MOOGAMBIGAI CHARITABLE AND EDUCATIONAL TRUST

Rajarajeswari College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)
#14, Ramohalli Cross, Kumbalagodu, Mysore Road, Bengaluru–560074

Computer Science and Design

III & IV Semester Scheme and Syllabus

(2025 Scheme)

VISION

To empower young minds through technology, research and innovation, to produce technically competent and socially responsible professionals in higher education.

MISSION

1. To deliver excellence in education through innovative teaching, impactful research, and continuous skill development, preparing students to meet global challenges with technical expertise and ethical responsibility.
2. To foster a transformative learning environment that integrates technology, research and practical experience, empowering students to become skilled professionals and socially conscious leaders.
3. To cultivate a culture of lifelong learning and professional excellence by encouraging creativity, research, and community engagement, equipping students with the skills to thrive in a dynamic world.
4. To provide a holistic educational experience that combines advanced technology, hands-on research, and community-focused learning, shaping students into competent, ethical professionals who contribute positively to society.

QUALITY POLICY

Rajarajeswari College of Engineering is committed to imparting quality technical education that nurtures competent, ethical professionals with global relevance. We ensure academic excellence through a dynamic, outcome-based curriculum, experienced faculty, and cutting-edge infrastructure. Continuous improvement is driven by innovation, research and strong industry collaboration. We foster holistic development and a progressive environment that supports lifelong learning, teamwork, and professional growth.

CORE VALUES

Academic Excellence, Integrity, Innovation, Global Competence, Continuous Improvement.

Computer Science and Design

DEPARTMENT VISION

To craft a new generation of innovations who merge technology with design to shape them into skilled and socially responsible professionals.

DEPARTMENT MISSION

1. To deliver the environment to become industry ready professionals, Researchers and Entrepreneurs by offering courses on cutting edge technology and advanced practical courses for the students.
2. Impart high quality experiential learning to get expertise in modern software tools and to cater to the real time requirements of the industry.
3. Inculcate problem solving and team building skills promote lifelong learning to demonstrate with a sense of societal and ethical responsibilities.

PROGRAM OUTCOMES (POs)

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, modeling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modeling 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, 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 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 EDUCATIONAL OBJECTIVES (PEOs)

PEO-1: Graduates of the program will possess strong educational foundation in mathematics, science, and computer science & engineering necessary for pursuing career and/ or higher studies in computing as well as other allied professions.

PEO-2: Graduates with an interest in, and aptitude for lifelong learning will be engaged in learning, understanding, and applying new ideas and technologies as the field evolves to solve engineering problems, design appropriate computing systems that are technically sound economically viable and socially acceptable and be responsible engineering and computing professionals.

PEO-3: Graduates will be informed leaders, effective communicators who work efficiently with diverse teams, promote and practice appropriate ethical moral and codes.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO-1: Understand, apply, and demonstrate advanced technical skills in problem solving and leadership, as well as an understanding of system integration and the practical technological problems of end users.

PSO-2: An ability to design, implement, and evaluate a software or a software/hardware system, component, or process to meet desired needs within realistic constraints such as memory, runtime efficiency, as well as other socio-economic constraints.

INDEX

III Semester			
Sl. No.	**Course Code**	**Course Title**	**Page No.**
1.	B25MCS301	Probability and Statistics	1
2.	B25CS302	Data Structures and its Applications	3
3.	B25CS303	Digital Design and Computer Organization	5
4.	B25CS304	Operating Systems	8
5.	B25CS305	C++Programming	10
6.	B25CSL306	Data Structures and its Applications Lab	12
7.	B25CSL307	C++ Programming Lab	15

IV Semester			
Sl. No.	**Course Code**	**Course Title**	**Page No.**
1.	B25MCS401	Discrete Mathematical Structures	17
2.	B25CS402	Database management Systems	19
3.	B25CS403	Design and Analysis of Algorithms	21
4.	B25CS404	Java Programming	24
5.	B25CSL406	Database Management Systems Lab	26
6.	B25CSL407	Java Programming Lab	29

MOOGAMBIGAI CHARITABLE AND EDUCATIONAL TRUST

Rajarajeswari College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

Scheme of Teaching and Examinations – 2025 Scheme

Outcome Based Education (OBE) and Choice Based Credit System (CBCS)

III Semester **Academic Year: 2026-27**

S.No	Course Category and Course Code		Course Title	TD	Teaching Hours / Semester				Examination				Credits
					Lecture	Tutorial	Practical	TW + SL	CIE Marks	SEE Marks	Total marks	SEE Duration in hours	
					L	T	P	S					
1.	ASC	B25MCS301	Probability and Statistics	Maths	**42**	**28**	**0**	**50**	**50**	**50**	**100**	**3**	**4**
2.	PCC	B25CS302	Data Structures and its Applications	CG	**28**	**28**	**0**	**34**	**50**	**50**	**100**	**3**	**3**
3.	IPCC	B25CS303	Digital Design and Computer Organization	CG	**42**	**0**	**28**	**50**	**50**	**50**	**100**	**3**	**4**
4.	PCC	B25CS304	Operating Systems	CG	**42**	**0**	**0**	**48**	**50**	**50**	**100**	**3**	**3**
5.	PCC	B25CS305	C++ Programming	CG	**42**	**0**	**0**	**48**	**50**	**50**	**100**	**3**	**3**
6.	PCCL	B25CSL306	Data Structures and its Applications Lab	CG	**0**	**0**	**28**	**2**	**50**	**50**	**100**	**3**	**1**
7.	AEC	B25CSL307	C++ Programming Lab	CG	**0**	**0**	**28**	**2**	**50**	**50**	**100**	**3**	**1**
8.	SDC	B25CGP308	Community / Societal Project	CG	**0**	**0**	**0**	**30**	**50**	**50**	**100**	**3**	**1**
9.	NCMC	B25MAD309	Additional Mathematics - I	Maths	**42**	**0**	**0**	**0**	**100**	**-**	**100**	**-**	**PP**
10.	NCMC	B25NCK310A/ B/C/D/E	National Service Scheme /National Cadet Corps / Physical Education Yoga / Music	HSS	**0**	**0**	**0**	**28**	**100**	**-**	**100**		**PP**
11.	SEC	B25SKL311	Skill Laboratory	Concerned Dept	**0**	**0**	**30**	**0**	**100**	**-**	**100**	**-**	**1**
								TOTAL	**700**	**400**	**1100**		**21**

ASC: Applied Science Courses, BSC: Basic Science Course, HSMC: Humanities Studies and Management courses, IPCC: Integrated Professional Core Course, PCC: Professional Core Course, PCCL: Professional Core Course laboratory, NCMC: Non-Credit Mandatory Course, AEC: Ability Enhancement Course, SDC: Skill Development Course, L: Lecture, T: Tutorial, P: Practical, S: Self-Learning, CIE: Continuous Internal Evaluation, SEE: Semester End Evaluation, PP/NP: Pass/Not Pass, XX: Programme Code (EC, CS, IS etc.,), Y: A/B/C/D**,** K: Indicates Common Course to all the streams of Engineering

NSS / NCC / PE / Yoga / Music: All students are required to register for any one of the following courses: National Service Scheme/National Cadet Corps/Physical Education/Yoga/ Music - with the respective course coordinator during the first week of the third semester. College shall submit Continuous Internal Evaluation (CIE) marks for each semester based on the activities completed by students under these selected courses. Students may opt for different activities/options across semesters. For instance, a student participating in PE during 3rd semester may choose NSS in the 4th semester or Yoga. Activities shall be conducted over two semesters (III&IV), and successful completion of the registered course/or courses along with the required CIE score is mandatory for the award of the degree. Institutions must ensure that events are appropriately scheduled and reflected in the semester-wise calendar for NSS, PE, Music, and Yoga activities.

These courses shall not be considered for the calculation of SGPA or CGPA and for vertical progression. However, completion of course(s) is compulsory for degree eligibility.

