over networks.

Course Syllabus

Unit-I:

Introduction: Network hardware, Network software, Reference models: OSI, TCP/IP, Connection oriented network and connectionless network. The Physical Layer: Guided transmission media, twisted pairs, coaxial cable, fiber optics, Wireless Transmission.

Unit-II:

The Data Link Layer: Design issues, error detection and correction, elementary data link protocols, sliding window protocols. The Medium Access Sub layer: Channel allocations problem, multiple access protocols: ALOHA, CSMA, Collision free protocols, Ethernet, Data Link Layer switching.

Unit-III:

The Network Layer: Network layer design issues, Routing Algorithms: Shortest path routing, flooding, distance vector routing, link state routing. Congestion control algorithms, the network layer in the internet: IPv4, Sub-netting, Super-netting, CIDR, NAT and IPv6.

Unit-IV:

The Transport Layer: Transport service, Transport layer protocols: UDP and TCP, Introduction, The TCP service model, The TCP protocol, The TCP Segment Header, TCP connection establishment, connection release, TCP sliding window, TCP Timer management, TCP Congestion control, and Performance issues.

Unit-V:

The Application Layer: Domain name system- DNS Name Space, Domain Resource Records, Name Servers. Application Layer Protocols: Simple Network Management Proto- col (SNMP), Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Telnet.4

Books and Materials

Text Books:

1. Tanenbaum, Andrew S., Nick Feamster, and David J. Wetherall *Computer Networks* , 6th ed., Pearson, 2024.
2. Sidhu, Bhavneet *An Integrated Approach to Computer Networks*, 1st ed., Khanna Publishing House, 2019.

Reference Books:

1. Stallings, William *Cryptography and Network Security: Principles and Practice*, 8th ed., Pearson, 2020.
2. Forouzan, Behrouz A. *Data Communications and Networking with TCP/IP Protocol Suite*, 6th reprint, McGraw Hill, 2022.

A9604 - Web Application Engineering

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course offers a comprehensive introduction to scripting languages and their practical use in full-stack web development. It equips students with the skills to build responsive front-end interfaces using JavaScript and React.js, and scalable back-end services using Node.js, Express.js, MongoDB, MySQL, and Spring Boot. Starting with modern JavaScript concepts and runtime environments like Node.js, the course explores client-side development through component-based architectures in React. It then advances to RESTful API development, database integration using both SQL (MySQL) and NoSQL (MongoDB) systems, and concludes with enterprise-grade application development using the Spring Boot framework. The course emphasizes hands-on learning through lab exercises and projects that simulate real-world development tasks, ensuring students gain both conceptual clarity and technical proficiency across the stack.

Course Pre/Co-requisites

A9601 - Object Oriented Programming Through Java

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 8: Decent Work and Economic Growth
SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9604.1. Apply modern JavaScript features (ES6), and use Node.js to build interactive web applications.

A9604.2. Develop modular and reusable user interfaces using React.js, including props, state, and JSX.

A9604.3. Build scalable single-page applications using advanced React features such as Hooks, Routing, and Redux.

A9604.4. Create RESTful APIs using Node.js with Express and perform database operations with MySQL and MongoDB.

A9604.5. Design and implement RESTful backend services using Spring Boot integrated with relational and NoSQL databases.

Course Syllabus

Unit-I:

JavaScript Essentials and Node.js Environment: JavaScript syntax: variables, arrays, functions, events ES6 features: Arrow functions, let/const, spread/rest, destructuring. Introduction to Node.js: Node.js architecture and use cases, Units and Packages, Working with npm, REPL and file handling. Introduction to Express.js: Middleware and basic routing, RESTful API with GET and POST.

Unit-II:

React.js Fundamentals and Component Development: Introduction to React and JSX, Virtual DOM and rendering, React components: Functional and Class-based, Props and State, Lifecycle methods.

Unit-III:

Advanced React – Hooks, Routing and Redux: Handling events and forms in React, Controlled and uncontrolled components, useState, useEffect, useContext, React Router for multi-page navigation, Redux: store, reducers, actions, Introduction to Material UI.

Unit-IV:

Backend Development with Node.js and Databases: RESTful API development using Express.js, Connecting Node.js with MySQL: Creating database and tables, Performing CRUD operations using SQL, Introduction to MongoDB: Collections, documents, CRUD operations, MongoDB Compass GUI, Using Mongoose ORM with Node.js.

Unit-V:

Backend API Development using Spring Boot: Introduction to Spring Boot and Spring Initializer, Project structure and configuration (application. Properties), Spring Annotations: @RestController, @RequestMapping, @Autowired, REST API creation with Spring Boot. Integration with MySQL and MongoDB: JPA with MySQL using JpaRepository, MongoDB with MongoRepository, API testing with Postman, Exception handling and validation.

