Faculty of Engineering and Technology (Co-Ed)

Department of Computer Science and Engineering

B. Tech 2nd year (III and IV Semester)

Scheme of Teaching and Examination

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

(Effective from the academic year 2024-2025)

Vision and Mission of Faculty of Engineering and Technology (Co-Ed)

VISION OF FACULTY OF ENGINEERING AND TECHNOLOGY(Co-Ed)

To be a premier technological institution that contribute for sustainable development of our nation & the world at large through achieving excellence in technical education and research which facilitating transformation of students into socially responsible citizens and competent professionals of the highest quality.

MISSION OF FACULTY OF ENGINEERING AND TECHNOLOGY(Co-Ed)

- Provide the affordable and quality education and achieve excellence in teaching learning by designing industry need based curriculum.
- Create good research environment that produces innovations and nurture research scholars.
- Collaborate with industries and other institutions of excellence in order to exchange of expertise.
- To inculcate the significance of human values based on the concept of Dasoha Philosophy of Lord Sharnbasveshwara i.e , "service to Humanity in Service to God" and professional ethics to serve the society.

VISION OF DEPARTMENT

To be recognized globally as a department of computer science and engineering focusing on social issues, embracing new technologies, providing highly talented technocrats and entrepreneurs with sound knowledge in ethics occupying top positions and are adaptable and sustainable in ever changing technological realm. To build a strong research and teaching environment par with the latest needs.

MISSION OF DEPARTMENT

M1: To impart quality technical education by designing curriculum in collaboration with industry requirements
M2: To transform young talents into highly competent individuals who work well in a team or as a single.
M3: To train the computer science Engineering graduates to cater to the needs of society and solve real-world problems by providing strong foundation.
M4: To develop a strong, inter and multi-research culture in the department by collaborating with other department of the university.

PROGRAM EDUCATIONAL OBJECTIVES (PEO'S)

PEO 1	Apply basic knowledge, principles and skills in the field of Computer Science to meet the job specification. (Knowledge / Practical Skills)
PEO 2	Implement the responsibility for solving problems analytically, critically, effective, innovative and market- oriented. (Critical Thinking and Problem Solving / Life-long Learning and Information Management / Entrepreneurship Skills/Researcher)
PEO 3	Acts effectively as an individual or in a group to convey information within the organization and community. (Team Working Skills / Communication Skills)
PEO 4	Practicing good values and ethics in a professional manner in the community and able to act as a leader. (Professional, Social, Ethics, and Humanity / Leadership Skills)

PROGRAM OUTCOMES (PO'S)

PO1: Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals and computing to solve Computer Science and Engineering related problems.

PO2: Problem Analysis : Identify, formulate , Research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO3: Design / Development of Solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for public health and safety, and cultural ,societal

PO4: Conduct Investigations of Complex Problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage :Create, select and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities related to Computer Science and Engineering with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and Team Work: Function effectively as an individual and as a member or leader to diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective report and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project Management and Finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-Long Learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOME(PSO'S)

Graduates of the Computer Science and Engineering program will be able to

PSO 1: Understand and recognize the fundamental concepts in basic science, humanities and programming languages like C/C++/java etc. to solve engineering problems.

PSO 2: Design, develop, apply concepts from diverse fields , analyse various computer science engineering design and management principles, mathematical foundations, sustainability and emerging challenges in the computation domain for effective computational solutions for real-life and research problems.

PSO 3: Apply modern programming languages, frameworks, and software tools in engineering and emerging trends principles to develop viable solutions for Information Technology Enabled Services and diverse fields.

Sharnbasva University, Kalaburagi

Scheme of Teaching and Examination2021-22

[As per Nep, Outcome Based Education (OBE) and Choice Based Credit System (CBCS) Scheme]

(Effective from the academic year 2021-22)

Programme: B. Tech: Computer Science and Engineering

III SEMESTER

Sl.No.	Course Code		Course Title	Teaching Department	Teaching Hours/week			Examination				Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P					
1	BS	21MAT31	Engineering Mathematics -III	Mathematics	3			3	50	50	100	03
2	PCC	21CS32	Logic Design and Computer Architecture	CSE	3		2	3	50	50	100	04
3	PCC	21CS33	Data Structures and its Applications using C	CSE	3			3	50	50	100	03
4	PCC	21CS34	Design and Analysis of Algorithms	CSE	3			3	50	50	100	03
5	PCC	21CS35	Object-Oriented Programming with Java	CSE	2			3	50	50	100	02
6	PCC	21CSL36	Data Structures and its Applications using C Lab	CSE			2	3	50	50	100	01
7	PCC	21CSL37	Design and Analysis of Algorithms Lab	CSE			2	3	50	50	100	01
8	PCC	21CSL38	Object-Oriented Programming with Java Lab	CSE			2	3	50	50	100	01
9	PW	21PRJ39	Project-III	CSE			2	3	50	50	100	01
10	HSS	21KANKK310 /21KANAK310	Kannada Kali-III/Ayda Kategalu	Humanities	1			2	50	50	100	01

11	AEC	21AEC311X	Ability Enhancement Course-III	CSE			2	3	50	50	100	01
Total					**15**	**0**	**12**	**32**	**550**	**550**	**1100**	**21**
Note: BS-Basic Science, PCC- Programme Core Course, PW-Project Work, AEC- Ability Enhancement Course, HSS-Humanity and Social Science, NCMC-Non-Credit Mandatory Course												
21KANKK310 Kannada Kali-III is form on Kannada speaking, reading and writing students and 21KANAK310 Ayda Kategalu is for the students who speak, readand write Kannada. Project (PRJ): Based on the ability/abilities of the student/s and recommendations of the mentor, a single discipline or multidisciplinary mini project can beAssigned to an individual student or to a group having not more than 4 students.												

Ability Enhancement Course-III												
Course code under 21AEC311X			**Course Title**									
21AEC3111			Unix and Shell Programming									
21AEC3112			Scilab									
Courses prescribed to lateral entry Diploma holders admitted to III semester of Engineering programs												
12	NCMC	21MATDIP31	Additional Mathematics– I	Mathematics	3	0	-	00	100	00	100	00

1) Non-Credit Mandatory Courses (NCMC) Additional Mathematics-I and II prescribed for III and IV semesters respectively, to the lateral entry Diploma holders admitted to III semester of B. Tech. programs, shall attend the classes during the respective semesters to complete all the formalities of the course and appear for the university examination. In case any student fails to secure the minimum 50% of the prescribed CIE marks, he/she shall be deemed to have secured F grade. In such a case, the students have to fulfill the requirements during subsequent semester/s.
2) These courses shall not be mandatory for vertical progression, but completion of the courses shall be mandatory for the award of degree.

Courses prescribed to lateral entry B.Sc. degree holders admitted to III semester of Engineering programs

Lateral entry students from B.Sc. stream, shall clear the noncredit courses Computer Aided Engineering Drawing, Elements of Civil Engineering of First Year Engineering Programme. These Courses shall not be considered for vertical progression, but completion of the courses shall be mandatory For the award of degree.

AICTE Activity Points to be earned by students admitted to B.Tech. programme (For more details refer to Chapter 6, AICTE Activity PointProgramme, Model Internship Guidelines):

Over and above the academic grades, every regular student admitted to the 4 years Degree programme and every student entering 4 years Degree programme through lateral entry, shall earn 100 and 75 Activity points respectively for the award of degree through AICTE Activity Point Programme. Students transferred from other universities to fifth semester are required to earn 50 activity points from the year of entry to Sharnbasva University. The Activity Points earned shall be reflected on the students eighth semester Grade card.

The activities can be spread over the years, any time during the semester week end holidays, as per the liking and convenience of the student from the year of entry to the programme. However, minimum hours requirement should be fulfilled. Activity Points (noncredit) have no effect on SGPA/CGPA and shall not be considered for vertical progression.

In case students fail to earn the prescribed activity points, eighth semester Grade Card shall be issued only after earning the required activity points. Student shall be admitted for the award of the degree only after the release of the Eighth semester Grade Card.

Sharnbasva University, Kalaburagi
Scheme of Teaching and Examination 2021-22
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
(Effective from the academic year 2021-22)

Programme: B. Tech: Computer Science and Engineering

IV SEMESTER

Sl.No.	Course Code		Course Title	Teaching Department	Teaching Hours/ week			Examination				Credits
					Theory Lecture	Tutorial	Practical/ Drawing	Duration in Hours	CIE Marks	SEE Marks	Total Marks	
					L	T	P					
1	BS	21MAT41	Engineering Mathematics-IV	Mathematics	3			3	50	50	100	03
2	PCC	21CS42	Operating System	CSE	3			3	50	50	100	03
3	PCC	21CS43	Python Programming	CSE	3			3	50	50	100	03
4	PCC	21CS44	Database Management Systems	CSE	3			3	50	50	100	03
5	BS	21CS45	Automata Theory and Computability	CSE	2	1		3	50	50	100	03
6	PCC	21CSL46	Operating System Lab	CSE			2	3	50	50	100	01
7	PCC	21CSL47	Python Programming Lab	CSE			2	3	50	50	100	01
8	PCC	21CSL48	Database Management Systems Lab	CSE			2	3	50	50	100	01
9	PW	21PRJ49	Project-IV	CSE			2	3	50	50	100	01
10	HSS	21KANKK410 /21 KANMD410	Kannada Kali-IV/ Mahadasohigalu	Humanities	1			2	50	50	100	01
11	AEC	21AEC411X	Ability Enhancement Course-IV				2	3	50	50	100	01
Total					**15**	**1**	**10**	**32**	**550**	**550**	**1100**	**21**

Note: BS-Basic Science, PCC- Program Core Course, PW-Project Work, AEC- Ability Enhancement Course, HSS-Humanity and Social Science, NCMC-Non-Credit Mandatory Course

21KANKK410 Kannada Kali-IV is for non-Kannada speaking, reading and writing students and 21KANMD410 Mahadasohigalu is for the students who speak, read and write Kannada.