The courseB25XX307Y – Ability Enhancement Course Laboratory can be offered either as a single compulsory course. Alternatively, the course B25XX307Y - Ability Enhancement Course Laboratory shall be offered as multiple elective options under the course codesB25XX307Y (where Y =A/B/C/D)

HOD **Dean-Academics** **Principal**

MOOGAMBIGAI CHARITABLE AND EDUCATIONAL TRUST

Rajarajeswari College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

Scheme of Teaching and Examinations – 2025 Scheme

Outcome Based Education (OBE) and Choice Based Credit System (CBCS)

IV Semester **Academic Year: 2026-27**

S. No	Course Category and Course Code		Course Title	TD	Teaching Hours / Semester				Examination				Credits
					Lecture	Tutorial	Practical	TW + SL	CIE Marks	SEE Marks	Total marks	SEE Duration in hours	
					L	T	P	S					
1.	ASC	B25MCS401	Discrete mathematical structures	Maths	**42**	**28**	**0**	**50**	**50**	**50**	**100**	**3**	**4**
2.	PCC	B25CS402	Database Management Systems	CG	**42**	**28**	**0**	**50**	**50**	**50**	**100**	**3**	**4**
3.	IPCC	B25CS403	Design and Analysis of Algorithms	CG	**42**	**0**	**28**	**50**	**50**	**50**	**100**	**3**	4
4.	PCC	B25CS404	Java Programming	CG	**42**	**0**	**0**	**48**	**50**	**50**	**100**	**3**	**3**
5.	BSC	B25BEK405	Biology for Engineers	CG	**28**	**0**	**0**	**32**	**50**	**50**	**100**	**2**	**2**
6.	PCCL	B25CSL406	Database Management Systems Lab	CG	**0**	**0**	**28**	**2**	**50**	**50**	**100**	**3**	**1**
7.	AEC	B25CSL407	Java Programming Lab	CG	**0**	**0**	**28**	**2**	**50**	**50**	**100**	**3**	**1**
8.	SDC	B25CGP408	Environmental Science Project	CG	**0**	**0**	**0**	**30**	**50**	**50**	**100**	**3**	**1**
9.	NCMC	B25MAD409	Additional Mathematics - II	Maths	**42**	**0**	**0**	**0**	**100**	**-**	**100**		**PP**
10.	NCMC	B25PEK410A/ B/C/D/E	National Service Scheme/National Cadet Corps/Physical Education Yoga / Music		**0**	**0**	**0**	**28**	**100**	**-**	**100**		**PP**
11.	SEC	B25SKL311	Skill Laboratory	Concerned Dept	**0**	**0**	**30**	**0**	**100**	**-**	**100**	**-**	**1**
								TOTAL	**700**	**400**	**1100**		**21**

ASC: Applied Science Courses, BSC: Basic Science Course, HSMC: Humanities Studies and Management courses, IPCC: Integrated Professional Core Course, PCC: Professional Core Course, PCCL: Professional Core Course laboratory, NCMC: Non-Credit Mandatory Course, AEC: Ability Enhancement Course, SDC: Skill Development Course, L: Lecture, T: Tutorial, P: Practical, S: Self-Learning, CIE: Continuous Internal Evaluation, SEE: Semester End Evaluation, PP/NP: Pass/Not Pass, XX: Programme Code (EC, CS, IS etc.,), Y: A/B/C/D**,** K: Indicates Common Course to all the streams of Engineering

MOOGAMBIGAI CHARITABLE AND EDUCATIONAL TRUST

Rajarajeswari College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

Scheme of Teaching and Examinations – 2025 Scheme

Outcome Based Education (OBE) and Choice Based Credit System (CBCS)

NSS / NCC / PE / Yoga / Music: All students are required to register for any one of the following courses: National Service Scheme/National Cadet Corps/Physical Education /Yoga/ Music - with the respective course coordinator during the first week of the third semester.

Colleges shall submit Continuous Internal Evaluation (CIE) marks for each semester based on the activities completed by students under the selected course. Students may opt for different activities/options across semesters. For instance, a student participating in PE during 3rd semester may choose NSS in the 4th semester or Yoga. Activities shall be conducted over two semesters (III and IV), and successful completion of the registered course/or courses along with the required CIE score is mandatory for the award of the degree. Institutions must ensure that events are appropriately scheduled and reflected in the semester-wise calendar for NSS, PE, Music, and Yoga activities.

These courses shall not be considered for the calculation of SGPA or CGPA and for vertical progression. However, completion of course(s) is compulsory for degree eligibility.

The course **B25XX407Y** – Ability Enhancement Course Laboratory can be offered either as a single compulsory course. Alternatively, the course **B25XX407Y** - Ability Enhancement Course Laboratory shall be offered as multiple elective options under the course codes **B25XX407Y** (where Y =A/B/C/D)

Environmental Science Project: The Environmental Science Project is an applied learning component designed to develop students' awareness, understanding, and responsibility toward the environment. It provides an opportunity to study real-world environmental issues and apply scientific and engineering principles to design feasible and sustainable solutions.

HoD **Dean-Academics** **Principal**

Semester-III					
Probability and Statistics					
Category: ASC					
(Common to CSE, CSD, CSE(IC), ISE, AIML)					
Course Code	:	**B25MCS301**	CIE	:	50 Marks
Teaching Hours L:T:P:S	:	42:28:0:50	SEE	:	50 Marks
Total Hours/Semester	:	120	Total	:	100 Marks
Credits	:	4	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To introduce the concept of random variables, probability distributions, specific discrete and continuous distributions with practical application in Computer Science Engineering and social life situations.
2.	To understand and analyze the probabilistic relationships between multiple random variables and determining relationships like co-variance and correlation.
3.	To Provide the principles of statistical inference sand the basics of hypothesis testing with emphasis on some commonly Encountered hypotheses.
4.	To Determine whether an input has a statistically significant effect on the system's response through ANOVA testing.

Module-1	**No. of Hrs**
Probability Distributions: Review of Basic Probability Theory. Random Variables (Discrete And Continuous), Probability Mass and Density Functions, Mean And Variance. Binomial, Poisson, Exponential and Normal Distributions - Problems (Derivations For Mean And Standard Deviation For Binomial And Poisson Distributions Only)-Illustrative Examples.	14
Module-2	**No. of Hrs**
Joint Probability Distribution & MarkovChain: Joint Probability Distribution For Two Discrete Random Variables, Expectation, Covariance and Correlation. Markov Chain: Introduction To Stochastic Process, Probability Vectors, Stochastic Matrices, Regular Stochastic Matrices, Markov Chains, Higher Transition Probabilities, Stationary Distribution of Regular Markov Chains and Absorbing States.	14
Module-3	**No. of Hrs**
Statistical Inference-I: Introduction, Sampling Distribution, Standard Error, Testing Of Hypothesis, Levels Of Significance, Test of Significance, Confidence Limits, Simple Sampling Of Attributes, Test Of Significance For Large Samples, Comparison Of Large Samples.	14
Module-4	**No. of Hrs**
Statistical Inference-II: Sampling variables, Central limit theorem and confidence limits for unknown means. Test of significance for means of two small samples, Student's distribution, Chi-square distribution as a test of goodness of fit .F-distribution.	14
Module-5	**No. of Hrs**
Design of Experiments &ANOVA: Principles of experimentation in design, analysis of completely randomized design, randomized block design. The ANOVA, technique, basic principle of ANOVA, one-way ANOVA, two-way ANOVA, Latin-square design, and analysis of co-variance.	14

Course Outcomes: At the end of the course, the students will be able to	
CO1	Apply fundamental principles of probability to analyze uncertain events and real-world engineering problems.
CO2	Model and an analyze random variables and their probability distributions to interpret stochastic behavior in engineering contexts.
CO3	Apply appropriate probability distribution models for solving practical engineering scenarios**.**
CO4	Utilize statistical methods and tools for data analysis and decision-making in engineering problem- solving.
CO5	Compute and interpret confidence intervals for population parameters**,** particularly the mean, to support statistical inference.

Text Books	
1.	Ronalde, Walpole, Raymond HMyers, SharonL Myers&Keying Ye"Probability & Statistics for Engineers & Scientists", Pearson Education, 9th edition, 2017.
2.	PeterBruce, Andrew Bruce & Peter Gedeck "Practical Statistics for Data Scientists"O 'Reilly Media, Inc., 2nd edition 2020.
Reference Text Books	
1.	ErwinKreyszig, "Advanced Engineering Mathematics", John Wiley&Sons, 9th Edition, 2006.
2.	B.S. Grewal "Higher Engineering Mathematics", Khannapublishers, 44th Ed. 2021.
3.	G Haribaskaran "Probability, Queuing Theory & Reliability Engineering", Laxmi Publication, LatestEdition,2006

Web links and Video lectures(e-Resources)
1. https://nptel.ac.in/courses/12286025 2. VTUEDUSATPROGRAMME– 20 3. http://www.class-central.com/subject/math(MOOCs)

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**40 (Average of Best two Assessments)**	**50**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
AAT	**Any Two Assessments as per Regulations**	**20**	**10**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The Question paper for each course contains two parts, Part – A and Part – B.
2. Part – A consists of **Short Answer Questions** (2 Marks/1 mark) for 20 marks covering the complete syllabus and it is compulsory. Multiple Choice Questions are not allowed.
3. Part – B consists of 10 questions, two questions of 16 marks (with max. of 3 sub questions) from each module with internal choice. Students shall answer five full questions, selecting one full question from each module.

CO-PO Mapping

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

Level3-High, Level2-Moderate, Level1 -Low

Semester III					
Data Structures and its Applications **Category: PCC** **(Common to CSE, CSD, AIML, CSE(IC), ISE)**					
Course Code	:	**B25CS302**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	28:28:0:34	SEE	:	50 Marks
Total Hours/Semester	:	90	Total	:	100 Marks
Credits	:	3	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To explain fundamentals of data structures and their applications essential for programming/problem solving.
2.	To apply stack, Queue and recursion operations to solve real world problems.
3.	To illustrate linear representation of data structures: Stack, Queues, Lists, Trees and Graphs.
4.	To demonstrate sorting and searching algorithms.
5.	To find suitable data structure during application development/Problem Solving.