Books and Materials

Text Books:

1. Wandschneider, Marc. *Learning Node.js: A Hands-On Guide to Building Web Applications in JavaScript*, 2nd ed., Addison-Wesley Professional, 2016.
2. Sidelnikov, Greg. *React.js Book: Learning React JavaScript Library from Scratch*, independently published, 2017.
3. Walls, Craig.*Spring Boot in Action*, Manning Publications, 2016.
4. Bierer, Doug. *MongoDB 4 Quick Start Guide: Learn the Skills You Need to Work with the World's Most Popular NoSQL Database*, Packt Publishing, Sept. 2018.

Reference Books:

1. Brown, Ethan *Web Development with Node and Express: Leveraging the JavaScript Stack*, 2nd ed., O'Reilly Media, 2019.
2. Wieruch, Robin. *The Road to React: Your Journey to Master Plain Yet Pragmatic React.js*, Independently Published, updated 2022.

A9515 – Software Engineering

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course offers a comprehensive understanding of software engineering principles, focusing on the evolution and importance of software in the modern world. It introduces foundational process frameworks and methodologies to guide software development effectively. Students will explore how to gather, analyze, and document user and system requirements. Emphasis is placed on structured approaches to software design and the use of modeling techniques for visualizing system architecture. The course also delves into key concepts of software testing to ensure product functionality and reliability. It highlights the importance of measurement and metrics to evaluate both process efficiency and software quality. Strategies for managing risks and ensuring consistent software quality are also covered. Students will gain insights into quality assurance practices and industry standards. By the end, learners will be equipped with essential tools and techniques for building reliable, maintainable, and high-quality software systems. The course aims to bridge theoretical concepts with practical application in real-world software development.

Course Pre/Co-requisites

This course has no specific Pre/co-requisites

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 9: Industry, Innovation, and Infrastructure
SDG 12: Responsible Consumption and Production
SDG 17: Partnerships for the Goal

Course Outcomes

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

A9515.1. Apply software engineering principles and appropriate process models to structure software development activities for real-world applications.

A9515.2. Examine software requirements and the requirements engineering process to ensure accurate, complete, and feasible specifications for software development.

A9515.3. Make use of software design principles and UML modeling techniques to develop effective design solutions for software systems.

A9515.4. Choose appropriate software testing techniques and quality metrics to ensure the reliability of software systems.

A9515.5. Utilize risk management strategies and quality assurance practices to ensure software systems with quality standards.

Course Syllabus

Unit-I:

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. A Generic view of process: Software engineering- a layered technology, a process framework. Process models: The waterfall model, Spiral model, Frame work: Agile methodology.

Unit-II:

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

Unit-III:

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, class diagrams, sequence diagrams, use case diagrams, State Machine diagram, and Deployment diagram.

Unit-IV:

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. Metrics for Process and Products: Software measurement, metrics for software quality.

Unit-V:

Risk and Quality management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. Quality Management: Quality, Achieving Software Quality, Formal Technical Reviews, Elements of Software Quality Assurance, software reliability, the ISO 9000 quality standards.

Books and Materials

Text Books:

1. Pressman, Roger S., and Bruce R. Maxim *Software Engineering: A Practitioner's Approach* , 9th ed., McGraw-Hill Education, 2020.
2. Sommerville, Ian *Software Engineering* , 10th ed., Pearson, 2015.
3. Booch, Grady, James Rumbaugh, and Ivar Jacobson *The Unified Modeling Language User Guide*, 2nd ed., Addison-Wesley Professional, 2017.

Reference Books:

1. Peters, James F., and Witold Pedrycz *Software Engineering: An Engineering Approach*, Wiley India Pvt. Ltd, 2007.
2. Jawadekar, Waman *Software Engineering: Principles and Practice* , , Tata McGraw-Hill Education, 2004.
3. Page-Jones, Meilir. *Fundamentals of Object-Oriented Design in UML* , Addison-Wesley Professional, 2000.

A9516 – Design and Analysis of Algorithms

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
45	0	0	45	90	3	40	60	100

Course Description

Course Overview

This course will provide students with a strong foundation in developing efficient algorithmic solutions using strategies such as divide and conquer greedy methods, dynamic programming, backtracking, and branch and bound. It emphasizes both theoretical understanding and practical implementation, enabling students to analyse time and space complexities, solve real-world computational problems, and compare algorithmic approaches. The course also introduces advanced topics like NP-Hard and NP-Complete problems, equipping learners with the ability to identify computational boundaries and choose appropriate algorithmic techniques for various applications.

Course Pre/Co-requisites

A9501 – Programming for Problem Solving

A9503 – Data Structures

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 8: Decent Work and Economic Growth

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9516.1. Demonstrate the asymptotic notations, divide and conquer techniques to decompose complex problems into small and simple problems.