Project(PRJ): Based on the ability /abilities of the student/s and recommendations of the mentor, a single discipline or multidisciplinary mini project can be assigned to an individual students or to a group having not more than 4 students.

Ability Enhancement Course-IV

Course code under 21AEC411X	**Course Title**
21AEC4111	Scripting Languages
21AEC4112	XML Programming

Courses prescribed to lateral entry Diploma holders admitted to III semester of Engineering programs

10	NCMC	21MATDIP41	Additional Mathematics – II	Mathematics	3	1	-	3	00	100	100	00

3) Non Credit Mandatory Courses (NCMC) Additional Mathematics-I and II prescribed for III and IV semesters respectively, to the lateral entry Diploma holders admitted to III semester of B. Tech. programs, shall attend the classes during the respective semesters to complete all the formalities of the course and appear for the university examination. In case any student fails to register for the said course/fails to secure the minimum 50% of the prescribed CIE marks, he/she shall be deemed to have secured F grade. In such a case, the students have to fulfill the requirements during subsequent semester/s to appear for SEE.

4) These courses shall not be mandatory for vertical progression, but completion of the courses shall be mandatory for the award of degree.

Courses prescribed to lateral entry B.Sc. degree holders admitted to III semester of Engineering programs

Lateral entry students from B.Sc. stream, shall clear the non-credit courses Computer Aided Engineering Drawing, Elements of Civil Engineering of First Year Engineering Program. These Courses shall not be considered for vertical progression, but completion of the courses shall be mandatory for the award of degree.

AICTE Activity Points to be earned by students admitted to B.Tech. program (For more details refer to Chapter 6,AICTE Activity Point Program, Model Internship Guidelines):

Over and above the academic grades, every regular student admitted to the 4 years Degree program and every student entering 4 years Degree program through lateral entry, shall earn 100 and 75 Activity points respectively for the award of degree through AICTE Activity Point Program. Students transferred from other universities to fifth semester are required to earn 50 activity points from the year of entry to Sharnbasva University. The Activity Points earned shall be reflected on the students eighth semester Grade card.

The activities can be spread over the years, anytime during the semester weekends holidays, as per the liking and convenience of the student from the year of entry to the program. However, minimum hours requirement should be fulfilled. Activity Points (noncredit) have no effect on SGPA/CGPA and shall not be considered for vertical progression.

In case students fail to earn the prescribed activity points, eighth semester Grade Card shall be issued only after earning the required activity points. Student shall be admitted for the award of the degree only after the release of the Eighth semester Grade Card.

ENGINEERING MATHEMATICS –III
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2021-2022)
SEMESTER –III

Course Code	**21MAT31**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**03**	**SEE Marks**	**50**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS – 03

Course Objectives:

1. Introduce most commonly used analytical and numerical methods in the different engineering fields.
2. Learn Laplace transform and Z-transforms to solve ODE and PDE's.
3. Understanding the statistical methods, numerical methods.
4. Solve the problem related to Interpolation.
5. To discuss the random variable and associated probability distributions.
6. Understand the vector space and associated results.
7. Understand the basic concepts of set theory, relations, functions and mathematical logic.

Modules	**Hours**
Module I	
Laplace Transforms: Definition, Laplace transforms of Elementary functions, properties (without proof) periodic function, Unit step function, Unit impulse function. **Inverse Laplace Transforms:** Definition, Convolution Theorem (without proof) and Finding Inverse Laplace transform by convolution Theorem. Solution of Linear Differential equations using Laplace Transforms and Applications	**08**
Module II	
Probability Distribution: Random variables (discrete and continuous) probability mass/density functions. Binomial distribution, Poisson distribution. Exponential and Normal distributions. Problems.	**08**
Module III	
Statistical Methods: Correlation-karl Pearson's co-efficient of correlation problems. Regression analysis lines of regression, Rank correlation (without proof)-problems. **Curve Fitting:** Curve fitting by the method of least square. Fitting of the curves of the form□ = □□ +□, □ = □□^{2} + □□ + □ & □ = □□$^{□□}$. **Numerical Methods:** Numerical solution of algebraic transcendental equations by Regula-Falsi Method and Newton-Raphson method.	**08**
Module IV	

Finite Difference: Forward and Backward differences, Newton's forward and backward interpolation formulae. Divided difference-Newton's divided difference formulae. Lagrange's-interpolation formula and inverse interpolation formula (all formula without proof) problems. **Numerical Integration:** Simpsons $(\frac{1}{3})^{rd}$, $(\frac{3}{8})^{th}$ rules, Weddle's rule (without proof) problems	**08**
Module V	
Z- Transforms: Difference Equations, Basic definitions, Damping rule, Shifting rule, Initial and Final Value theorems (without proof) and problems. Inverse Z-transforms. Applications of Z-transforms to solve difference equation. **Linear Algebra:** Introduction to Vector space and sub space, definitions, illustrative examples and simple problems, Basis and dimensions, Linear independent and linear dependent vectors	**08**

Question Paper Pattern:

- The SEE question paper will be set for 100 marks and the marks scored will be proportionately reduced to 50.
- The question paper will have ten full questions carrying equal marks.
- Each full question carries 20 marks.
- There will be two full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer five full questions, selecting one full question from each module.

Text Books:

1. B.S. Grewal: Higher Engineering Mathematics, Khanna Publishers, 43rd Ed., 2015.
2. E. Kreyszig: Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed.(Reprint), 2016.

Reference Books:

1. C.Ray Wylie, Louis C.Barrett : "Advanced Engineering Mathematics", 6th Edition,
2. McGraw-Hill Book Co., New York, 1995.
3. James Stewart: "Calculus –Early Transcendentals", Cengage Learning India Private Ltd., 2017.
4. B.V.Ramana: "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
5. Srimanta Pal & Subobh C Bhunia: "Engineering Mathematics", Oxford UniversityPress, 3rd Reprint, 2016.
6. Gupta C.B., Singh S.R. and Mukesh Kumar: "Engineering Mathematics for Semester I & II", Mc-Graw Hill Education (India) Pvt. Ltd., 2015.

Web links and Video Lectures:

1. http://nptel.ac.in/courses.php?disciplineID=111
2. http://www.class-central.com/subject/math
3. http://academicearth.org.

CO#	Course Outcomes
CO1	Apply the knowledge of Laplace transform from time domain to frequency domain in Signal and image processing and to find inverse Laplace transform.
CO2	Learn to solve the random variable in both discrete and continuous and their probability distribution, Mass on various engineering problems.
CO3	Make the use ot the concept of correlation and regression lines for solving the problems and numerical techniques to solve engineering problems.
CO4	Understanding the concepts of Finite differences to solve the problems on interpolation.
CO5	Understand the concept of relations , functions and to learn the law of logical Equivalence and Implications.

Logic Design and Computer Architecture **[As per Choice Based Credit System (CBCS) scheme]** **(Effective from the academic year 2022-2023)** **SEMESTER –III**			
Course Code	**21CS32**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**04**	**SEE Marks**	**50**
Total Number of Lecture Hours	**50**	**Exam Hours**	**03**
CREDITS – 04			

Course Objectives:

1. Describe, Illustrate and Analyze Combinational Logic circuits, Simplification of Algebraic equations using Karnaugh Maps and Quine McClusky Techniques.
2. Describe, Design and Analyze Encoders, Digital Multiplexers, Master-Slave Flip-Flops, Synchronous and Asynchronous Sequential.
3. How Computer System works & the Basic Principles, Instruction Level Architecture and Instruction Execution.
4. The Current state of art in Memory system design How I/O devices are accessed and its principles.
5. To provide the knowledge on Instruction Level Parallelism and Understand Concepts of advanced pipelining techniques, Computer Arithmetic and parallel Processing.

Modules	**Hours**
Module I	
Basic of Gates: Review of Basic Logic Gates, Positive and Negative Logic, Combinational Logic Circuits :Sum-of-Products Method, Truth Table to Karnaugh Map, Pairs Quads, and Octets, Karnaugh Simplifications, Don't-care Conditions, Product-of-sums Method, Product-of-sums simplifications, Simplification by Quine-McClusky Method, Data-Processing Circuits: Multiplexers, Demultiplexers, 1-of-16 Decoder, Encoders, Exclusive-or Gates, Parity Generators and Checkers.	**10**
DEMONSTRATION:1) Design and implement Half adder, Full Adder, Half Substractor, Full Substractor using basic gates. 2) Given a 4-Variable logic expression, simplify it using Entered Variable Map and realize the simplified logic expression using 8:1 multiplexer IC.	