Module– 1	**No. of Hrs**
Introduction: Data Structures, Classifications (Primitive &Non-Primitive), Data structure Operations, Review of Arrays, Structures, Self-Referential Structures, and Unions. Pointers and Dynamic Memory Allocation Functions. Representation of Linear Arrays in Memory, dynamically allocated arrays. Array Operations: Traversing, inserting, deleting, searching, and sorting. Multidimensional Arrays, Polynomials and Sparse Matrices. Strings: Basic Terminology, Storing, Operations and Pattern Matching algorithms. Programming Examples.	11
Module– 2	**No. of Hrs**
Stacks: Definition, Stack Operations, Array Representation of Stacks, Stacks using Dynamic Arrays, Stack Applications: Polish notation, Infix to postfix conversion, evaluation of postfix expression. Recursion - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi, Ackerman's function. Queues: Definition, Array Representation, Queue Operations, Circular Queues, Circular queues using Dynamic arrays, Dequeues, Priority Queues, A Mazing Problem. Multiple Stacks and Queues. Programming Examples.	11
Module– 3	**No. of Hrs**
Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage Collection. Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists, Circular linked lists, and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Sparse matrix representation. Programming Examples.	11
Module– 4	**No. of Hrs**
Trees: Terminology, Binary Trees, Properties of Binary trees, Array and linked Representation of Binary Trees, Binary Tree Traversals - Inorder, postorder, preorder; Additional Binary tree operations. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression, Programming Examples	11

Module– 5	**No. of Hrs**
Graphs: Definitions, Terminologies, Matrix and Adjacency List Representation of Graphs, Elementary Graph operations, Traversal methods: Breadth First Search and Depth First Search. Sorting and Searching: Insertion Sort, Radix sort, Address Calculation Sort. Hashing: Hash Table organizations, Hashing Functions, Static and Dynamic Hashing. Files and Their Organization: Data Hierarchy, File Attributes, Text Files and Binary Files, Basic File Operations, File Organizations and Indexing.	11

Course Outcomes: At the end of the course, the students will be able to	
CO1	Use different types of data structures, operations and algorithms
CO2	Apply searching and sorting operations on files
CO3	Use stack, Queue, Lists, Trees and Graphs in problem solving
CO4	Implement all data structures in a high-level language for problem solving.
CO5	Identify the alternative implementations of data structure to solve real world problems

Text Books	
1.	Ellis Horowitz and SartajSahni, Fundamentals of Data Structures in C, 2nd Edition, Universities Press, 2014.
2.	Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Edition, McGraw Hill, 2014.

References Text Books	
1.	Gilberg& Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Edition, Cengage Learning, 2014.
2.	Reema Thareja, Data Structures using C, 3rd Ed, Oxford press, 2012.
3.	Jean-Paul Tremblay & Paul G. Sorenson, An Introduction to Data Structures with Applications, 2nd Edition, McGraw Hill, 2013.

ASSESSMENT STRUCTURE:
The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**40 (Average of Best two Assessments)**	**50**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
AAT	**Any Two Assessments as per Regulations**	**20**	**10**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):
1. The Question paper for each course contains two parts, Part – A and Part – B.
2. Part – A consists of **Short Answer Questions** (2 Marks/1 mark) for 20 marks covering the complete syllabus and it is compulsory. Multiple Choice Questions are not allowed.
3. Part – B consists of 10 questions, two questions of 16 marks (with max. of 3 sub questions) from each module with internal choice. Students shall answer five full questions, selecting one full question from each module.

CO-PO Mapping

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

Semester-III					
Digital Design and Computer Organization **Category: IPCC** **(Common to CSE, AIML, CSD, ISE)**					
Course Code	:	**B25CS303**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	42:0:28:50	SEE	:	50 Marks
Total Hours/Semester	:	120	Total	:	100 Marks
Credits	:	4	SEE Duration	:	3 Hrs

Course Learning Objective is	
1	To demonstrate the functionalities of binary logic system.
2	To explain the working of combinational and sequential logic system.
3	To realize HDL Verilog programmes.
4	To realize the basic structure of computer system.
5	To illustrate the working of I/O operations and processing unit.

Module– 1	**No. of Hrs.**
Introduction: Binary Logic, Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Digital Logic Gates, Introduction, The Map Method, Four-Variable Map, Don't-Care Conditions, NAND and NOR Implementation, Other Hardware Description Language – Verilog Model of a simple circuit.	9
Module– 2	**No. of Hrs.**
Combinational Logic: Introduction, Combinational Circuits, Design Procedure, Binary Adder-Sub tractor, Decoders, Encoders, Multiplexers. HDL Models of Combinational Circuits – Adder, Multiplexer, Encoder. Sequential Logic: Introduction, Sequential Circuits, Storage Elements: Latches, Flip-Flops.	9
Module– 3	**No. of Hrs.**
Basic Structures of Computer: Functional Units, Basic Operational Concepts, Bus structure, Performance – Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Machine Instructions and Programs: Memory Location and Addresses, Memory Operations, Instruction and Instruction sequencing, Addressing Modes	8
Module– 4	**No. of Hrs.**
Input/output Organization: Accessing I/O Devices, Interrupts – Interrupt Hardware, Enabling and Disabling Interrupts, Handling Multiple Devices, Direct Memory Access: Bus Arbitration, Speed, size and Cost of memory systems. Cache Memories – Mapping Functions	8
Module– 5	**No. of Hrs.**
Basic Processing Unit: Some Fundamental Concepts: Register Transfers, Performing ALU operations, fetching a word from Memory, Storing a word in memory. Execution of a Complete Instruction. Pipelining: Basic concepts, Role of Cache memory, Pipeline Performance.	8

Course Outcomes: At the end of the course, the students will be able to	
CO1	Apply the K–Map techniques to simplify various Boolean expressions.
CO2	Design different types of combinational and sequential circuits along with Verilog programs.
CO3	Describe the fundamentals of machine instructions, addressing modes and Processor performance.
CO4	Explain the approaches involved in achieving communication between processor and I/O devices
CO5	Analyse internal Organization of Memory and Impact of cache/Pipelining on Processor Performance.

LABORATORY

Sl. No	List of Experiments
1.	Given a 4-variable logic expression, simplify it using appropriate technique and simulate the same using basic gates.
2.	Design a 4 bit full adder and subtractor and simulate the same using basic gates.
3.	Design a 2 bit half adder and subtractor and simulate the same using NAND and NOR gates
4.	Given a 4-variable logic expression, simplify it using appropriate technique and simulate the same using Verilog HDL
5.	Design Verilog HDL to implement simple circuits using structural, Data flow and Behavioural model.
6.	Design Verilog HDL to implement Binary Adder-Subtractor – Half and Full Adder, Half and Full Subtractor.
7.	Design Verilog HDL to implement Decimal adder.
8.	Design Verilog program to implement Different types of multiplexer like 2:1, 4:1 and 8:1.
9.	Design Verilog program to implement types of De-Multiplexer.
10.	Design Verilog program for implementing various types of Flip-Flops such as SR, JK and D.

Text Books	
1.	M. Morris Mano & Michael D. Ciletti, Digital Design With an Introduction to Verilog Design, 5th edition, Pearson Education
2.	Carl Hamacher, ZvonkoVranesic, SafwatZaky, Computer Organization, 5th Edition, Tata McGraw Hill.
References	
1.	Carl Hamacher, ZvonkoVranesic, SafwatZaky, NaraigManjikian, "Computer Organization and Embedded Systems", 6th Edition, Tata McGraw-Hill, 2012.
2.	William Stallings, "Computer Organization and Architecture – Designing for Performance", 10th Edition, Pearson Education, 2016.
3.	M. Morris Mano, "Digital Logic and Computer Design", Pearson Education, 2016.

ASSESSMENT STRUCTURE:
The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**40 (Average of Best two Assessments)**	**50/2 = 25**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
AAT	**Any One Assessments as per regulations**	**20**	**10**	
Laboratory	**Lab Conduction & Record**	**Evaluating each expt. for 10 marks**	**10**	**25**
	Lab Internal Test	**50**	**15**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The Question paper for each course contains two parts, Part – A and Part – B.
2. Part – A consists of **Short Answer Questions** (2 Marks/1 mark) for 20 marks covering the complete syllabus and it is compulsory. Multiple Choice Questions are not allowed.
3. Part – B consists of 10 questions, two questions of 16 marks (with max. of 3 sub questions) from each module with internal choice. Students shall answer five full questions, selecting one full question from each module.
4. The laboratory component of the **integrated course** shall be CIE only. However, in SEE, the questions from the practical component shall be included.
5. The maximum marks from the practical component to be included in the SEE question paper is **16 marks.**
6. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping

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

Semester-III					
Operating Systems **Category: PCC** **(Common to AIML, CSD, ISE, CSE)**					
Course Code	:	**B25CS304**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	42:0:0:48	SEE	:	50 Marks
Total Hours/Semester	:	90	Total	:	100 Marks
Credits		3	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To introduce the fundamental concepts of operating systems, their functions, and types.
2.	To explain process management, scheduling algorithms, and inter-process communication.
3.	To describe memory management techniques, including paging, segmentation, and virtual memory.
4.	To illustrate file systems, storage management, and input-output operations.
5.	To explore synchronization mechanisms, deadlocks, and security aspects in operating systems.
6.	To provide insights into modern operating systems such as Linux, Windows, and mobile OS.