A9516.2. Utilize Greedy method to find out feasible solutions of problems..

A9516.3. Examine complex engineering problems to identify optimal solutions through dynamic programming.

A9516.4. Make use of backtracking and branching methods to solve the problems by verifying all possibilities of solutions.

A9516.5. Inspect nondeterministic algorithms to solve polynomial and non-polynomial problems..

Course Syllabus

Unit-I:

Introduction, Divide and Conquer: Algorithm definition, Pseudo code Specifications, Performance Analysis-Space Complexity, Time Complexity, Recurrence relations: Substitution, recursion tree and master theorem, Asymptotic Notations-Big-Oh, Omega, and Theta. Divide And Conquer-General Method, Finding Maximum and Minimum, Merge Sort, Quick sort, Strassen's Matrix Multiplication.

Unit-II:

Disjoint Sets and Greedy Method: Disjoint Set operations, Union and find algorithms. Greedy General Method, Real Knapsack Problem, Job sequencing with deadlines, Minimum-cost spanning trees- Prim's Algorithm and Kruskal's algorithm, Single source shortest Path.

Unit-III:

Dynamic Programming: General method, All pairs shortest path, Matrix Chain Multiplication, Optimal Binary search trees, 0/1 Knapsack, the travelling salesman problem, Reliability Design.

Unit-IV:

Back Tracking: The General Method, The n-Queens Problem, Sum of subsets, Graph coloring, Hamiltonian cycles, Knapsack Problem.

Unit-V:

Branch and Bound, NP-Hard and NP Complete Problems: General method, applications - Travelling sales person problem, 0/1 knapsack problem LC Branch and Bound solution, FIFO Branch and Bound solution. NP-Hard and NP-Complete Problems – Basic concepts, Non-deterministic algorithms, NP-Hard and NP Complete Classes.

Books and Materials

Text Books:

1. Horowitz, Ellis, Sartaj Sahni, and Sanguthevar Rajasekaran *Fundamentals of Computer Algorithms* , 2nd ed., Universities Press, New Delhi, 2008.
2. Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms.* , 4th ed., MIT Press, 2022.

Reference Books:

1. Lee, R. C. T., S. S. Tseng, R. C. Chang, and Y. T. Tsai . *Introduction to the Design and Analysis of Algorithms: A Strategic Approach* , McGraw-Hill Education India, 2005.
2. Weiss, Mark Allen. *Data Structures and Algorithm Analysis in C++*, 4th reprint, Pearson, 2013.

A9514 – Computer Networks Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This networking lab provides practical exposure to building and analyzing computer networks. It begins with connecting systems in a local network and implementing framing techniques like bit and character stuffing. Students practice error detection using CRC and flow control with sliding window and Go-Back-N protocols. Routing is explored through distance vector algorithms to generate routing tables. Security concepts are introduced with encryption and decryption techniques. Traffic management and congestion control are covered using leaky bucket and frame sorting methods.NS2 simulations include designing topologies, TCP/UDP communication, wireless networks, and performance analysis.The lab concludes with network design using loop constraints and Packet Tracer for real-world scenarios.

Course Pre/Co-requisites

A9501 – Programming for Problem Solving

A9502 – Programming for Problem Solving Laboratory

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

SDG 11: Sustainable Cities and Communities

Course Outcomes

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

A9514.1. Apply framing, error detection, and flow control techniques to ensure reliable data transmission across networks.

A9514.2. Implement routing algorithms and congestion control mechanisms for managing network traffic.

A9514.3. Choose an appropriate cryptographic algorithm to develop secure network applications.

A9514.4. Use simulation tools like NS2 and Packet Tracer to design, analyse, and optimize wired and wireless network topologies.

A9514.5. Develop client-server communication and protocol-based solutions (TCP/UDP) to model and evaluate real-world networking scenarios.

Course Syllabus

List of Experiments:

1. Connect Computers in Local Area Network.
2. Implement:
   a. Bit-Stuffing
   b. Character Stuffing
3. Implement CRC and Error handling techniques
4. Implement Sliding-Window protocols
5. Implement distance vector routing algorithm for obtaining routing tables at each node

6. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism
7. Implement data encryption and data decryption
8. Write a program for:
   a. congestion control using Leaky bucket algorithm
   b. frame sorting techniques used in buffers
9. Design client-server model using NS2
10. Design a sample topology using NS2
11. Implement Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) between pair of systems using NS2.
12. Design wireless network using NS2
    i. Simulate to Find the Number of Packets Dropped
    ii. Simulate to Find the Number of Packets Dropped by TCP/UDP
    iii. Simulate to Find the Number of Packets Dropped due to Congestion
    iv. Simulate to Compare Data Rate & Throughput
    v. Simulate to Plot Congestion for Different Source/Destination
    vi. Simulate to Determine the Performance with respect to Transmission of Packets
13. Design a network using loop constraint
14. Design a Network using Packet Tracer

Laboratory Equipment/Software/Tools Required:

1. A computer Systems installed with Ubuntu perating System (Open Source/Freeware).
2. NS2 (Open Source/Freeware).
3. Packet Tracer.