Module II	
Flip-Flops: RS Flip-Flops Gated Flip-Flops, Edge-Triggered Flip-Flops, Flip-Flop Timings, JK Master-Slave Flip-Flop, and Various Representations of Flip-flops. **Registers:** Types of Registers, Application of Shift Registers. **Counters:** Asynchronous Counters, Synchronous Counters, Decade Counters, Counter Design as a Synthesis problem, A Digital Clock. **Design of Synchronous Sequential Circuit:** Model Selection, State Transition Diagram, State Synthesis Table, Design Equations and Circuit Diagram, State Reduction Technique.	**10**
DEMONSTRATION: 1) Realize a J-K Master/Slave Flip-Flop using NAND gates and verify its truth table. 2)Design and implement code converter I) Binary to Grey II) Grey to Binary Code using Basic Gates	
Module III	
Basic Structure of Computers: Functional Units, Basic Operational Concepts, Bus Structures, Performance. Machine instructions and Programs: Memory locations and addresses, Memory operations, Instructions and instruction sequencing, Addressing modes, Assembly language, basic input output operations. **Input output Organization**: Accessing I/O devices, Interrupts, DMA.	**10**
Module IV	
Memory Hierarchy: Introduction, Cache Performance, Six basic cache optimization, Virtual memory, And Memory hierarchy design: 10 advanced optimizations of cache performances	10
Module V	
Basic Processing Unit: Single bus organization, Multiple bus organization, Hardwired and micro-programmed design approaches. **Pipelining**: Introduction, Major hurdles of Pipelining, How is pipelining implemented?, Instruction level Parallelism: Concepts and Challenges.	**10**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Carl Hamacher, Z. Vranesic & S.Zaky, "Computer Organization", 5th Edition, Tata McGrawHill Publishing Company Ltd. New Delhi, 2002.

2. John L. Hennessy and David A. Patterson, Computer Architecture: A quantitative approach, 5th edition, Morgan Kaufmann Elseveir, 2013

3. Donald P Leach, Albert Paul Malvino & Goutham saha: Digital Principles and Applications,

8th Edition, Tata Mcgraw Hill, 2015.

Reference Books:

1. Morris Mano, "Computer System Architecture", PHI, 19862.William Stallings Computer Organization & Architecture, 7th Edition, PHI 2006.
2. Kai Hwang and Naresh Jotwani, Advanced Computer Architecture (SIE): Parallelism, Scalability, Programmability, McGraw Hill Education 3/e. 2015.
3. R D Sudhakar Samuel: IIustrative approach to Logic Design, Sanguine-Pearson, 2010.

CO#	COURSE OUTCOMES
CO1	Design and Develop digital Logic circuits using Boolean algebra and logic gates.
CO2	Analyzing and Implementing the test combinational and sequential logic circuits using HDLs like Verilog or VHDL.
CO3	Identify basic structure of computer and its performance measures.
CO4	Demonstrate memory hierarchy and virtual memory management, interrupt handling and DMA.
CO5	Optimize and evaluate computer system performance using various benchmarks and metrics.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	-	-	-	-	-	-	2	3	-
CO2	3	3	1	-	-	-	-	-	-	-	-	-	2	3	-
CO3	3	3	2	-	-	-	-	-	-	-	-	-	2	3	-
CO4	3	1	3	-	-	-	-	-	-	-	-	-	2	3	-
CO5	3	3	2	-	-	-	-	-	-	-	-	-	2	3	-

Data Structures and its Applications using C
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER – III

Course Code	**21CS33**	**CIE Marks**	**50**
Number Lecture Hour/Week	**03**	**SEE Marks**	**50**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS-03

Course Objectives:

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental datastructures

Modules	**Hours**
Module I	
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.	**08**
Module II	
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, Fibonacci Sequence, Tower of Hanoi, Ackerman's function. **Queues:** Definition, Array Representation. Queue Operations, Circular Queues,Dequeues, Priority Queues. Multiple Stack and Queues. Programming Examples	**08**
Module III	
Linked Lists: Definition, Representation of linked lists in Memory, Memory allocation; Garbage collection, Linked list operations: Traversing, Searching, Insertion, and Deletion. Doubly Linked lists and header linked lists. Linked Stacks and Queues. Applications of Linked lists – Polynomials, Sparse matrix representation. ProgrammingExamples	**08**
Module IV	

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-copying binary tree, testing equality. Threaded binary trees, Binary Search Trees – Definition, Insertion, Deletion, Traversal, Searching, Application of Trees-Evaluation of Expression, Programming Examples	**08**
Module V	
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, selection sort. **Hashing**: Hash Table organizations, Hashing Functions.	**08**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from eachmodule.

Text Books:

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Ed, Universities Press, 2014.
2. Seymour Lipschutz, Data Structures Schaum's Outlines, Revised 1st Ed, McGrawHill, 2014.

Reference Books:

1. Gilberg & Forouzan, Data Structures: A Pseudo-code approach with C, 2nd Ed, 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, 2 nd Ed, McGraw Hill, 2013.
4. A M Tenenbaum, Data Structures using C, PHI, 1989.
5. Robert Kruse, Data Structures and Program Design in C, 2nd Ed, PHI, 1996.

CO#	COURSE OUTCOMES
CO1	Implementation of Dynamic memory allocation functions and use of pointers
CO2	Analysis the use of different types of data structures, operations and algorithms
CO3	Applying the use stack, Queue, Lists, Trees and Graphs in problem solving
CO4	Implement all data structures in a high-level language for problem solving.
CO5	Analysis the use of graph structures, hash functions and collision resolution

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	1	2	-	-	-	-	-	-	-	-	2	2	3	-
CO2	3	3	2	-	-	-	-	-	-	-	-	2	2	3	-
CO3	3	3	2	-	-	-	-	-	-	-	-	2	2	3	-
CO4	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO5	3	3	1	-	-	-	-	-	-	-	-	2	2	3	-

Design and Analysis of Algorithms
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER – III

Course Code	**21CS34**	**CIE Marks**	**50**
Number Lecture Hours/Week	**03**	**SEE Marks**	**50**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS-03

Course Objectives:

1. Explain various computational problem solving techniques.
2. Apply appropriate method to solve a given problem.
3. Describe various methods of algorithm analysis.

Modules	**Hours**
Module I	
Introduction: What is Algorithm? Notation of Algorithm, Asymptotic Notations: Big-Oh notation (O), Omega notation (Ω), Theta notation (Θ) with examples, Mathematical analysis of non-Recursive and recursive Algorithms with Examples. **Brute force design technique:** Selection Sort, Bubble Sort, Sequential Search and Brute Force String Matching. Advantages and Disadvantages of Brute force design.	**08**
Module II	
Divide and Conquer: General method, Recurrence equation for divide and conquer, solving it using Master's theorem, Merge sort, Quick sort, Binary search, Strassen's matrix multiplication, Advantages and Disadvantages of divide and conquer. **Decrease and Conquer Approach:** Introduction, Insertion Sort, Depth-First Search and Breadth-First Search, Topological Sorting.	**08**
Module III	
Space-Time Tradeoffs: Introduction, Sorting by Counting, Input Enhancement in String Matching Harspool's algorithm. **Dynamic Programming: Transitive Closure:** Warshall's Algorithm**.** All Pairs Shortest Paths: Floyd's Algorithm.**Transform and Conquer Approach:** Introduction, Heaps and Heap Sort.	**08**
Module IV	
Greedy Method: Minimum cost spanning trees: Prim's Algorithm, Kruskal's Algorithm with performance analysis. **Single source shortest paths:** Dijkstra's Algorithm. **Optimal Tree problem:** Huffman Trees and Codes.**Backtracking:** General method, N-Queens problem,	**08**
Module V	

Backtracking: Sum of subsets problem, Hamiltonian cycles. **Branch and Bound:** Assignment Problem, Travelling Sales Person problem, 0/1 Knapsack problem **NP-Complete and NP-Hard problems:** P, NP, NP-Cmplete and NP-Hard classes	**08**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. Introduction to the Design and Analysis of Algorithms, Anany Levitin:, 2nd Edition, 2009.Pearson.
2. Computer Algorithms/C++, Ellis Horowitz, Satraj Sahni and Rajasekaran, 2nd Edition, 2014,Universities Press/

Reference Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronal L. Rivest, CliffordStein, 3rd Edition, PHI.
2. Design and Analysis of Algorithms, S. Sridhar, Oxford (Higher Education).