Module– 1	**No. of Hrs**
Introduction to Operating Systems: Definition, Purpose, and Evolution of Operating Systems, Types of Operating Systems: Batch, Time-sharing, Distributed, Real-Time, and Mobile OS, Operating System Structure: Monolithic, Layered, Microkernel, and Hybrid, System Calls and Operating System Services.	9
Module– 2	**No. of Hrs**
Process Management & CPU Scheduling Processes: Concept, Process Control Block (PCB), Process States, Threads: Single vs. Multithreading, User vs. Kernel Threads, CPU Scheduling: Scheduling Criteria, Preemptive & Nonpreemptive Scheduling, Scheduling Algorithms: FCFS, SJF, Priority Scheduling, Round Robin, Multi-Level Queue Scheduling, Inter process Communication (IPC) and Synchronization, Deadlocks: Detection, Prevention, Avoidance (Banker's Algorithm), Recovery.	9
Module– 3	**No. of Hrs**
Memory Management: Memory Allocation: Contiguous & Non-Contiguous Allocation, Paging and Segmentation, Virtual Memory: Demand Paging, Page Replacement Algorithms (FIFO, LRU, Optimal), Thrashing and Working Set Model.	8
Module– 4	**No. of Hrs**
File System & Storage Management File Concepts: File Attributes, File Types, File Access Methods, File System Structure: Directory Structure, File Allocation Methods (Contiguous, Linked, Indexed), Disk Scheduling Algorithms: FCFS, SSTF, SCAN, C-SCAN, LOOK, File Protection & Security.	8
Module– 5	**No. of Hrs**
I/O Systems, Security, and Case Studies: I/O Hardware, Device Drivers, Interrupt Handling, Security and Protection in Operating Systems, Authentication and Access Control Mechanisms, Case Study: UNIX/Linux, Windows, Android Operating System.	8

Course Outcomes: At the end of the course, the students will be able to	
CO1	Understand the basic structure, functionalities, and design principles of an operating system.
CO2	Analyze process scheduling techniques and inter-process communication mechanisms.
CO3	Implement memory management techniques such as paging, segmentation, and virtual memory.
CO4	Evaluate file system structures, disk scheduling algorithms, and storage management strategies
CO5	Apply synchronization techniques to avoid race conditions and deadlocks.

Text Books	
1.	Abraham Silberschatz, Peter B. Galvin, Greg Gagne- "Operating System Concepts, 10th Edition, Wiley, 2018"
2.	William Stallings – "Operating Systems: Internals and Design Principles, 9th Edition, Pearson, 2018".

Reference Books	
1	Andrew S. Tanenbaum – "Modern Operating Systems, 4th Edition, Pearson, 2015".
2	Dhananjay M.Dhamdhere–"Operating Systems: A Concept-Based Approach, 3rd Edition, McGraw-Hill, 2017"
3	Gary Nutt – "Operating Systems, 3rd Edition, Pearson, 2004"

ASSESSMENT STRUCTURE:
The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**40 (Average of Best two Assessments)**	**50**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
AAT	**Any Two Assessments as per regulations**	**20**	**10**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Grand Total				**100**

SEMESTER END EXAMINATION (SEE):

1. The Question paper for each course contains two parts, Part – A and Part – B.
2. Part – A consists of **Short Answer Questions** (2 Marks/1 mark) for 20 marks covering the complete syllabus and it is compulsory. Multiple Choice Questions are not allowed.
3. Part – B consists of 10 questions, two questions of 16 marks (with max. of 3 sub questions) from each module with internal choice. Students shall answer five full questions, selecting one full question from each module.

CO-PO Mapping

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

Semester-III					
C++ Programming **Category: PCC** **(Common to CSE, CSD)**					
Course Code	:	**B25CS305**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	42:0:0:48	SEE	:	50 Marks
Total Hours/Semester	:	90	Total	:	100 Marks
Credits	:	3	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To understand object-oriented programming using C++and Gain knowledge about the capability to store information together in an object.
2.	To illustrate the capability of a class to rely upon another class and functions.
3.	To Create and process data in files using file I/O functions.
4.	To understand the generic programming features of C++ including Exception handling.

Module– 1	**No. of Hrs**
An overview of C++: What is object-Oriented Programming? Introducing C++ Classes, The General Form of a C++ Program. Classes and Objects: Classes, Friend Functions, Friend Classes, Inline Functions, Parameterized Constructors, Static Class Members, When Constructors and Destructors are Executed, The Scope Resolution Operator, Passing Objects to functions, Returning Objects, Object Assignment	9
Module– 2	**No. of Hrs**
Arrays, Pointers, References, and the Dynamic Allocation Operators: Arrays of Objects, Pointers to Objects, The Pointer, Pointers to derived types, Pointers to class members. Functions Overloading, Copy Constructors: Functions Overloading, Overloading Constructor Functions. Copy Constructors, Default Function Arguments, Function Overloading and Ambiguity	9
Module– 3	**No. of Hrs**
Operator Overloading: Creating a Member Operator Function, Operator Overloading Using a Friend Function, overloading new and delete Inheritance: Base-Class Access Control, Inheritance and Protected Members, Inheriting Multiple Base Classes, Constructors, Destructors and Inheritance, Granting Access, Virtual Base Classes	8
Module– 4	**No. of Hrs**
Virtual Functions and Polymorphism: Virtual Functions, The Virtual Attribute is Inherited, Virtual Functions are Hierarchical, Pure Virtual Functions, Using Virtual Functions, Early v/s Late Binding. Templates: Generic Functions, Applying Generic Functions, Generic Classes. The type name and export Keywords. The Power of Templates	8
Module– 5	**No. of Hrs**
Exception Handling: Exception Handling Fundamentals, Handling Derived-Class Exceptions, Exception Handling Options, Applying Exception Handling. The C++ I/O System Basics: C++ Streams, The C++ Classes, Formatted I/O File I/O: and File Classes, Opening and Closing a File, Reading and Writing Text Files, Detecting EOF	8

Course Outcomes: At the end of the course, the students will be able to	
CO1	Illustrate the basic concepts of object-oriented programming.
CO2	Design appropriate classes for the given real-world scenario.
CO3	Use the knowledge of inheritance for developing optimized solutions
CO4	Apply the concepts of templates and exception handling for the given problem
CO5	Use the concepts of input output streams for file operations

Text Books	
1.	Herbert schildt, The Complete Reference C++, 4th edition, TMH, 2005
References	
1.	Allen B. Downey, “Think Stats: Exploratory Data Analysis in Python”, Green Tea Press, 2014.
2.	Bhave, “Object Oriented Programming With C++”, Pearson Education, 2004.
3.	A K Sharma, “Object Oriented Programming with C++”, Pearson Education, 2014

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment1	50	40 (Average of Best two Assessments)	50
	Internal Assessment2	50		
	Internal Assessment3	50		
AAT	Any Two Assessments as per Regulations	20	10	
SEE	Semester End Examination	100	50	50
Total Marks				100

SEMESTER END EXAMINATION (SEE):

1. The Question paper for each course contains two parts, Part – A and Part – B.
2. Part – A consists of **Short Answer Questions** (2 Marks/1 mark) for 20 marks covering the complete syllabus and it is compulsory. Multiple Choice Questions are not allowed.
3. Part – B consists of 10 questions, two questions of 16 marks (with max. of 3 sub questions) from each module with internal choice. Students shall answer five full questions, selecting one full question from each module.

CO-PO Mapping

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

Semester-III					
Data Structures and its Applications Lab **Category: PCCL** **(Common to CSE, CSD, ISE, AIML)**					
Course Code	:	**B25CSL306**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	0 : 0 : 28 : 2	SEE	:	50 Marks
Total Hours/Semester	:	30	Total	:	100 Marks
Credits	:	1	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To implement linear data structures such as arrays, stacks, queues, linkied lists.
2.	To apply suitable data structures for solving various computational problems effectively and efficiently.
3.	To implement nonlinear data structures such as trees, and graphs.
4.	To apply nonlinear data structure to provide the solution for the given problem.
5.	To familiarize students with file handling and explore applications of data structures in real-world scenarios.