Books and Materials

Text Books:

1. Tanenbaum, Andrew S., Nick Feamster, and David J. Wetherall. *Computer Networks*, 6th ed., Pearson, 2024.
2. Sidhu, Bhavneet. *An Integrated Approach to Computer Networks*, 1st ed, Khanna Publishing House, 2019.

Reference Books:

1. Forouzan, Behrouz A. *Data Communications and Networking with TCP/IP Protocol Suite*, 6th ed., McGraw-Hill, 2022.
2. Nayak, Ajit Kumar, Satyananda Champati Rai, and Rajib Mall*Computer Network Simulation Using NS2.*, 1st ed., CRC Press, 2016.

A9605 - Web Application Engineering Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This course provides a solid foundation in scripting languages for full-stack web development. Students learn to build dynamic front-end interfaces using JavaScript and React.js.It covers back-end development with Node.js, Express.js, and databases like MongoDB and MySQL.Spring Boot is introduced for developing enterprise-level applications. Emphasis is placed on practical learning through labs and real-world projects. The course ensures both conceptual understanding and technical competence across the stack.

Course Pre/Co-requisites

A9602 - Object Oriented Programming Through Java Laboratory

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 8: Decent Work and Economic Growth
SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9605.1. Explain how JavaScript, Node.js, and modern ECMAScript features are applied to implement client-side interactivity and server-side file processing.

A9605.2. Design and build single-page applications (SPAs) using React, managing state, routing, and asynchronous data fetching.

A9605.3. Implement advanced state management and UI styling techniques using Redux and Material UI for scalable application development.

A9605.4. Apply and integrate REST APIs using Express.js and Spring Boot with data storage in MySQL and MongoDB.

A9605.5. Develop interactive client-side applications and server-side file processing using JavaScript and Node.js.

Course Syllabus

List of Experiments:

1. E-Commerce Cart System: Build an interactive shopping cart that allows users to add/remove items, update quantities, and see real-time price calculations. Use modern JavaScript features like arrow functions and array methods.
2. Dynamic Form Validation: Create a user registration form that validates inputs in real-time and simulates checking username availability against a mock API using async/await.
3. Server Log Analyzer: Develop a Node.js application that processes server log files, extracts error messages, and generates a summary report using core modules like fs and path.
4. Task Management App: Implement a Todo application where users can add, complete, and delete tasks. Use React state to manage the task list and props for component communication.

5. Weather Dashboard: Build a weather application that fetches data from an API and displays current conditions. Compare class and functional component implementations.
6. Job Application Form: Create a multi-step job application form with field validation, conditional rendering, and form submission handling using React hooks.
7. Theme Switcher: Develop a theme switching functionality (light/dark mode) that persists user preference across sessions using Context API.
8. Blog Platform: Build a blog website with multiple pages (Home, Articles, About) using React Router, including nested routes for article categories.
9. E-Commerce Filters: Enhance a product listing page with filter functionality managed by Redux and styled with Material UI components.
10. Bookstore API: Create a RESTful API for a bookstore with endpoints to manage books (CRUD operations) using Express.js and middleware.
11. User Management System: Develop a system to handle user registration and profiles with MySQL database integration and password hashing.
12. Blog with MongoDB: Implement a blogging platform where users can create posts and comments using MongoDB for data storage.
13. Employee Directory: Build a Spring Boot API for managing employee records with proper REST endpoints and JSON responses.
14. Inventory System: Create a product inventory management system with Spring Data JPA integration for either MySQL or MongoDB4. Books and Materials.

Laboratory Equipment/Software/Tools Required:

1. Computer Systems (PCs) installed with Ubuntu OS (Open source/ Freeware).
2. JDK (Open Source/Freeware), Visual Studio Code Editor.
3. Front-End & Back-End Tools: React, Redux, Material UI, and React Router for front-end; Node.js + Express.js, MySQL, MongoDB, and Spring Boot for server-side and database operations.
4. Supporting & Testing Tools: CURL for API testing, Git/GitHub for version control.

Books and Materials

Text Books:

1. Wandschneider, Marc. *Learning Node.js: A Hands-On Guide to Building Web Applications in JavaScript*, 2nd ed., Addison-Wesley Professional, 2016.
2. Sidelnikov, Greg. *React.js Book: Learning React JavaScript Library from Scratch*, independently published, 2017.
3. Walls, Craig.*Spring Boot in Action*, Manning Publications, 2016.
4. Bierer, Doug. *MongoDB 4 Quick Start Guide: Learn the Skills You Need to Work with the World's Most Popular NoSQL Database*, Packt Publishing, Sept. 2018.