CO#	COURSE OUTCOMES
CO1	Identify various time and space complexities of various algorithms for well known problems like searching, sorting etc.
CO2	Estimate the computational complexity of different algorithms using divide and conquer, decrease and conquer design paradigm.
CO3	Design efficient algorithms for sorting, string matching, and graph problems using dynamic programming.
CO4	Apply greedy and backtracking techniques to solve various problems.
CO5	Implement the programs by using backtracking and branch and bound and analyze the complexities.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO2	3	3	2	-	-	-	-	-	-	-	-	2	2	3	-
CO3	3	3	3	-	-	-	-	-	-	-	-	2	2	3	-
CO4	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO5	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-

Object-Oriented Programming with JAVA
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER – III

Course Code	**21CS35**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**

CREDITS – 02

Course Objectives:

1. To impart the basic concepts of data structures and algorithms.
2. To understand concepts about searching and sorting techniques
3. To understand basic concepts about stacks, queues, lists, trees and graphs.
4. To enable them to write algorithms for solving problems with the help of fundamental data structures

Modules	**Hours**
Module I	
Introduction to Java History of Java, Bytecode, Features of Java, Java Applications, Differences between C , C++ and Java, Principles of Object Oriented Programming, Building and Running Java Program, Java Tokens, Data Types, Variables, Operators, Classes and Objects: Class Fundamentals, Object creation; Type Conversion and Casting, Arrays, Access Specifiers.	**06**
Module II	
Control Statements, Methods, Constructors **Control Statements:** Java Selection Statements, Iterative Statements, Jump Statements; **Methods:** Method Definition, Method accessing, Method That Takes Parameters, Methodoverloading; **Constructors:** Constructor Definition, Parameterized Constructors, Constructor overloading;Using this keyword, Instance variable hiding; Java static keyword.	**06**
Module III	
Inheritance, Polymorphism **Inheritance:** Inheritance basics, Types of Inheritance, Simple and Multi-level Inheritance, Using super keyword, the call of Constructors; **Polymorphism:** Run-time Polymorphism- Method overriding, Dynamic Method Dispatch; Using abstract class and final keyword.	**06**
Module IV	
Packages, Interfaces, String Handling **Packages:** Defining Package, Access Protection, Importing Packages; **Interfaces:** Defining and Implementing Interfaces, Nested Interfaces,	**06**

String Handling: String Constructors, String Operations: Concatenation, Conversion; Character Extraction.	
Module V	
Exception Handling, Multithreading **Exception Handling:** Exception Handling Fundamentals, Handling Exceptions using keywords,Nested try statements, **Multithreading:** Thread Creation, Creating Multiple Threads, Thread Priorities, Synchronization, Interthread Communication: Producer Consumer Problem	**06**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill

Reference Books:

1. Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education, 2008, ISBN: 9788131720806.
2. Rajkumar Buyya,S Thamarasi selvi, xingchen chu, Object oriented Programming with java, Tata McGraw Hill education private limited.
3. E Balagurusamy, Programming with Java a primer, Tata McGraw Hill companies.
4. Anita Seth and B L Juneja, JAVA One step Ahead, Oxford University Press, 2017.

CO#	COURSE OUTCOMES
CO1	Demonstrate the ability to develop Java applications for a variety of use cases.
CO2	Design Java programs using control statements, methods, constructors, and other core Java concepts.
CO3	Apply the concepts of inheritance and Polymorphism for solving in real world problems
CO4	Understanding of how to create modular, flexible, and efficient Java programs using packages, interfaces, and effective string handling techniques.
CO5	Illustrates the concepts of Exception Handling and Multithreaded Programming

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO2	2	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO3	2	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO4	2	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO5	2	2	3	-	-	-	-	-	-	-	-	2	2	3	-

Data Structures and its Applications using C Lab [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2022-2023) SEMESTER – III			
Course Code	**21CSL36**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**
CREDITS – 01			

Course Objectives:

- To design, develop, test and debug in C/C++ language considering appropriate data structure.
- Illustrate and implement data types such as stack, queue and linked list and apply themfor the given problem.
- Illustrate and implement the trees and other data structures.

• Implement all the programs in 'C / C++' Programming Language and Linux / Windows asOS

Sl.No.	Title of the Program
1.	Design, Develop and Implement a menu driven Program in C for the following array operations. a. Creating an array of N Integer Elements b. Display of array Elements with Suitable Headings c. Inserting an Element (ELEM) at a given valid Position (POS) d. Deleting an Element at a given valid Position (POS) e. Exit. Support the program with functions for each of the above operations.
2.	Design, Develop and Implement 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.	Design, Develop and Implement 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.	Design, Develop and Implement 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.	Design, Develop and Implement 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.	Design, Develop and Implement 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.	Design, Develop and Implement 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.	Design, Develop and Implement 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.	Design, Develop and Implement 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) = 6x2 y 2 z-4yz5 +3x3 yz+2xy5 z-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.	Design, Develop and Implement 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.	Design, Develop and Implement 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 In order, Preorder and Post Order c. Search the BST for a given element (KEY) and report the appropriate message d. Exit
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 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. Design and 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.

Conduct of Practical Examination:

• Experiment distribution

a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 50 marks

Course Outcomes:

CO#	COURSE OUTCOMES
CO1	Demonstrate theoretical concepts of Arrays, Queues, stack, Linked list, graphs & trees data structures through series of experiments.
CO2	Implement various data structures using C/C++
CO3	Debug syntactical errors, and troubleshoot the problems issues effectively
CO4	Analyze the data and interpret the results.
CO5	Prepare a well-organized Data Structures laboratory report.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO2	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO3	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO4	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO5	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3

Design and Analysis of Algorithms Lab **[As per Choice Based Credit System (CBCS) scheme]** **(Effective from the academic year 2022-2023)** **SEMESTER – III**			
Course Code	**21CSL37**	**CIE Marks**	**50**
Number Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Hours	**30**	**Exam Hours**	**03**
CREDITS-01			

Course Objectives:

1. Design and implement various algorithms in C
2. Employ various design strategies for problem solving.
3. Measure and compare the performance of different algorithms

Sl.No.	**Title of the experiment**
1.	Design a program to search a key element of n integers using binary search algorithm and compute time complexity
2.	Design a program to Sort a given set of n integer elements using Quick Sort method and compute itstime complexity.
3.	Design a program to sort set of n integer elements using Merge Sort method and compute its timecomplexity.
4.	Implement the 0/1 Knapsack problem using (a) Dynamic Programming method. (b) Greedy method.
5.	Design a program to print all the node reachable from a given starting node in a given digraph usingDFS method.
PART – B	
1.	Write a Program find shortest paths to other vertices using Dijkstra's algorithm.
2.	(a) Write a program to find a Minimum Cost Spanning Tree of a given connected undirected graphusing Kruskal's algorithm. (b) Write a program to find Minimum Cost Spanning Tree of a given connected undirected graph usingPrim's algorithm.
3.	Write a program to (a) Implement All-Pairs Shortest Paths problem using Floyd's algorithm. (b) Implement transitive closure using Warshall Algorithm.
4.	Design and implement to find a subset of a given set.
5.	Implement Travelling Salesman problem using Dynamic program

Conduct of Practical Examination:

• Experiment distribution

a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 50 marks

CO#	COURSE OUTCOMES
CO1	Understanding of algorithmic design paradigms and the techniques used for analyzing their efficiency
CO2	Implement programs using various design strategies
CO3	Debug and troubleshoot software issues effectively
CO4	Analyze the data and interpret the results
CO5	Prepare a well-organized laboratory report

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO2	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO3	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO4	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO5	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3

Object-Oriented Programming with Java Lab
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER – III

Course Code	**21CSL38**	**CIE Marks**	**50**
Number Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**

CREDITS-01

Course Objectives:

1. Demonstrate the use of Eclipse/Netbeans IDE to create Java Applications.
2. Using java programming to develop programs for solving real-world problems.
3. Reinforce the understanding of basic object-oriented programming concepts.

Sl.No.	Title of the Program
1.	(a) Write a Java program to implement class mechanism and create object to access the members of class. (b) Write a Java program to implement shift operators
2.	(a) Write a Java program to illustrate Type Casting of the datatype and type conversion. (b) Write a Java program to iterate over Arrays using for Each loop to compute average of n naturalnumbers using Scanner class.
3.	(a) Write a program in Java to demonstrate method overloading using iterative statements. (b) Write a program in Java to demonstrate constructor overloading using this keyword
4.	(a) Write a program in Java that implements multi-level inheritance. (b) Write a Java program to implement method overriding that shows use of super keyword.
5.	(a) Write a Java program to illustrate Dynamic Method Dispatch using hierarchical inheritance (b) Write a Java program for abstract class to find areas of different shapes**.**
6.	Write a Java program that implements interface using extends keyword
7.	(a) Write a Java program that illustrates Exception handling mechanisms. (b) Write a Java program to illustrates break and continue statements.
8.	Write a Java program to illustrate the working of Strings methods.
9.	Write a Java program that creates threads by extending Thread class. a. First thread display "Good Morning "every 1 sec, b. Second thread displays "Hello "every 2 seconds c. Third display "Welcome" every 3 seconds.
10.	Write a Java program for Producer and Consumer Problem using Threads.