Sl. No	**List of Experiments**
1.	Develop a Program in C for the following: a) Declare a calendar as an array of 7 elements (A dynamically Created array) to represent 7 days of a week. Each Element of the array is a structure having three fields. The first field is the name of the Day (A dynamically allocated String), The second field is the date of the Day (A integer), the third field is the description of the activity for a particular day (A dynamically allocated String). b) Write functions create(), read() and display(); to create the calendar, to read the data from the keyboard and to print weeks activity details report on screen.
2.	Develop a Program in C for the following operations on Strings. a. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP) b. Perform Pattern Matching Operation: Find and Replace all occurrences of PAT in STR with REP if PAT exists in STR. Report suitable messages in case PAT does not exist in STR Support the program with functions for each of the above operations. Don't use Built-in functions.
3.	Develop a menu driven Program in C for the following operations on STACK of Integers (Array Implementation of Stack with maximum size MAX) a. Push an Element on to Stack b. Pop an Element from Stack c. Demonstrate how Stack can be used to check Palindrome d. Demonstrate Overflow and Underflow situations on Stack e. Display the status of Stack f. Exit Support the program with appropriate functions for each of the above operations
4.	Develop a Program in C for converting an Infix Expression to Postfix Expression. Program should support for both parenthesized and free parenthesized expressions with the operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.
5.	Develop a Program in C for the following Stack Applications a. Evaluation of Suffix expression with single digit operands and operators: +, -, *, /, %,^ . b. Solving Tower of Hanoi problem with n disks

6.	Develop a menu driven Program in C for the following operations on Circular QUEUE of Characters (Array Implementation of Queue with maximum size MAX) a. Insert an Element on to Circular QUEUE b. Delete an Element from Circular QUEUE c. Demonstrate Overflow and Underflow situations on Circular QUEUE d. Display the status of Circular QUEUE e. Exit Support the program with appropriate functions for each of the above operations
7.	Develop a menu driven Program in C for the following operations on Singly Linked List (SLL) of Student Data with the fields: USN, Name, Programme, Sem, PhNo a. Create a SLL of N Students Data by using front insertion. b. Display the status of SLL and count the number of nodes in it c. Perform Insertion / Deletion at End of SLL d. Perform Insertion / Deletion at Front of SLL(Demonstration of stack) e. Exit
8.	Develop a menu driven Program in C for the following operations on Doubly Linked List (DLL) of Employee Data with the fields: SSN, Name, Dept, Designation,Sal, PhNo a. Create a DLL of N Employees Data by using end insertion. b. Display the status of DLL and count the number of nodes in it c. Perform Insertion and Deletion at End of DLL d. Perform Insertion and Deletion at Front of DLL e. Demonstrate how this DLL can be used as Double Ended Queue. f. Exit
9.	Develop a Program in C for the following operations on Singly Circular Linked List (SCLL) with header nodes a. Represent and Evaluate a Polynomial P(x,y,z) = 6x2y2z-4yz5+3x3yz+2xy5z-2xyz3 b. Find the sum of two polynomials POLY1(x,y,z) and POLY2(x,y,z) and store the result in POLYSUM(x,y,z) Support the program with appropriate functions for each of the above operations
10.	Develop a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers. a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2 b. Traverse the BST in Inorder, Preorder and Post Order c. Search the BST for a given element (KEY) and report the appropriate message d. Exit
11.	Develop a Program in C for the following operations on Graph(G) of Cities a. Create a Graph of N cities using Adjacency Matrix. b. Print all the nodes reachable from a given starting node in a digraph using DFS/BFS
12.	Given a File of N employee records with a set K of Keys (4-digit) which uniquely determine the records in file F. Assume that file F is maintained in memory by a Hash Table (HT) of m memory locations with L as the set of memory addresses (2-digit) of locations in HT. Let the keys in K and addresses in L are Integers. Develop a Program in C that uses Hash function H: K →L as H(K)=K mod m (remainder method), and implement hashing technique to map a given key K to the address space L. Resolve the collision (if any) using linear probing.

Course Outcomes: At the end of the course, the students will be able to	
CO1	Apply the concepts of pointers and structures in problem solving
CO2	Use different types of linked lists to solve problems
CO3	Demonstrate stack and queue data structures to solve problems.
CO4	Use the Binary search tree and graph data structures to solve problems.
CO5	Illustrate the operations performed on tree data structures, hash functions for problem solving.

ASSESSMENT STRUCTURE FOR LABORATORY:

- The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying 50% weightage.
- For both CIE and SEE, the student is required to conduct one experiment each from both Part A and Part B.
- To qualify and become eligible to appear for SEE, in the CIE component, a student must secure a minimum of 40% of 50 marks, i.e., 20 marks.
- In SEE component, Part A must be evaluating as 40% and Part B will be evaluating as 60%.
- To pass the SEE component, a student must secure a minimum of 35% of 50 marks, i.e., 18 marks. A student is deemed to have successfully completed the course if the combined total of CIE and SEE is at least 40 out of 100 marks.

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Laboratory	**Lab Conduction & Record**	**Evaluate each expt. for 10 marks.**	**15**	**50**
	Laboratory Test 1	**50**	**15**	
	Laboratory Test 2	**50**	**20**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

CO-PO Mapping

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

Level 3 - High, Level 2 - Moderate, Level 1 - Low

Semester-III					
C++ Programming Lab **Category: AEC** **(Common to CSE, CSD)**					
Course Code	:	**B25CSL307**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	0 : 0 : 28 : 2	SEE	:	50 Marks
Total Hours/Semester	:	30	Total	:	100 Marks
Credits	:	1	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To understand object-oriented programming using C++and Gain knowledge about the capability to store information together in an object.
2.	To illustrate the capability of a class to rely upon another class and functions.
3.	To Create and process data in files using file I/O functions.
4.	To understand the generic programming features of C++ including Exception handling.

Sl. No	**Name of the experiment**
1.	Develop a C++ program to find the largest of three numbers
2.	Develop a C++ program to sort the elements in ascending and descending order.
3.	Develop a C++ program using classes to display student name, roll number, marks obtained in two subjects and total score of student.
4.	Develop a C++ program for a bank empolyee to print name of the employee, account_no. & balance. Print invalid balance if amount.
5.	Develop a C++ program to create a class named shape. Create three sub classes namely: circle, triangle and square, each class has two member functions named draw () and erase (). Demonstrate virtual functions concepts by developing suitable methods, defining member data and main program.
6.	Develop a C++ program using Operator Overloading for overloading Unary minus operator.
7.	Develop a C++ program to implement Multiple inheritance for performing arithmetic operation of two numbers.
8.	Develop a C++ program using Constructor in Derived classes to initialize alpha, beta and gamma and display corresponding values.
9.	Develop a C++ program to create a text file, check file created or not, if created it will write some text into the file and then read the text from the file.
10.	Develop a C++ program to write and read time in/from binary file using fstream
11.	Develop a function which throws a division by zero exception and catch it in catch block. Write a C++ program to demonstrate usage of try, catch and throw to handle exception.
12.	Develop a C++ program that handles array out of bounds exception using C++

Course Outcomes: At the end of the course, the students will be able to	
CO1	Illustrate the basic concepts of object-oriented programming.
CO2	Design appropriate classes for the given real world scenario.
CO3	Use the knowledge of inheritance for developing optimized solutions
CO4	Apply the concepts of templates and exception handling for the given problem
CO5	Use the concepts of input output streams for file operations

ASSESSMENT STRUCTURE FOR LABORATORY:

- The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying 50% weightage.
- For both CIE and SEE, the student is required to conduct one experiment each from both Part A and PartB.
- To qualify and become eligible to appear for SEE, in the CIE component, a student must secure a minimum of 40% of 50 marks, i.e., 20 marks.
- In SEE component, Part A must be evaluating as 40% and Part B will be evaluating as 60%.
- To pass the SEE component, a student must secure a minimum of 35% of 50 marks, i.e., 18 marks. A student is deemed to have successfully completed the course if the combined total of CIE and SEE is at least 40 out of 100 marks.

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Laboratory	**Lab Conduction & Record**	**Evaluate each expt. for 10 marks**	**15**	**50**
	Laboratory Test 1	**50**	**15**	
	Laboratory Test 2	**50**	**20**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

CO-PO Mapping

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

Level 3 - High, Level 2 - Moderate, Level 1 – Low

Semester-IV					
Discrete Mathematical Structures **Category: ASC** **(Common to AIML, ISE, CSD, CSE)**					
Course Code	:	**B25MCS401**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	42:28:0:50	SEE	:	50 Marks
Total Hours/Semester	:	120	Total	:	100 Marks
Credits	:	4	SEE Duration	:	3Hrs

Course Learning Objective is	
1.	To enhance logical thinking and problem-solving abilities in students.
2.	To analyzingandsolvingcombinatorialproblemsusinglogicalreasoningandcreativity.
3.	To solve problems using counting techniques and combinatorics.
4.	To analyze and solve problems involving graph connectivity.
5.	To analyze and solve problems involving tree structures.