Reference Books:

1. Brown, Ethan *Web Development with Node and Express: Leveraging the JavaScript Stack*, 2nd ed., O'Reilly Media, 2019.
2. Wieruch, Robin. *The Road to React: Your Journey to Master Plain Yet Pragmatic React.js*, Independently Published, updated 2022.

A9606 - IoT Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

The purpose of this course is to explore the students with various sensors, Microcontrollers (Aurdino) and Microprocessors (Raspberry pi). Hands-on with the integration of sensors and boards. The students write code using Aurdino IDE And Thorny python IDE so that the students should be capable to do real time projects in IoT.

Course Pre/Co-requisites

A9501 - Programming for Problem Solving

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

SDG 11: Sustainable Cities and Communities

Course Outcomes

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

A9606.1. Identify the need of various hardware and software components in developing IoT applications.

A9606.2. Construct IoT applications using Raspberry Pi interface.

A9606.3. Develop IoT applications on embedded platform using Arduino.

A9606.4. Apply Arduino programming in IoT applications.

A9606.5. Analyze and Design IoT solutions to real-world applications.

Course Syllabus

Theory:

1. **Introduction to IoT:** Defining IoT, Characteristics of IoT ,Physical design of IoT, Logical design of IoT,Functional blocks of IoT, Applications of IoT.
2. **IoT Physical Devices and Endpoints:** Introduction to Raspberry Pi-Interfaces (serial, SPI, I2C), Programming Raspberry PI with Python- Controlling LED with Raspberry PI, interfacing an LED and Switch with Raspberry PI and Interfacing a light sensor (LDR) with Raspberry PI.
3. **Programming Arduino:** Introduction, Arduino Boards, Programming-variables, if, loops, functions, digital inputs and outputs, the serial monitor, arrays and strings, analog inputs and outputs, using libraries, Arduino data types and commands. Programming Arduino Uno with Arduino- Controlling LED with Arduino, interfacing an LED and Switch with Arduino and Interfacing a light sensor (LDR)with Arduino.

List of Experiments:

1. Study and Configure Raspberry Pi .
2. Study and Install IDE of Arduino.
3. Write a program using Raspberry Pi for Blink LED.
4. Write a program using Arduino IDE for Blink LED.

5. Implement IoT based weather monitoring system using Raspberry Pi.
6. Write a Arduino program for monitor temperature and humidity using DHT (Digital Humidity and Temperature) sensor and Arduino (weather monitoring).
7. Implement Raspberry Pi based Automated Street Lighting System.
8. Implement Arduino based Automated Street Lighting System.
9. Write an Arduino program for Distance Measurement Using Ultrasonic Sensor and dis- playing on LCD.
10. Implement Raspberry Pi program for Distance Measurement Using Ultrasonic Sensor and displaying on LCD.
11. Write a Program for RGB LED using Raspberry Pi.
12. Write a Program for RGB LED using Arduino.
13. Write a Program for Interfacing PIR Motion Sensor With Rasbperry Pi.
14. Case Study on Smart Home System.

Laboratory Equipment/Software/Tools Required:

1. Computer Systems (PCs) installed with Ubuntu OS (Open source/ Freeware).
2. Arduino IDE(open source/ Freeware), Python(open source/ Freeware)
3. Multimeter
4. Electric Soldering Kit
5. Breadboards
6. Raspberry-Pi Board
7. Type C power supply
8. Micro HDMI To VGA Convertor
9. 16GB Class 10 Memory Card
10. Arduino Uno Board with cable
11. Jumper wires (Male-Male, Male-Female, Female-Female)
12. Resisters pack
13. DHT11 (Digital Humidity and Temperature) sensor
14. LDR (Light Dependent Resistors) Sensor
15. LCD display
16. Ultrasonic Range Finder Module Sensor
17. LED (Red, Green, Yellow)
18. RGB LED
19. PIR Sensor
20. Buzzer

Books and Materials

Text Books:

1. Bahga, Arshdeep, and Vijay Madisetti. *Internet of Things: A Hands-On Approach* Universities Press, 2015.

2. Monk, Simon *Programming Arduino Next Steps: Going Further with Sketches*, 2nd ed., McGraw-Hill Education, 2019.
3. Niku, Saeed B. *Introduction to Robotics: Analysis, Control, Applications*, 3nd ed., Wiley, Feb. 2020.

Reference Books:

1. Raj, Pethuru, and Anupama C. Raman *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*, CRC Press, 2017.
2. Wallace, Shawn, Matt Richardson, and Wolfram Donat *Getting Started with Raspberry Pi*, 4th reprint, Make: Community (SPD/O'Reilly), Oct. 2021.