Conduct of Practical Examination:

• Experiment distribution

a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 50 marks

CO#	COURSE OUTCOMES
CO1	Demonstrate theoretical concepts of constructor, inheritance, threads and Exception Handling through series of experiments.
CO2	Develop a program using basic programming constructs and standard libraries.
CO3	Apply advanced debugging techniques and utilize integrated development environment (IDEs) to efficiently identify, diagnose, and resolve software issues in java applications.
CO4	Employ advanced data analysis technique and utilize java libraries to process, analyze and interpret data effectively.
CO5	Develop comprehensive and well-structured laboratory reports.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO2	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO3	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO4	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3
CO5	3	2	3	-	2	-	-	-	-	-	-	-	2	-	3

PROJECT-III
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER – III

Course Code	**21PRJ39**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**
CREDITS – 01			

Course Objectives:

- Identify real-world problems across programming, databases, and networking domains and understand their business and technical implications.
- Apply systematic methodologies to design, implement, and optimize solutions.
- Resolve technical challenges through debugging, research, and collaboration.
- Take responsibility for specific roles in a team and collaborate effectively to achieve project goals.
- Present project progress and findings clearly and confidently to both technical and non-technical audiences.

Document the entire project in a structured, professional laboratory report.

Project Guidelines:

- Project work shall preferably be batch wise.
- Evaluation is based on concept clarity, system design, implementation, testing, presentation, and documentation quality, with a focus on proper coding standards, teamwork, and effective communication.
- Viva-voce examination in project work shall be conducted batch-wise.
- Minimum requirement of CIE marks for Project work shall be 50% of the maximum marks.
- Students failing to secure a minimum of 50% of the CIE marks in Project work shall not be eligible for the SEE Project examination.

  For a pass in a Project/Viva-voce examination, a student shall secure a minimum of 40% of the maximum marks prescribed.

CO#	COURSE OUTCOMES
CO1	Identify the topic from various domains (example programming databases, networking) to real world problems.
CO2	Develop methodology for the problem.
CO3	Resolve issues that arise during the project .
CO4	Learn to assign and accept roles and responsibilities within a team and write a good technical reports.
CO5	Exhibit skills in presenting their project findings & progress orally

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3
CO2	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3
CO3	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3
CO4	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3
CO5	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3

AYDA KATEGALU (DoCÄ a³v*n³æ³Ä) [As per Choice Based Credit System (CBCS) Scheme] (Effective from the academic year 2022-2023) SEMESTER – III			
Course Code	**21KANAK310**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**01**	**SEE Marks**	**50**
Total Number of Lecture Hours	**20**	**Exam Hours**	**02**
CREDITS - 01			

Course Objectives:

1) a£ßq ¨ÁµÁ eÁ£z 9øvÀ vÀsr¸ÄvÀzÀ.
2) a£ßq §gvsn aÄøeÄ wBÄv½a vÀsr¸ÄvÀzÀ.
3) a£ßq £ÁqÄ £Àr, ¸o¸øwoCÄ §nÎ w½¸ÄvÀzÀ.
4) a£ßq ¨ÁµÁ ¥9vÄv£Àß ¨B¸ÄvÀvÀzÀ.

MODULE I

1) vs¸ø£ vÄonvÄ– vCÁ¹ voa¤9± soCÄ¸onÁg (&9evÁ¸)
2) as£oCÄRgÁ& - egoC£

MODULE II

3) zÁø – ge±9&g aob
4) vCÁR – a9±v vÄBR

MODULE III

5) aÁqÄ - ¹zgÁvÄ ºs£eï
6) D¸oCÄo§ e¾Ánee£ vÊø – gzÁ£oz ¸Á°

MODULE IV

7) e§°nBÄ – gÁ5v9oz.6Á¸e9¸
8) evÈegÄ - ¦. ®oa9±

MODULE V

9) 9§Zsø£ ¥s9¸Á¦9¸Ä– a.¦ ¥sstZoz.e9C¹
10) ºoR£gvÄ£oCÄ ºsgn–gÁC±9&g e9gvCÁe

Course Outcomes :

CO#	COURSE OUTCOMES
CO1	a£ßq ¸Áœe §nÎ 9øeÄasBÄeÁg.
CO2	a£ßq ¨ÁµÁeÁ£z vÄºev£Àß w½zÄasBÄeÁg.
CO3	¨ÁµÁ9vCÁ£v£Àß ¨B¹asBÄeÁg.
CO4	a£ßq ¸Áœe aÈwnB §nÎ D¸& vÀsqÄez.

¥gÁªÀ±5£³nbxhæÀ:

1) **DoCÄ a³x*n³æ³Ä:** ¥s. £Á£Á¸Áº9§ c¸ï, ºZqz ¥.¸ÁgÁon ±gs§¸v 9±9zÁ®oCÄ a®§ÄgR.

KANNADA KALI-III
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER – III

Course Code	**21KANKK310**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**01**	**SEE Marks**	**50**
Total Number of Lecture Hours	**20**	**Exam Hours**	**02**

CREDITS - 01

Course Objectives:

1) 9£¸¨Á¶a 9zÁytn½n a£ßq vCÁe£ÁqÄv¸zÄ §goCÄÄv aË±®¸ a°¸Äv¸zÄ.
2) a£ßq ¨ÁµÁ eÁ£z 9øv¸ vÄsr¸Äv¸zÄ.
3) a£ßq §gvsn aÄøeÄ wBÄv½a v¸sr¸Äv¸zÄ.
4) a£ßq £Áqä £Ár, ¸o¸øwoCÄ §nî w½¸Äv¸zÄ.
5) a£ßq ¨ÁµÁ ¥9vÄv£Äß ¨B¸Äv¸v¸zÄ.

MODULE I

Lesson 1. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

Lesson 2. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

MODULE II

Lesson 3. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

Lesson 4. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

MODULE III

Lesson 5. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

Lesson 6. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

MODULE IV

Lesson 7. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

Lesson8. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

MODULE V

Lesson 9. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

Lesson 10. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.

CO#	COURSE OUTCOMES
CO1	To understand the necessity of local language for comfortable life.
CO2	To speak, read write Kannada language as per requirement.
CO3	To communicate [converse] in Kannada language in their daily life with Kannada speakers.
CO4	To listen and understand the Kannada language properly.
CO5	To speak in polite conversation.

Dzsg³nbxhæÄ:

1. a£ßq a° - ¥s.£Á£Á¸Áºg§ ºZqz ¥.¸ÁgÁon ±gs§¸v 9±9zÁ®oCÄ a®§ÄgR.
2. vCÁeÁqÄ a£ßq - a£ßq ¸Áœe ¥øµeÄ - ¨onBsgÄ.

*

Unix and Shell Programming
[As per Choice Based Credit System(CBCS)scheme]
(Effective from the academic year 2022-2023)
SEMESTER – III

Course Code	**21AEC3111**	**CIE Marks**	**50**
Teaching Hours/Week	**02**	**SEE Marks**	**50**
Total No .of Lecture hours	**30**	**Exam Hours**	**03**

CREDITS -01

Course Objectives:

1. To mention the features of Unix
2. To understand the basic concepts of UNIX Architecture and basic Commands
3. To understand different types of Files, File system and basic file system commands.
4. To understand the concepts related to basic scripting language

Explore Unix

Introduction: operating System, objective, History, Features of UNIX, Kernel and Shell.

Unix File System: File and Common Commands, Shell, more about files, Directories, UNIX system, Basics of file directories.

Unix commands with syntax: Syntax and Unix commands ,

Unix shells: History of Unix shells, Shell Command files, Shell programming on files .

Programs List

1	**Basic Commands** a. Use the **ls** command to list files in your home directory. b. Use the **pwd** command to print the current working directory. c. Use the **cd** command to navigate to different directories. d. Use the **mkdir** command to create a new directory. e. Use the **touch** command to create an empty file.
2	**File Manipulation** a. Create a file named "mytext.txt" and add some text to it using a text editor. b. Use the **cp** command to make a copy of "mytext.txt" with a different name. c. Use the **mv** command to rename the copied file. d. Use the **rm** command to delete a file. e. Use the **cat** command to display the contents of a file.
3	**Text Processing** a. Use the **echo** command to print a message to the terminal. b. Use the **grep** command to search for a specific word in a file. c. Use the **wc** command to count the number of lines, words, and characters in a file. d. Use the **sort** command to sort the lines of a file. e. Use the **head and tail** commands to display the first and last few lines of a file.

4	**Permissions** a. Use the **ls -l** command to view the permissions of files in a directory. b. Use the **chmod** command to change the permissions of a file. Use the **chown** command to change the owner of a file.
5	**Variables** a. Assign a value to a variable and use echo to display its content. b. Combine variables and text in a echo statement. c. Experiment with different types of quotes (single, double) and understand their effects on variable interpolation.
6	**Scripting Basics** a. Write a simple script that prints a greeting message. b. Declare variables in a script and display their values. c. Use user input in a script to personalize the output.
7	**Control Structures** a. Write a script that uses an if statement to check if a number is positive or negative. b. Use a for loop to print numbers from 1 to 5. c. Implement a while loop that counts down from 3 to 1
8	**Command Substitution** **a.** Use command substitution to capture the output of a command and assign it to a variable. b. Incorporate command substitution into a script to dynamically obtain information.
9	**Job Control** a. Use the **ps** command to list running processes. b. Use the **kill** command to terminate a specific process.
10	**Environment Variables** a. Display the values of environment variables such as HOME, PATH, and USER. b. Experiment with modifying the value of an environment variable.