Module–1	**No. of hours**
Set Theory: Fundamentals of Set Theory, Set Operations, Laws of Set Theory, Counting And Venn Diagrams, Cartesian Product, Relations, Types of Relations, Functions, Types of Functions, Function of Composition, Inverse Functions. Mathematical Induction.	14
Module–2	**No. of hours**
Mathematical Logic: Logic And Counting: Fundamentals of Logic, Propositional Logic, Logical Connectives And Truth Tables, Logic Equivalence, Tautology And Contradiction, Contingency, Methods of Proof And Disproof.	14
Module–3	**No. of hours**
Elementary Combinatorics: Basis of counting, Combinations& Permutations, with repetitions, Constrained Repetitions, Binomial coefficients, Binomial Multinomial theorems, The principles of Inclusion–Exclusion, Pigeon-hole Principles and its applications.	14
Module–4	**No. of hours**
Graph Theory: Introduction, representing graphs, operations on graphs, Directed graphs, graph Isomorphism, Paths, Cycles, Euler Graph, Hamilton graph, Planar graphs.	14
Module–5	**No. of hours**
Trees and Lattices Theory: Introduction, applications of Trees, Spanning trees, Minimum Spanning trees, Prim's And Kruskul's Algorithms. **Lattices Theory:** Lattices and Algebraic systems, Principles of duality, basic properties.	14

Course Outcomes: At the end of the course, the students will be able to	
CO1	Apply concepts of set theory, relations, functions, and mathematical induction to solve Mathematical and engineering problems.
CO2	Apply counting techniques and mathematical logic to analyze and solve real-life problems.
CO3	Apply fundamental combinatorial methods to solve counting and arrangement problems.
CO4	Use basic concepts of graph theory to model and analyze theoretical and practical problems.
CO5	Analyze the problems involving tree structures in discrete mathematical contexts**.**

Text Books	
1.	RalphP. Grimaldi: Discrete and Combinatorial Mathematics, 5th Edition, Pearson Education.
2.	RichardBronson,Schaum'sOutlineofMatrixOperations,McGraw-Hillpublications,2nd Edition.

Reference Text Books	
1.	Gregory Hartman, Fundamentals of Matrix Algebra, 3rd Edition.
2.	GaryHaggard, JohnSchlipf, Discrete Mathematics for Computer Science, Thomson Books.

Web links and Video lectures(e-Resources)
• https://nptel.ac.in/courses/12286025 • VTUEDUSATPROGRAMME– 20 • http://www.class-central.com/subject/math(MOOCs)

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**40 (Average of Best two Assessments)**	**50**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
AAT	**Any Two Assessments as per regulations**	**20**	**10**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The Question paper for each course contains two parts, Part – A and Part – B.
2. Part – A consists of **Short Answer Questions** (2 Marks/1 mark) for 20 marks covering the complete syllabus and it is compulsory. Multiple Choice Questions are not allowed.
3. Part – B consists of 10 questions, two questions of 16 marks (with max. of 3 sub questions) from each module with internal choice. Students shall answer five full questions, selecting one full question from each module.

CO-PO Mapping:

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

Level3-High, Level2–Moderate, Level1 -Low

Semester-IV					
Database Management Systems **Category: PCC** **(Common to AIML, ISE, CSD, CSE, CSE(IC))**					
Course Code	:	**B25CS402**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	42:28:0:50	SEE	:	50 Marks
Total Hours/Semester	:	120	Total	:	100 Marks
Credits	:	4	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To introduce the fundamental concepts of database systems and data models, particularly the relational model.
2.	To understand and apply relational algebra and relational calculus for query formulation.
3.	To develop SQL queries for data definition, manipulation, and control.
4.	To understand the concepts of normalization and apply it to improve database design and eliminate anomalies.
5.	To gain knowledge of transaction processing, concurrency control, and recovery techniques to ensure data integrity and consistency.

Module– 1	**No. of Hrs**
Introduction to Databases: Introduction, Characteristics of database approach, Advantages of using the DBMS approach, History of database applications. **Overview of Database Languages and Architectures:** Data Models, Schemas, and Instances. Three schema architecture and data independence, database languages, and interfaces, The Database System environment. **Conceptual Data Modelling using Entities and Relationships:** Entity types, Entity sets, attributes and structural constraints, Weak entity types, ER diagrams.	14
Module– 2	**No. of Hrs**
Relational Model: Relational Model Concepts, Relational Model Constraints and relational database schemas, Update operations, transactions, and dealing with constraint violations. **Relational Algebra:** Unary and Binary relational operations, additional relational operations (aggregate, grouping) Examples of Queries in relational algebra. **Mapping Conceptual Design into a Logical Design:** Relational Database Design using ER-to Relational mapping.	14
Module– 3	**No. of Hrs**
SQL-Advance Queries: Views in SQL, Nested and co-related queries, Exists & Not Exists, joins & outer joins, specifying constraints as assertions and action Triggers, Check constraints. Comparisons using Null values & three-Valued logic. **SQL:** Specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL.	14
Module– 4	**No. of Hrs**
Normalization: Database Design Theory –Introduction to Normalization using Functional and Multivalued Dependencies: Informal design guidelines for relation schema, Functional Dependencies, Normal Forms based on Primary Keys, First, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form.	14
Module– 5	**No. of Hrs**
Transaction Processing - Introduction to Transaction Processing, Transaction and System concepts, Desirable properties of Transactions, Characterizing schedules based on recoverability, Characterizing schedules based on Serializability, Transaction support in SQL. **Concurrency Control in Databases-** Two-phase locking techniques for Concurrency control, Concurrency control based on Timestamp ordering, Multiversion Concurrency control techniques, Validation Concurrency control techniques.	14

Course Outcomes: At the end of the course, the students will be able to	
CO1	Design entity relationship for the given scenario.
CO2	Design the relational model and mapping conceptual design in to a logical design.
CO3	Apply various structured Query Language (SQL) statements for database manipulation.
CO4	Analyze various normalization forms for the given application.
CO5	Understand the concepts of transaction management and recovery mechanisms.

Text Books	
1.	Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
2.	Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill
3.	Elmasri and Navathe, —Fundamentals of Database Systems, 7th Edition, Pearson education, 2016
References Text Books	
1.	Abraham Silberschatz, Henry F. Korth and S. Sudarshan's Database System Concepts 6th Edition Tata Mcgraw Hill Education Private Limited
2.	G. K. Gupta —Database Management Systems, 3rd Edition, McGraw – Hill, 2018
3.	Dr. P.S. Deshpande, SQL and PL/SQL for Oracle 10g, Black Book, Dreamtech Press, 2012

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**40 (Average of Best two Assessments)**	**50**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
AAT	**Any Two Assessments as per regulations**	**20**	**10**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The Question paper for each course contains two parts, Part – A and Part – B.
2. Part – A consists of **Short Answer Questions** (2 Marks/1 mark) for 20 marks covering the complete syllabus and it is compulsory. Multiple Choice Questions are not allowed.
3. Part – B consists of 10 questions, two questions of 16 marks (with max. of 3 sub questions) from each module with internal choice. Students shall answer five full questions, selecting one full question from each module.

CO-PO Mapping

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

Semester-IV					
Design and Analysis of Algorithms **Category: IPCC** **(Common to- AIML, ISE, CSD, CSE, CSE(IC))**					
Course Code	:	**B25CS403**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	42:0:28:50	SEE	:	50 Marks
Total Hours/Semester	:	120	Total	:	100 Marks
Credits	:	4	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To learn the methods for analyzing algorithms and evaluating their performance.
2.	To demonstrate the efficiency of algorithms using asymptotic notations.
3.	To solve problems using various algorithm design methods, including brute force, greedy, divide and conquer, decrease and conquer, transform and conquer, dynamic programming, backtracking, and branch and bound.
4.	To learn the concepts of P and NP complexity classes.
5.	To apply algorithmic techniques in real-world problem solving and enhance logical and analytical thinking.

Module– 1	**No. of Hrs**
Introduction: What is an Algorithm? Fundamentals of Algorithmic Problem Solving. **Fundamentals of the analysis of algorithm efficiency:** Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non recursive Algorithms, Mathematical Analysis of Recursive Algorithms. **Brute force approaches:** Selection Sort and Bubble Sort, Sequential Search and Brute Force String Matching.	9
Module– 2	**No. of Hrs**
Brute force approaches: Exhaustive Search (Travelling Salesman problem and Knapsack Problem). **Decrease-and-conquer:** Insertion Sort, Topological Sorting. **Divide and conquer:** Merge Sort, Quick Sort, Binary Tree Traversals, Multiplication of Large Integers and Strassen's Matrix Multiplication.	9
Module– 3	**No. of Hrs**
Transform and Conquer: Balanced Search Trees, Heaps and Heapsort. **Space-time tradeoffs:** Sorting by Counting: Comparison counting sort. **Input Enhancement in String Matching:** Horspool's Algorithm.	8
Module– 4	**No. of Hrs**
Dynamic programming: Three basic examples, The Knapsack Problem and Memory Functions, Warshall's and Floyd's Algorithms. **The greedy method:** Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffman Trees and Codes	8
Module– 5	**No. of Hrs**
Limitations of algorithmic power: Decision Trees, P, NP, and NP-Complete Problems. **Coping with limitations of algorithmic power:** Backtracking (n-Queens problem, Subset-sum problem), Branch-and-Bound (Knapsack problem), Approximation algorithms for NPHard problems (Knapsack problem).	8

Course Outcomes: At the end of the course, the students will be able to	
CO1	Apply asymptotic notational method to analyze the performance of the algorithms in terms of time complexity.
CO2	Demonstrate divide & conquer approaches and decrease & conquer approaches to solve computational problems.
CO3	Make use of transform & conquer and dynamic programming design approaches to solve the given real world or complex computational problems.
CO4	Apply greedy and input enhancement methods to solve graph & string based computational problems.
CO5	Analyze various classes (P,NP and NP Complete) of problems.