A9006 – Computational Mathematics Laboratory

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This course provides hands-on experience in solving mathematical problems using computational tools. This course covers numerical methods and implementation using MATLAB or Python. The course helps to develop skills in algorithm development, data visualization, and scientific computing. In addition, the computational methods for real- world mathematical modeling can be applied.

Course Pre/Co-requisites

A9001 - Matrices and Calculus

A9002 - Ordinary Differential Equations and Vector Calculus

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education

SDG 9: Industry, Innovation, and Infrastructure

Course Outcomes

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

A9006.1. Develop the code to find the Eigen values and Eigen Vectors using Python/MATLAB

A9006.2. Develop the code to find solution of Algebraic and Transcendental using Python/MATLAB

A9006.3. Develop the code to find solution of Linear system of equations using Python/MATLAB

A9006.4. Write the code to solve problems of First-Order linear differential equations with constant coefficients

A9006.5. Write the code to solve problems of Higher order linear differential equations with constant coefficients

Course Syllabus

List of Experiments:

Visualize all solutions graphically using programs.

1. Eigen values and Eigen Vectors
   a. Finding real and complex Eigen values.
   b. Finding Eigen vectors.
2. Solution of Algebraic and Transcendental Equations - Bisection method, Newton Raphson Method
   a. Root of a given equation using Bisection method.
   b. Root of a given equation using Newton Raphson Method.
3. Linear system of equations - Jacobi's iteration method and Gauss-Seidal iteration method
   a. Solution of given system of linear equations using Jacobi's method.
   b. Solution of given system of linear equations using Gauss-Seidal method.

4. First-Order ODEs - Exact and non-exact equations, Applications: exponential growth/decay, Newton's law of cooling
   a. Solving exact and non-exact equations.
   b. Solving exponential growth/decay and Newton's law of cooling problems.
5. Higher order linear differential equations with constant coefficients
   a. Solving homogeneous ODEs.
   b. Solving non-homogeneous ODEs.

Books and Materials

Text Books:

1. Rajkumar Basal, Ashok Kumar Geo, and Manoj Kumar Sharma. *MATLAB and Its Applications in Engineering*. Pearson.
2. Kenneth A. Lambert. *The Fundamentals of Python: First Programs*. Cengage Learning, 2011.
3. Allen B. Downey. *Think Python*. 1st ed., O'Reilly Media.

Reference Books:

1. William Mitchell, Povel Solin, Martin Novak, et al. *Introduction to Python Programming*. NCLab Public Computing, 2012.
2. Jacob Fredslund. *Introduction to Python Programming*. 2007.
3. John C. Lusth. *An Introduction to Python*. University of Alabama, 2011.
4. Dave Kuhlman. *Introduction to Python*. 2008.

A9517 – User Experience Design

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	30	0	30	1	40	60	100

Course Description

Course Overview

This course introduces Flutter as a modern cross-platform development framework and guides learners through the foundations needed to build applications. It begins with an orientation to the framework, its architecture, and how widgets function at the heart of Flutter applications. Learners will explore the distinction between stateless and stateful components and how they interact within widget and element trees. The setup process for different operating systems is covered step by step, ensuring a smooth environment configuration. Once the tools are ready, learners quickly progress into building a simple app and experiencing the rapid development cycle with hot reload. The course also emphasizes the importance of themes and styling for creating visually consistent applications. Fundamentals of the Dart programming language are introduced to strengthen coding skills and enable smooth integration with Flutter. Students learn best practices for structuring projects, organizing code, and refactoring for maintainability. A wide variety of common UI components and input controls are demonstrated to build interactive interfaces. Finally, the course explores animations and dynamic behaviors, enabling learners to design engaging and responsive user experiences.

Course Pre/Co-requisites

A9601 - Object Oriented Programming
A9604 - Web Application Engineering

Relevant SDG(s)

SDG 4 – Quality Education
SDG 8 - Decent Work and Economic Growth
SDG 11 - Sustainable Cities and Communities

Course Outcomes

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

A9517.1. Setup Flutter SDK and project configuration skills to build and verify cross-platform mobile applications.

A9517.2. Implement Flutter UI components, themes, and navigation techniques to create functional and interactive mobile interfaces.

A9517.3. Develop Dart programs using various programming constructs to solve real time problems.

A9517.4. Design animation, form validation, and external package integration in Flutter applications to enhance application usability and performance.

A9517.5 Organize Flutter widget trees, applying modular programming techniques to build maintainable and reusable application components.