Text Books:

1. Sumitabha Das., Unix Concepts and Applications., 4thEdition., Tata McGraw Hill (Chapter 1,2,3,4,5,6,8,13,14)
2. W. Richard Stevens: Advanced Programming in the UNIX Environment, 2nd Edition, PearsonEducation, 2005 (Chapter 3,7,8,10,13,15)
3. Unix System Programming Using C++ - Terrence Chan, PHI, 1999. (Chapter 7,8,9,10)

Conduct of Practical Examination:

• Experiment distribution

a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 50 marks

CO#	COURSE OUTCOMES
CO1	Utilize Basic UNIX commands effectively.
CO2	Write effective scripts using software tools
CO3	Debug and troubleshoot software issues effectively
CO4	Analyze the data and interpret the results
CO5	Prepare a well-organized laboratory report

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3	-	-	-	-	-	-	2	-	-	2	-	3
CO2	2	2	3	-	-	-	-	-	-	2	-	-	2	-	3
CO3	2	2	3	-	-	-	-	-	-	2	-	-	2	-	3
CO4	2	2	3	-	-	-	-	-	-	2	-	-	2	-	3
CO5	2	2	3	-	-	-	-	-	-	2	-	-	2	-	3

Scilab **[As per Choice Based Credit System(CBCS)scheme]** **(Effective from the academic year 2022-2023)** **SEMESTER – III**			
Course Code	**21AEC3112**	**CIE Marks**	**50**
Teaching Hours/Week	**02**	**SEE Marks**	**50**
Total No .of Lecture hours	**30**	**Exam Hours**	**03**
CREDITS -01			
Course Objectives:			
• **Learn core skills of SCILAB** • **Able to understand the basic concepts of programming**			
1. Study of Basic Scilab Commands 2. Matrix Constructors and Operations 3. Matrix Bitwise, Relational & Logical Operations 4. Control Structures (If-Else, If-Elseif –Else, Select) 5. Control Structures (For, While, Break and Continue) 6. Graphics – 2D Plots 7. Scilab – Civil Application Program (1) 8. Scilab – Civil Application Program (2) 9. Scilab – Electronics Application Program (1) 10. Scilab – Electronics Application Program (2)			

Conduct of Practical Examination:

• Experiment distribution

a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 5 0 marks

CO#	COURSE OUTCOMES
CO1	Learn fundamental Scilab programming concepts, including matrix operations, control structures, and data visualization, to solve computational problems.
CO2	Implement and demonstrate SCI lab basic programs
CO3	Debug and troubleshoot software issues effectively
CO4	Analyze the data and interpret the results
CO5	Prepare a well-organized laboratory report

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3
CO2	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3
CO3	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3
CO4	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3
CO5	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3

ENGINEERING MATHEMATICS – IV

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2022-2023)

SEMESTER – IV

Course Code	**21MAT41**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**03**	**SEE Marks**	**50**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS – 03

Course Objectives:

- Learn Fourier series and Fourier transforms.
- Conversant with numerical methods to solve ordinary differential equations.
- Know then complex combers, Analytic function and associated results and problems.
- Understand Joint probability distribution and stochastic processes arising in science andengineering.
- Understand the definition of sequence, series and its importance.
- Discuss the elementary concepts of graph theory.
- Know the finite difference method and use in solving partial differential equation.

Modules	Hours
Module I	
Fourier Series: Periodic functions, Dirichlet's condition, Fourier Series of periodic functions wit period 2π and with arbitrary period 2c. Fourier series of even and odd functionsHalf range Fourier Series, practical harmonic analysis	**08**
Module II	
Joint probability distribution: Joint Probability distribution for two discrete random variables,expectation, covariance, correlation coefficient. **Stochastic process:** Stochastic processes, probability vector, stochastic matrices, fixed points, regular stochastic matrices, Markov chains, higher transition probability-simple problems.	**08**
Module III	
Numerical Methods: Numerical solution of ordinary differential equations of first order and first degree, Taylor's series method, modified Euler's-method Runge - Kutta method of fourth order. Milne's and Adams-Bashforth predictor and corrector methods (No derivations of formulae)	**08**
Module IV	
Numerical Methods: Numerical solution of second order ordinary differential equations, Runge- Kutta Method and Milne's Method, Numerical solution of P.D.E: Numerical solution of Heat equation, Wave equation, problems.	**08**
Module V	
Fourier Transforms: Infinite Fourier transforms, Fourier sine and cosine transforms. Inverse Fourier- transform (5 Assignment Problem). **Complex line Integrals:** Cauchy's Integration theorem, Cauchy integral formula, Laurent's Series, types of singularities. Residue, Poles, Cauchy's Residue theorem (without proof) and Problems. **Transformations:** Bilinear transformations and problems.	**08**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.

The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. B.S. Grewal : Higher Engineering Mathematics, Khanna Publishers, 43rd Ed., 2015.
2. E. Kreyszig : Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed.(Reprint), 2016.

Reference books:

1. C.Ray Wylie, Louis C.Barrett : "Advanced Engineering Mathematics", 6th Edition,
2. McGraw-Hill Book Co., New York, 1995.
2. James Stewart: "Calculus –Early Transcendentals", Cengage Learning India Private Ltd., 2017.
3. B.V.Ramana : "Higher Engineering Mathematics" 11th Edition, Tata McGraw-Hill, 2010.
4. Srimanta Pal & Subobh C Bhunia:"Engineering Mathematics", Oxford University Press,3rd Reprint,2016.
5. Gupta C.B., Singh S.R. and Mukesh Kumar : "Engineering Mathematics for Semester I & II",Mc-Graw Hill Education (India) Pvt.Ltd., 2015.

Web links and Video Lectures:

1. http://nptel.ac.in/courses.php?disciplineID=111
2. http://www.class-central.com/subject/math
3. http://academicearth.org.

CO#	COURSE OUTCOMES
CO1	Understanding the Periodic function and Fourier series expansion of different functions and its application to analyze circuits.
CO2	Learn to solve the problems on Joint probability distribution and to know the concept of stochastic processes and Markov's chains in discrete time.
CO3	Solving the first order first degree ordinary differential equations arising in flow problems by numerical methods.
CO4	Make the use of second order ordinary and partial differential equations arising in heat and wave equations by numerical methods.
CO5	Make the use of the models using advanced concept of graphs in the real - world applications

OPERATING SYSTEM

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2022-2023)

SEMESTER – IV

Course Code	**21CS42**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**03**	**SEE Marks**	**50**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS – 03

Course Objectives:

1. Demonstrate the need for OS and different types of OS.
2. Apply suitable techniques for management of different resources
3. Use processor, memory, storage and file system commands
4. Analyze deadlock identification and prevention mechanisms, segmentation and paging techniques
5. Realize the different concepts of OS in platform of usage through case studies

Modules	**Hours**
Module I	
Introduction to operating systems, System structures: What operating systems do; Computer System organization; Computer System architecture; Operating System structure; Operating System operations; Process management; Memory management; Storage management; Protection and Security; Distributed system; Special-purpose systems; Computing environments. **Operating System Services:** User - Operating System interface; System calls; Types of system calls; System programs; Operating system design and implementation; Operating System structure; Virtual machines; Operating System generation; System boot. **Process Management:** Process concept; Process scheduling; Operations on processes; Inter process communication	**08**
Module II	
Multi-threaded Programming: Overview; Multithreading models; Thread Libraries; Threading issues. Process Scheduling: Basic concepts; Scheduling Criteria; Scheduling Algorithms; Multiple-processor scheduling; Thread scheduling. **Process Synchronization:** Synchronization: The critical section problem; Peterson's solution; Synchronization hardware; Semaphores; Classical problems of synchronization; Monitors	**08**
Module III	
Deadlocks: Deadlocks; System model; Deadlock characterization; Methods for handling deadlocks; Deadlock prevention; Deadlock avoidance; Deadlock detection and recovery from deadlock. **Memory Management:** Memory management strategies: Background; Swapping; Contiguous memory allocation; Paging; Structure of page table; Segmentation.	**08**
Module IV	
Virtual Memory Management: Background; Demand paging; Copy-on-write; Page replacement; Allocation of frames; Thrashing. **File System, Implementation of File System: File system:** File concept; Access methods; Directory structure; File system mounting; File sharing; Protection: Implementing File system: File system structure; File system implementation;	**08**

Directory implementation; Allocation methods; Free space management.	
Module V	
Secondary Storage Structures, Protection: Mass storage structures; Disk structure; Disk attachment; Disk scheduling; Disk management; Swap space management. Protection: Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of access matrix, Access control, Revocation of access rights, Capability- Based systems. **Case Study:** The Linux Operating System: Linux history; Design principles; Kernel modules; Process management; Scheduling; Memory Management; File systems, Input and output; Inter- process communication.	**08**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks:

1. Operating System Concepts, by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, 9th Edition, Wiley India, 2012.
2. Operating systems, by H.M.Deitel, D.J.Deitel, D.R.Choffnes, 3rd edition, Pearson Education.