LABORATORY

Sl. No	List of Experiments
1.	Design and implement C/C++ Program to sort a given set of n integer elements using Selection Sort method and compute its time complexity.
2.	Design and implement C/C++ Program to sort a given set of n integer elements using Quick Sort method and obtain its Time complexity
3.	Design and implement C/C++ Program to sort a given set of n integer elements using Merge Sort method and compute its time complexity.
4.	Design and implement C/C++ Program to obtain the Topological ordering of vertices in a given digraph
5.	Design and implement C/C++ Program to solve 0/1 Knapsack problem using Dynamic Programming method.
6.	a) Design and implement C/C++ Program to solve All-Pairs Shortest Paths problem using Floyd's algorithm. b) Design and implement C/C++ Program to find the transitive closure using Warshal's algorithm.
7.	Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Kruskal's algorithm.
8.	Design and implement C/C++ Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm
9.	Design and implement C/C++ Program to find shortest paths from a given vertex in a weighted connected graph to other vertices using Dijkstra's algorithm
10.	Design and implement C/C++ Program to find a subset of a given set S = {sl , s2,.....,sn} of n positive integers whose sum is equal to a given positive integer d.
11.	Design and implement C/C++ Program to solve discrete Knapsack and continuous Knapsack problems using greedy approximation method.
12.	Design and implement C/C++ Program for N Queen's problem using Backtracking.

Textbooks	
1.	Introduction to the Design and Analysis of Algorithms, By AnanyLevitin, 3rd Edition (Indian), 2017, Pearson

References Textbooks	
1.	Computer Algorithms/C++, Ellis Horowitz, SatrajSahni and Rajasekaran, 2nd Edition, 2014, Universities Pres
2.	Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, Clifford Stein, 3rd Edition, PHI.
3.	Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education)

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	Internal Assessment1	50	40 (Average of Best two Assessments)	50/2 = 25
	Internal Assessment2	50		
	Internal Assessment3	50		
AAT	Any Two Assessments as per regulations	20	10	
Laboratory	Lab Conduction & Record	Evaluating each expt. for 10 marks	10	25
	Lab Internal Test	50	15	
SEE	Semester End Examination	100	50	50
Grand Total				100

SEMESTER END EXAMINATION (SEE):

1. The Question paper for each course contains two parts, Part – A and Part – B.
2. Part – A consists of **Short Answer Questions** (2 Marks/1 mark) for 20 marks covering the complete syllabus and it is compulsory. Multiple Choice Questions are not allowed.
3. Part – B consists of 10 questions, two questions of 16 marks (with max. of 3 sub questions) from each module with internal choice. Students shall answer five full questions, selecting one full question from each module.
4. The laboratory component of the **integrated course** shall be CIE only. However, in SEE, the questions from the practical component shall be included.
5. The maximum marks from the practical component to be included in the SEE question paper is **16 marks.**
6. Question papers to be set as per the Blooms Taxonomy levels.

CO-PO Mapping

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

Semester IV					
Java Programming **Category: PCC** **(Common to – CSE,ISE,CSD)**					
Course Code	:	**B25CS404**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	42:0:0:48	SEE	:	50 Marks
Total Hours/Semester	:	90	Total	:	100 Marks
Credits		3	SEE Duration	:	3 Hrs

Course Learning Objective is	
1	To learn primitive constructs JAVA programming language.
2	To introduce the fundamentals of object-oriented programming concepts using Java, including classes, objects, inheritance, and polymorphism
3	To develop the ability to write reusable and maintainable code using key OOP principles like encapsulation and abstraction.
4	To familiarize students with Java syntax, control structures, and exception handling mechanisms
5	To provide knowledge of multithreading and file handling to build efficient and robust Java programs.

Module– 1	**No. of Hrs**
An Overview of Java: Object-Oriented Programming (Two Paradigms, Abstraction, The Three OOP Principles), Using Blocks of Code, Lexical Issues (Whitespace, Identifiers, Literals, Comments, Separators, The Java Keywords). Data Types, Variables, and Arrays: The Primitive Types (Integers, Floating-Point Types, Characters, Booleans), Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, Introducing Type Inference with Local Variables. Operators: Arithmetic Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The? Operator, Operator Precedence, Using Parentheses. Control Statements: Java's Selection Statements (if, The Traditional switch), Iteration Statements (while, do-while, for, The For-Each Version of the for Loop, Local Variable Type Inference in a for Loop, Nested Loops), Jump Statements (Using break, Using continue, return)	9
Module–2	**No. of Hrs**
Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection. Methods and Classes: Overloading Methods, Objects as Parameters, Argument Passing, Returning Objects, Recursion, Access Control, understanding static, introducing final, Introducing Nested and Inner Classes	9
Module–3	**No. of Hrs**
Inheritance: Inheritance Basics, using super, creating a Multilevel Hierarchy, When Constructors Are Executed, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, Local Variable Type Inference and Inheritance, The Object Class. Interfaces: Interfaces, Default Interface Methods, Use static Methods in an Interface, Private Interface Methods	8
Module– 4	**No. of Hrs**
Packages: Packages, Packages and Member Access, Importing Packages. Exceptions: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java's Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions	8
Module– 5	**No. of Hrs**
Multithreaded Programming: The Java Thread Model, The Main Thread, Creating a Thread, Creating Multiple Threads, UsingisAlive() and join(), Thread Priorities, Synchronization, Interthread Communication, Suspending, Resuming, and Stopping Threads, Obtaining a Thread's State. Enumerations, Type Wrappers and Autoboxing: Enumerations (Enumeration Fundamentals, Thevalues() and valueOf() Methods), Type Wrappers (Character, Boolean, The Numeric Type Wrappers), Autoboxing (Autoboxing and Methods, Autoboxing/Unboxing Occurs in Expressions, Autoboxing/Unboxing Boolean and Character Values).	8

Course Outcomes: At the end of the course, the students will be able to	
CO1	Demonstrate proficiency in writing simple programs involving branching and looping structures.
CO2	Design a class involving data members and methods for the given scenario.
CO3	Apply the concepts of inheritance and interfaces in solving real world problems
CO4	Use the concept of packages and exception handling in solving complex problem
CO5	Apply concepts of multithreading, autoboxing and enumerations in program development

Textbooks	
1.	Java: The Complete Reference, 12th Edition, by Herbert Schildt, November 2021, McGraw-Hill, ISBN: 9781260463422

Reference Textbooks	
1.	Programming with Java, 6th Edition, by E Balagurusamy, Mar-2019, McGraw Hill Education, ISBN: 9789353162337.
2.	Thinking in Java, Fourth Edition, by Bruce Eckel, Prentice Hall, 2006 (https://sd.blackball.lv/library/thinking_in_java_4th_edition.pdf)

ASSESSMENT STRUCTURE:

The assessment in each course is divided equally between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each carrying 50% weightage. To qualify and become eligible to appear for SEE, in the CIE, a student must score at least 40% of 50 marks, i.e., 20 marks. To pass the SEE, a student must score at least 35% of 50 marks, i.e., 18 marks. Notwithstanding the above, a student is considered to have passed the course, provided the combined total of CIE and SEE is at least 40 out of 100 marks.

CONTINUOUS INTERNAL EVALUATION (CIE):

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Theory	**Internal Assessment1**	**50**	**40 (Average of Best two Assessments)**	**50**
	Internal Assessment2	**50**		
	Internal Assessment3	**50**		
AAT	**Any Two Assessments as per regulations**	**20**	**10**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Total Marks				**100**

SEMESTER END EXAMINATION (SEE):

1. The Question paper for each course contains two parts, Part – A and Part – B.
2. Part – A consists of **Short Answer Questions** (2 Marks/1 mark) for 20 marks covering the complete syllabus and it is compulsory. Multiple Choice Questions are not allowed.
3. Part – B consists of 10 questions, two questions of 16 marks (with max. of 3 sub questions) from each module with internal choice. Students shall answer five full questions, selecting one full question from each module.