Course Syllabus

Theory:

1. **Introducing Flutter and Getting Started:** Introducing Flutter, Defining Widgets and Elements, Understanding Widget Lifecycle Events, The Stateless Widget Lifecycle, The Stateful Widget Lifecycle,

Understanding the Widget Tree and the Element Tree, Stateless Widget and Element Trees, Stateful Widget and Element Trees. Installing the Flutter SDK: Installing on macOS (System Requirements, Get the Flutter SDK, Check for Dependencies, iOS Setup: Install Xcode), Android Setup (Install Android Studio, Set Up the Android Emulator), Installing on Windows (System Requirements, Get the Flutter SDK, Check for Dependencies, Install Android Studio, Set Up the Android Emulator), Installing on Linux (System Requirements, Get the Flutter SDK, Check for Dependencies, Install Android Studio, Set Up the Android Emulator), Configuring the Android Studio Editor.

Creating a Hello World App: Setting Up the Project, Using Hot Reload, Using Themes to Style Your App, Using a Global App Theme, Using a Theme for Part of an App, Understanding Stateless and Stateful Widgets, Using External Packages, Searching for Packages, Using Packages.

2. **Learning Dart Basics:** Use of Dart, Commenting Code, Running the `main()` Entry Point, Referencing Variables, Declaring Variables, Numbers, Strings, Booleans, Lists, Maps, Runes. Using Operators, Flow Statements (if and else, ternary operator, for loops, while and do-while, while and break, continue, switch and case). Using Functions, Import Packages, Using Classes, Class Inheritance, Class Mixins, Implementing Asynchronous Programming.
3. **Creating a Starter Project Template:** Creating and Organizing Folders and Files, Structuring Widgets, Understanding the Widget Tree, Introduction to Widgets, Building the Full Widget Tree, Building a Shallow Widget Tree, Refactoring with a Constant, Refactoring with a Method, Refactoring with a Widget Class.
4. **Using Common Widgets and Animations:** Using Basic Widgets: Safe Area, Container, Text, Rich Text, Column, Row, Column and Row Nesting. Buttons: Floating Action Button, Flat Button, Raised Button, Icon Button, Popup Menu Button, Button Bar. Using Images and Icons: Asset Bundle, Image, Icon. Using Decorators. Using the Form Widget to Validate Text Fields, Checking Orientation. Adding Animation to an App: Using Animated Container, Using Animated Cross Fade, Using Animated Opacity, Using Animation Controller, Using Staggered Animations.

List of Experiments:

1. Write a program to install and configure Flutter SDK and verify the setup using `flutter doctor`.
2. Write a program to create a new Flutter project and explore its folder structure.
3. Write a program to display "Hello World" using a `StatelessWidget` and demonstrate Hot Reload.
4. Write a program to implement global and local themes in a Flutter application.
5. Write a program to add an external package (e.g., `http`) and display fetched data.
6. Write a program in Dart to demonstrate variables, data types, and operators.
7. Write a program in Dart to implement control statements (if-else, for, while, switch) and functions.
8. Write a program in Dart to demonstrate classes, objects, and inheritance.
9. Write a program in Dart to implement asynchronous programming using `async`, `await`, and `Future`.
10. Write a program to create a structured widget tree and refactor it using methods and widget classes.
11. Write a program to demonstrate the use of basic widgets such as `Container`, `Row`, `Column`, `Text`, `Image`, and `Icon`.
12. Write a program to create a login form using `Form` and `TextFormField` with validation.
13. Write a program to implement navigation between two screens using different buttons.

14. Write a program to add animations using `AnimatedContainer`, `AnimatedOpacity`, and staggered animations.

Laboratory Equipment/Software/Tools Required:

1. Computer Systems installed with Ubuntu Operating System (Open Source/Freeware).
2. Flutter SDK, Dart SDK (bundled with Flutter), and Android Studio (latest version) with:Flutter plugin, Dart plugin (Open Source/Freeware).
3. Android Emulator setup, Xcode, Visual Studio Code (Open Source/Freeware).

Books and Materials

Text Book(s)

1. Marco L. Napoli, *Beginning Flutter: A Hands on Guide to App Development*, 1st Edition, Wrox Publisher.

Reference Book(s)

1. *Flutter for Beginners: An introductory guide to building cross-platform mobile applications with Flutter and Dart 2*, Packt Publishing Limited.
2. Rap Payne, *Beginning App Development with Flutter: Create Cross-Platform Mobile Apps*, 1st Edition, Apress.
3. Frank Zammetti, *Practical Flutter: Improve your Mobile Development with Google's Latest Open-Source SDK*, 1st Edition, Apress.

A9024 – Community Driven Product Evaluation

Teaching and Learning Scheme				Hours	Credits	Assessment Marks		
CI		LI	TW+SL	H	C	CIE	SEE	Total
L	T	P	SL					
0	0	0	45	45	1	40	60	100

Course Description

Course Overview

This course provides mathematical knowledge required to analyze problems encountered in engineering. In this course, the students are acquainted with ordinary differential equations of first and higher order and Laplace transforms, vector calculus. In addition, this course can be applied in many areas of engineering such as wireless communication, signal processing, robotics and animation.