Reference Books:

1. Operating Systems, A Concept-Based Approach, by DM Dhamdhere, 3rd Edition, Tata McGraw-Hill, 2012.
2. Modern Operating Systems, by Andrew S. Tanenbaum and Herbert Bos, 4th Edition, Pearson, 2014.
3. Ann McHoes Ida M Fylnn, Understanding Operating System, Cengage Learning, 6th Edition
4. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw-Hill, 2013.
5. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI(EEE), 2014.
6. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

CO#	COURSE OUTCOMES
CO1	Identify the structure of an operating system and its scheduling mechanism.
CO2	Interpret the allocation of resources for a process using scheduling algorithm.
CO3	Identify root causes of deadlock and provide the solution for deadlock elimination.
CO4	Explain about the storage structures and learn about the Linux Operating system.
CO5	Analyze Storage Structures and Implement Customized Case study.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	-	-	-	-	-	-	2	2	3	-
CO2	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO3	3	3	3	-	-	-	-	-	-	-	-	2	2	3	-
CO4	3	2	2	-	-	-	-	-	-	-	-	2	2	3	-
CO5	3	3	2	-	-	-	-	-	-	-	-	2	2	3	-

PYTHON PROGRAMMING

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2022-2023)

SEMESTER – IV

Course Code	**21CS43**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**03**	**SEE Marks**	**50**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS – 03

Course Objectives:

- Understand the syntax and semantics of the Python programming language.
- To illustrate the process of structuring the data using lists, and tuples.
- To inculcate knowledge of parsing of regular expressions and their usage in various application domains.
- Demonstrate the use of built-in functions to navigate the file system.
- Implement Object-Oriented Programming concepts in Python and discuss the files, Pandas, and Data Virtualization concepts.

Modules	**Hours**
Module I	
Python Basics: Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program, Flow control: Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with sys exit(), Functions: def Statements with Parameters, Return Values and return Statements, The None Value, Keyword Arguments and print(), Local and Global Scope, The global Statement, Exception Handling, A Short Program: Guess the Number	**08**
Module II	
Lists: The List Data Type, Working with Lists, Augmented Assignment Operators, Methods, Example Program: Magic 8 Ball with a List, List-like Types: Strings and Tuples, References, Dictionaries, and Structuring Data: The Dictionary Data Type, Pretty Printing, Using Data Structures to Model Real-World Things.	**08**
Module III	
Manipulating Strings: Working with Strings, Useful String Methods, Project: Password Locker, Project: Adding Bullets to Wiki Markup Reading and Writing Files: Files and File Paths, The OSpath Module, The File Reading/Writing Process, Saving Variables with the shelve Module, Saving Variables with the print. Format () Function, Project: Generating Random Quiz Files, Project: Mult clipboard.	**08**
Module IV	

Organizing Files: The shutil Module, Walking a Directory Tree, Compressing Files with the zip file Module, Project: Renaming Files with American-Style Dates to European-Style Dates, Project: Backing Up a Folder into a ZIP File, Debugging: Raising Exceptions, Getting the Traceback as a String, Assertions, Logging, IDLE‟s Debugger. **Classes and objects:** Programmer-defined types, Attributes, Rectangles, Instances as return values, Objects are mutable, Copying, Classes, and functions: Time, Pure functions, Modifiers, prototyping versus planning, Classes and methods: Object-oriented features, Printing objects, Another example, A more complicated example, The init method, The str method, Operator overloading, Type-based dispatch, Polymorphism, Interface, and implementation. Inheritance, Card objects, Class attributes, comparing cards, Decks, Printing the deck, Add, remove, shuffle and sort, Inheritance, Class diagrams, Data encapsulation.	**08**
Module V	
NumPy: Introduction to NumPy, creating arrays, Indexing Arrays, Array Transposition, Universal Array Function, Array Processing, Array Input, and Output. **Pandas and Data Visualization:** Introduction, Series and Data Frames in pandas and Data Visualization.	**08**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks/Reference Books: -

1. Sweigart,“Automate the Boring Stuff with Python”,1 stEdition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at https://automatetheboringstuff.com/)
2. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist”, 2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at http://greenteapress.com/thinkpython2/thinkpython2.pdf)
3. Learning to Program using Python by Cody Jackson, Second Edition, 2014.
4. Pandas for Everyone: Python Data Analysis by Daniel Y. Chen, First Edition, Pearson, 2018.
5. Python Data Science Handbook by Jake VanderPlas, O'Reilly, 2017.

CO#	COURSE OUTCOMES
CO1	Create basic programs using variables, conditionals, loops, and functions.
CO2	Use lists, tuples, and dictionaries in Python programs.
CO3	Use Python for regex pattern matching, file manipulation, efficient organization, and debugging to solve computational tasks.
CO4	Utilize the concepts of Object-Oriented Programming in Python.
CO5	Develop application python programs using Numpy and Pandas.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO2	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO3	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO4	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-
CO5	3	2	3	-	-	-	-	-	-	-	-	2	2	3	-

Database Management Systems

[As per Choice Based Credit System (CBCS) scheme]

(Effective from the academic year 2022-2023)

SEMESTER – IV

Course Code	**21CS44**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**03**	**SEE Marks**	**50**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS – 03

Course Objectives:

1. Provide a strong foundation in database concepts, technology, and practice.
2. Practice SQL programming through a variety of database problems.
3. Demonstrate the use of concurrency and transactions in database
4. Design and build database applications for real world problems.

Modules	**Hours**
Module I	
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, roles, and structural constraints, Weak entity types, ER diagrams, Examples.	**08**
Module II	
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, etc.) Examples of Queries in relational algebra. **Mapping Conceptual Design into a Logical Design:** Relational Database Design using ER-to- Relational mapping.	**08**
Module III	
SQL: SQL data definition and data types, specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, Additional features of SQL. **Advances Queries**: More complex SQL retrieval queries, Specifying constraints as assertions and action triggers, Views in SQL, Schema change statements in SQL. **Database Application Development:** Accessing databases from applications, An introduction to JDBC, JDBC classes and interfaces, SQLJ, Stored procedures, Case study: The internet Bookshop.	**08**
Module IV	
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, Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and	**08**

Fourth Normal Form, Join Dependencies and Fifth Normal Form. Examples on normal forms. **Normalization Algorithms:** Inference Rules, Equivalence, and Minimal Cover, Properties of Relational Decompositions, Algorithms for Relational Database Schema Design, Nulls, Dangling tuples, and alternate Relational Designs, Further discussion of Multivalued dependencies and 4NF, Other dependencies and Normal Forms	
Module V	
Transaction Processing: Introduction to Transaction Processing, Transaction and System concepts,Desirable properties of Transactions,Characterizing schedules based o 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, Multisession Concurrency control techniques, Validation Concurrency control techniques, Granularity of Data items and Multiple Granularity Locking.	**08**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Textbooks/Reference Books: -

1. Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
2. Database management systems, Ramakrishnan, and Gehrke, 3rd Edition, 2014, McGraw Hill

Reference Books: 1. Abraham Silberschatz, Henry F. Korth and S. Sudarshan's Database System Concepts 6th EditionTata Mcgraw Hill Education Private Limited

CO#	COURSE OUTCOMES
CO1	Design conceptual entity relationship diagrams for the real-world applications.
CO2	Apply knowledge of relational databases to solve practical problems,
CO3	Use Structured Query Language (SQL) for database manipulation and also demonstrate the basic of query evaluation.
CO4	Implement normalization algorithms using database design theory for different applications
CO5	Analyze and implement transaction processing, concurrency control and database recovery protocols in databases.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	2	-	-	-	-	2	2	3	-
CO2	3	2	3	-	-	-	2	-	-	-	-	2	2	3	-
CO3	3	2	3	-	-	-	2	-	-	-	-	2	2	3	-
CO4	3	2	3	-	-	-	2	-	-	-	-	2	2	3	-
CO5	3	2	3	-	-	-	2	-	-	-	-	2	2	3	-

AUTOMATA THEORY AND COMPUTABILITY
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER –IV

Course Code	**21CS45**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**03**	**SEE Marks**	**50**
Total Number of Lecture Hours	**40**	**Exam Hours**	**03**

CREDITS – 03

Course Objectives:

1. Introduce the fundamental concepts of Automata Theory, Formal Languages and compiler design
2. Principles Demonstrate Application of Automata Theory and Formal Languages in the field of compiler design
3. Develop understanding of computation through Push Down Automata and Turing Machines
4. Introduce activities carried out in different phases of Phases compiler
5. Identify the undecidability problems.