CO-PO Mapping

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

Semester-IV					
Database Management Systems Lab **Category: PCCL** **(Common to CSE, ISE, CSD, AIML)**					
Course Code	:	**B25CSL406**	CIE	:	50 Marks
Teaching Hours L : T : P : S	:	0 : 0 : 28: 2	SEE	:	50 Marks
Total Hours/Semester	:	30	Total	:	100 Marks
Credits	:	1	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To Provide a strong foundation in database concepts, technology, and practice.
2.	To Practice SQL programming through a variety of database problems.
3.	To Understand the relational database design principles.
4.	To Demonstrate the use of concurrency and transactions in database.
5.	To Design and build database applications for real world problems

Sl. No	**List of Experiments**
1.	Implementation of DDL commands of SQL with suitable examples • Create • Alter • Drop • Truncate • Rename
2.	Implementation of DML commands of SQL with suitable examples • Insert • Update • Delete • Select
3.	Implementation of different types of function with suitable examples • Aggregate Function • Using Parentheses • Column Aliases • Literal Character Strings • Duplicate Row
4.	Implementation of different types of operators in SQL • Arithmetic Operators • Concatenation Operator • Comparison Operator (=,>,>=,<,<=,<>) • Other Comparison Operator (BETWEEN, IN, LIKE, IS NULL) • Logical Operator (AND, OR, NOT)
5.	Implementation of different types of Joins • Inner Join • Outer Join i. Left outer join ii. Right outer join iii. Full outer Join • Natural Join
6.	Study and Implementation of • Group By & having clause • Order by clause • Sorting in Descending & Ascending order
7.	Study & Implementation of different types of constraints. • Primary Key & Foreign Key Constraints • NOT NULL Constraints • Default 1 Constraints • Check constraints

	• Domain Constraints • Entity Integrity Constraints • Referential Integrity Constraints
8.	Consider the following schema for a Library Database: BOOK(Book_id, Title, Publisher_Name, Pub_Year) BOOK_AUTHORS(Book_id, Author_Name) PUBLISHER(Name, Address, Phone) BOOK_COPIES(Book_id,Programme_id,No-of_Copies) BOOK_LENDING(Book_id,Programme_id,Card_No,Date_Out, Due_Date) LIBRARY_PROGRAMME(Programme_id, Programme_Name, Address) Write SQL queries to 1. Retrieve details of all books in the library–id, title, name of publisher, authors, number of copies in each Programme, etc. 2. Get the particulars of borrowers who have borrowed more than 3 books,but from Jan 2017 to Jun 2017. 3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation. 4. Partition the BOOK table base don year of publication .Demonstrate its working with a simple query. 5. Create a view of all books and its number of copies that are currently available in the Library.
9.	Consider the following schema for Order Database: SALESMAN(Salesman_id, Name, City, Commission) CUSTOMER(Customer_id, Cust_Name, City, Grade, Salesman_id) ORDERS(Ord_No,Purchase_Amt,Ord_Date,Customer_id,Salesman_id) Write SQL queries to 1. Count the customers with grades above Bangalore's average. 2. Find the name and numbers of all salesman who had more than one customer. 3. List all the salesman and indicate those who have and don't have customers in their cities (Use UNION operation.) 4. Create a view that finds the salesman who has the customer with the highest order of a day 5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.
10.	Consider the schema for Movie Database: ACTOR(Act_id, Act_Name, Act_Gender) DIRECTOR(Dir_id, Dir_Name, Dir_Phone) MOVIES(Mov_id,Mov_Title,Mov_Year,Mov_Lang, Dir_id) MOVIE_CAST(Act_id, Mov_id, Role) RATING(Mov_id,Rev_Stars) Write SQL queries to 1. List the titles of all movies directed by„Hitchcock" 2. Find the movie names where one or more actors acted in two or more movies. 3. List all actors who acted in a movie before 2000 and also in a movie after 2015 (use JOIN operation). 4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. 5. Update rating of all movies directed b 'Steven Spielber'to 5.

Course Outcomes: At the end of the course, the students will be able to	
CO1	Create, Update and query on the database.
CO2	Demonstrate the working of different concepts of DBMS.
CO3	Implement SQL commands for database schema creation and modification.
CO4	Apply Primary and Foreign Key constraints to enforce entity integrity and referential integrity.
CO5	Explain the purpose and functionality of different types of joins in relational databases.

ASSESSMENT STRUCTURE FOR LABORATORY:

- The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying 50% weightage.
- For both CIE and SEE, the student is required to conduct one experiment each from both Part A and Part B.
- To qualify and become eligible to appear for SEE, in the CIE component, a student must secure a minimum of 40% of 50 marks, i.e., 20 marks.
- In SEE component, Part A must be evaluating as 40% and Part B will be evaluating as 60%.
- To pass the SEE component, a student must secure a minimum of 35% of 50 marks, i.e., 18 marks. A student is deemed to have successfully completed the course if the combined total of CIE and SEE is at least 40 out of 100 marks.

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Laboratory	Lab Conduction & Record	Evaluate each expt. for 10 marks.	15	50
	Laboratory Test 1	50	15	
	Laboratory Test 2	50	20	
SEE	Semester End Examination	100	50	50
Total Marks				100

CO-PO Mapping

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

Level 3 - High, Level 2 - Moderate, Level 1 - Low

Semester IV					
Java Programming Lab **Category: AEC** **(Common to – CSE,ISE,CSD)**					
Course Code	:	**B25CSL407**	CIE	:	50 Marks
Teaching Hours L : T : P: S	:	0 : 0 : 28 : 2	SEE	:	50 Marks
Total Hours/Semester	:	30	Total	:	100 Marks
Credits		1	SEE Duration	:	3 Hrs

Course Learning Objective is	
1.	To learn primitive constructs JAVA programming language.
2.	To introduce the fundamentals of object-oriented programming concepts using Java, including classes, objects, inheritance, and polymorphism.
3.	To develop the ability to write reusable and maintainable code using key OOP principles like encapsulation and abstraction.
4.	To familiarize students with Java syntax, control structures, and exception handling mechanisms.
5.	To provide knowledge of multithreading and file handling to build efficient and robust Java programs.

Sl. No	**List of Experiments**
1	Develop a JAVA program to add TWO matrices of suitable order N (The value of N should be read from command line arguments).
2	Develop a stack class to hold a maximum of 10 integers with suitable methods. Develop a JAVA main method to illustrate Stack operations.
3	A class called Employee, which models an employee with an ID, name and salary, is designed as shown in the following class diagram. The method raiseSalary (percent) increases the salary by the given percentage. Develop the Employee class and suitable main method for demonstration.
4	A class called MyPoint, which models a 2D point with x and y coordinates, is designed as follows: • Two instance variables x (int) and y (int). • A default (or "no-arg") constructor that construct a point at the default location of (0, 0). •A overloaded constructor that constructs a point with the given x and y coordinates. • A method setXY() to set both x and y. • A method getXY() which returns the x and y in a 2-element int array. • A toString() method that returns a string description of the instance in the format "(x, y)". • A method called distance(int x, int y) that returns the distance from this point to another point given (x, y) coordinates • An overloaded distance(MyPoint another) that returns the distance from this point to the give MyPoint instance (called another) • Another overloaded distance() method that returns the distance from this point to the origin (0,0) Develop the code for the class MyPoint. Also develop a JAVA program (called TestMyPoint) to test all the methods defined in the class.
5	Develop a JAVA program to create a class named shape. Create three sub classes namely: circle, triangle and square, each class has two member functions named draw () and erase (). Demonstrate polymorphism concepts by developing suitable methods, defining member data and main program.
6	Develop a JAVA program to create an abstract class Shape with abstract methods calculate Area() and Calculate Perimeter(). Create subclasses Circle and Triangle that extend the Shape class and implement the respective methods to calculate the area and perimeter of each shape.
7	Develop a JAVA program to create an interface Resizable with methods resize Width (int width) and Resize Height(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods.
8	Develop a JAVA program to create an outer class with a function display. Create another class inside the outer class named inner with a function called display and call the two functions in the main class.
9	Develop a JAVA program to raise a custom exception (user defined exception) for Division By Zero using try, catch, throw and finally.
10	Develop a JAVA program to create a package named mypack and import & implement it in a suitable class.
11	Write a program to illustrate creation of threads using runnable class. (start method start each of the newly created thread. Inside the run method there is sleep() for suspend the thread for 500

	milliseconds)
12	Develop a program to create a class My Thread in this class a constructor, call the base class constructor, using super and start the thread. The run method of the class starts after this. It can be observed that both main thread and created child thread are executed concurrently

Course Outcomes: At the end of the course, the students will be able to	
CO1	Demonstrate proficiency in writing simple programs involving branching and looping structures.
CO2	Design a class involving data members and methods for the given scenario
CO3	Apply the concepts of inheritance and interfaces in solving real world problems.
CO4	Use the concept of packages and exception handling in solving complex problem
CO5	Apply concepts of multithreading, autoboxing and enumerations in program development

ASSESSMENT STRUCTURE FOR LABORATORY:

- The assessment for each course is equally divided between Continuous Internal Evaluation (CIE) and the Semester End Examination (SEE), with each component carrying 50% weightage.
- For both CIE and SEE, the student is required to conduct one experiment each from both Part A and Part B.
- To qualify and become eligible to appear for SEE, in the CIE component, a student must secure a minimum of 40% of 50 marks, i.e., 20 marks.
- In SEE component, Part A must be evaluating as 40% and Part B will be evaluating as 60%.
- To pass the SEE component, a student must secure a minimum of 35% of 50 marks, i.e., 18 marks. A student is deemed to have successfully completed the course if the combined total of CIE and SEE is at least 40 out of 100 marks.

CIE	Type of Assessment	Max. Marks	Max. Marks Scaling Down to	Total Marks
Laboratory	**Lab Conduction & Record**	**Evaluate each expt. for 10 marks.**	**15**	**50**
	Laboratory Test 1	**50**	**15**	
	Laboratory Test 2	**50**	**20**	
SEE	**Semester End Examination**	**100**	**50**	**50**
Grand Total				**100**

CO-PO Mapping

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

Level 3- High, Level 2- Moderate, Level 1-Low