Course Pre/Co-requisites

A9023 - Technology Entrepreneurship

Relevant Sustainable Development Goals (SDGs)

SDG 4: Quality Education
SDG 11: Sustainable Cities and Communities
SDG 12: Responsible Consumption and Production
SDG 17: Partnerships for the Goals

Course Outcomes

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

A9024.1. Apply structured evaluation frameworks to assess technical, functional, and social impact of products.

A9024.2. Conduct community-centered product testing and collect actionable feedback.

A9024.3. Benchmark products against industry standards and competitor solutions.

A9024.4. Analyze evaluation data to identify strengths, weaknesses, and areas for improvement.

A9024.5. Integrate knowledge from all prior courses to produce a comprehensive commercialization or patent readiness report.

Course Syllabus

Unit-I:

Product Evaluation Fundamentals: Purpose, scope, and importance of product evaluation in community contexts. Key Performance Indicators (KPIs), usability, and sustainability metrics. Ethical considerations in testing with communities.

Unit-II:

Standards, Compliance and Benchmarking: Relevant industry, safety, and environmental standards. Social impact and sustainability assessment frameworks. Competitive benchmarking and market gap analysis.

Unit-III:

Community Centered Testing and Data Collection: Designing and executing real-world product trials. Feedback mechanisms: surveys, interviews, observation, analytics. Collecting and categorizing qualitative and quantitative data.

Unit-IV:

Data Analysis and Product Improvement Planning: Analytical tools for interpreting evaluation results. Identifying design gaps and improvement opportunities. Translating insights into actionable product enhancement plans.

Unit-V:

Integrated Product Documentation: Consolidating insights from design thinking, product development, entrepreneurship, and evaluation. preparing a comprehensive commercialization or patent readiness dossier. and presenting outcomes to a review panel for validation and approval.

Activity Plan

Week	Unit	Objective	Teaching Method	In-Class Activities	Assignments / Assessments	CO Mapping
1	Unit-I: Evaluation Fundamentals	Understand purpose, scope & importance of product evaluation	Concept briefing	Discussion: "Why evaluation is crucial for community-driven products"	Reflection note on role of evaluation in product lifecycle	CO1
2	Unit-I: KPIs & Metrics	Learn KPIs, usability & sustainability measures	Concept briefing + Case examples	Teams define KPIs for a sample product	Submit KPI framework for selected case	CO1
3	Unit-I: Ethics in Evaluation	Apply ethical considerations in testing	Problem-based learning + Collaborative exchange	Debate: "Ethics vs. Innovation speed in testing"	Short essay on ethical challenge in testing	CO1
4	Unit-II: Standards & Compliance	Learn relevant industry & safety standards	Demo session + Case study	Review product compliance requirements from standards body	Submit compliance checklist for a product	CO2, CO3
5	Unit-II: Benchmarking	Apply benchmarking frameworks	Hands-on session + Benchmarking activity	Benchmark 2 community products against market leaders	Submit benchmarking chart	CO3
6	Unit-II: Market Gap Analysis	Identify market gaps & opportunities	Hands-on session	Teams map competitor strengths vs weaknesses	Submit market gap report	CO3
7	Unit-III: Community-Centered Testing	Design product trials with stakeholders	Hands-on workshop + Role-play	Simulate community feedback session	Submit trial design protocol	CO2

Week	Unit	Objective	Teaching Method	In-Class Activities	Assignments / Assessments	CO Mapping
8	Unit-III: Feedback Mechanisms	Practice data collection methods	Practical session + Peer feedback	Run mock survey/interview for a prototype	Submit collected sample data	CO2
9	Unit-IV: Data Analysis Tools	Analyze evaluation data	Analytical lab + Software demo	Use basic data tools (Excel/SPSS/PowerBI) to interpret results	Submit initial analysis report	CO4
10	Unit-IV: Product Improvement	Translate insights into action	Product improvement exercise	Teams identify weaknesses & propose enhancements	Submit product improvement plan	CO4
11	Unit-V: Integrated Documentation	Consolidate learnings from all prior courses	Documentation	Draft commercialization or patent dossier	Submit draft dossier	CO5
12	Unit-V: Final Showcase	Present integrated evaluation outcomes	Showcase + Expert panel review	Final presentations with reports, feedback loop	Submit final commercialization/patent readiness dossier	CO5

Books and Materials

Text Books:

1. Deependra Sharma *Entrepreneurship in India*, Routledge, 2023.
2. Cooper, Robert G *Winning at New Products: Creating Value Through Innovation*, Basic Books, 2011.

Reference Books:

1. Dr. S. Glory Swarupa, Swapna Vanamala *Innovation, Incubation and Intellectual Property Rights: Experiences of Developing Countries*, 2023.