Modules	**Teaching Hours**
Module I	
Why study the Theory of Computation, Languages and Strings: Strings, Languages. A Language Hierarchy, Computation, **Finite State Machines (FSM)**: Deterministic FSM, Regular languages, Designing FSM, Nondeterministic FSMs, From FSMs to Operational Systems, Simulators for FSMs, Minimizing FSMs.	**08**
Module II	
Regular Expressions (RE): what is a RE?, Kleene's theorem, Applications of REs, Manipulating and Simplifying REs. Regular Grammars: Definition, Regular Grammars and Regular languages. Regular Languages (RL) and Non-regular Languages: How many RLs, To show that a language is regular, Closure properties of RLs, to show some languages are not RLs.	**08**
Module III	
Context-Free Grammars (CFG): Introduction to Rewrite Systems and Grammars, CFGs and languages, designing CFGs, simplifying CFGs, proving that a Grammar is correct, Derivation and Parse trees, Ambiguity, Normal Forms. Pushdown Automata (PDA): Definition of non-deterministic PDA, Deterministic and Non-deterministic PDAs, No determinism and Halting, alternative equivalent definitions of a PDA, alternatives that are not equivalent to PDA	**08**
Module IV	

Algorithms and Decision Procedures for CFLs: Simplification of CFG, Elimination of ε- production and Unit Symbol, CFLs are closed under Union, Concatenation and Star-closure. CFLs are not closed under Intersection and complementation. **Turing Machine**: Turing machine model, Representation, Language acceptability by TM, design of TM, Techniques for TM construction. Extension to the basic Turing Machine	**08**
Module V	
Program techniques for Turing machine, The model of Linear Bounded automata, Multi-stack Machines, TM with semi-infinite tape. **Decidability:** Definition of an algorithm, decidability, decidable languages, Undecidable languages, halting problem of TM, Post correspondence problem. Complexity: Growth rate of functions, the classes of P and NP, Quantum Computation: quantum computers, Church-Turing thesis. **Applications:** G.1 Defining syntax of programming language, Appendix J: Security	**08**

Question paper pattern:

- The question paper will have ten questions.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Carl Hamacher, Z. Vranesic & S.Zaky, "Computer Organization", 5th Edition, Tata McGrawHill Publishing Company Ltd. New Delhi, 2002.
2. John L. Hennessy and David A. Patterson, Computer Architecture: A quantitative approach, 5th edition, Morgan Kaufmann Elseveir, 2013.
3. Donald P Leach, Albert Paul Malvino & Goutham Saha: Digital Principles and Applications, 8th Edition, Tata Mcgraw Hill, 2015.

Reference Books:

1. Morris Mano, "Computer System Architecture", PHI, 19862.William Stallings Computer Organization & Architecture, 7th Edition, PHI 2006.
2. Kai Hwang and Naresh Jotwani, Advanced Computer Architecture (SIE): Parallelism, Scalability, Programmability, McGraw Hill Education 3/e. 2015.
3. R D Sudhakar Samuel: IIustrative approach to Logic Design, Sanguine-Pearson, 2010.

CO#	COURSE OUTCOMES
CO1	Design a computational model Finite state machine with conversion between different types of FA and minimize the given FA for any regular language
CO2	Develop regular expressions, languages and apply it for designing compilers.
CO3	Develop context free grammar, push down automata for the given language and conversion between PDA & CFG.
CO4	Simplify CFG & apply the concept of Turing machine for a given Language.
CO5	Analyze and understand decidability and undesirability of various problems with their complexity analysis.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	3	3	-	-	-	-	-	-	-	-	-	2	3	-
CO2	2	3	3	-	-	-	-	-	-	-	-	-	2	3	-
CO3	2	3	3	-	-	-	-	-	-	-	-	-	2	3	-
CO4	2	3	3	-	-	-	-	-	-	-	-	-	2	3	-
CO5	2	3	3	-	-	-	-	-	-	-	-	-	2	3	-

OPERATING SYSTEM LAB
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER – IV

Course Code	**21CSL46**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**

CREDITS – 01

Course Objectives:

1. To learn the fundamentals of Operating Systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication
3. To learn the mechanisms involved in memory management in contemporary OS
4. To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols
5. To know the components and management aspects of concurrency management
6. To learn programmatically to implement simple OS mechanisms

PART-A

No.	Title of the experiment
1.	Write a C program to Simulate the following CPU scheduling algorithms a) FCFS b) SJF c) Round Robin d) Priority
2.	Write a C program to Simulate IPC techniques b) Pipes b) Message Queues c) Shared Memory
3.	Write a C Program to Simulate Classical Problems of Synchronization c) Readers-Writers b) Producers-Consumers C) Dining Philosophers
4.	Write a C Program to simulate Bankers Algorithm for Dead Lock Avoidance
5.	Write a C Programs to Simulate all page replacement algorithms d) FIFO b) LRU c) Optimal Etc.
6.	Write a C program to Simulate Disk Scheduling Algorithms FCFS b) SSTF etc.

PART – B

1.	Build a simple client-server program to transfer files from client to server.
2.	Build a simple shell to transfer a file from the server and pipe it through a word count program.
3.	Build a multithreaded server to transfer files from server to client. Each thread will read one file and send it back

Conduct of Practical Examination:

• Experiment distribution
   a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.
   b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 5 0 marks

CO#	COURSE OUTCOMES
CO1	Demonstrate theoretical concepts of design and implementation of operating system through a series of experiments
CO2	Develop a program for Scheduling, Page replacement & system calls
CO3	Debug and troubleshoot software issues effectively
CO4	Analyze the data and interpret the results
CO5	Prepare a well-organized laboratory report

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO2	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO3	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO4	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO5	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3

PYTHON PROGRAMMING LAB
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER – IV

Course Code	**21CSL47**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**

Course Objectives:

1. Develop program to solve real world problems using python programming.
2. Develop the programs using the concepts of control statements, data structures & files.
3. Apply features of object-oriented and Numpy, pandas package to develop computationally intensive programming & interpret the data.

No.	Title of the experiment
1.	a. Develop a program to read the student's details like Name, USN, and Marks in three subjects. Display the student's details, total marks, and percentage with suitable messages. b. Develop a program to read the name and year of birth of a person. Display whether the person is a senior citizen or not. c. Read N numbers from the console and create a list. Develop a program to print mean, variance, and standard deviation with suitable messages.
2.	a. Write a program to demonstrate different numbered data types in Python and perform a different arithmetic operation on numbers in Python. b. Write a program to create, concatenate and print a string and access a sub-string from a given string. c. Write a Python script to print the current date in the following format "Sun May 29 02:26:23 IST 2017". d. Read a multi-digit number (as char) from the console. Develop a program to print the frequency of each digit with suitable messages.
3.	a. Develop a program to find the largest of three numbers b. Develop a program to generate a Fibonacci sequence of length (N). Read N from the console. c. Write a program to calculate the factorial of a number. Develop a program to compute the binomial coefficient (Given N and R).
4.	a. Implementing programs using Functions (Largest number in a list, area of shape) b. implementing a real-time/technical application using Exception handling (Divide by Zero error, Voter's age validity, student mark range validation)
5.	a. "LIST1" is a list that contain "N" different SRN of students read using a user-defined function with the help of input () function.it is required to add the SRN of "M" for more students that are to be appended or inserted into "LIST1" at the appropriate place. The program must return the index of the SRN entered by the user. b. TUPLE1" and "TUPLE2" are two tuples that contain "N" values of Different data types read using the user -defined function "READ" with the help of the input () function. Elements of "TUPLE1" and "TUPLE2" are to be read one at a time and the "larger" value among them should be placed into "TUPLE3". Display all tuples.

6.	**a.** SET1 and SET2 are two sets that contain unique integers. SET3 is to be created by taking the union or intersection of SET1 and SET2 using the user defined function Operation (). Perform either union or intersection by reading the choice from the user. Do not use built-in function union () and intersection () and also the operators "|" and "&". **b.** The Dictionary "DICT1" contains N Elements and each element in the dictionary has the operator as the KEY and operands as VALUES. Perform the operations on operands using operators stored as keys. Display the results of all operations
7.	Implementing programs using Strings. (Reverse, palindrome, character count, replacing characters)
8.	**a.** Develop a program to print the 10 most frequently appearing words in a text file. [Hint: Use dictionary with distinct words and their frequency of occurrences. Sort the dictionary in the reverse order of frequency and display dictionary slice of first 10 items] **b.** Develop a program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: String method strip(), list method sort(), append(), and file method open(), readlines(), and write()]. **c.** Develop a program to backing up a given Folder (Folder in a current working directory) into a ZIP File by using relevant modules and suitable methods.
9.	Develop a program that uses class Students which prompts the User to enter marks in three subjects and calculate total marks, Percentage and display the scorecard details. [Hint: Use a list to store the marks in three subjects and total marks. Use_init( ) method to initialize name, USN and the lists to store marks and total, Use get Marks() method to read marks into the list, and display() method to display the scorecard details.]
10.	Implementing program using modules and python Standard Libraries (pandas, NumPy, Matplotlib, SciPy)

Conduct of Practical Examination:

• Experiment distribution

a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 5 0 marks

CO#	COURSE OUTCOMES
CO1	Demonstrate theoretical concepts of Python strings, lists, tuples, functions and file manipulation through series of programs.
CO2	Design and develop solutions to given problems using Python.
CO3	Debug syntactical errors, and troubleshoot programming issues effectively.
CO4	Analyze the programs and interpret the results
CO5	Prepare a well-organized Python programming laboratory report

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO2	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO3	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO4	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO5	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3

DATABASE MANAGEMENT SYSTEMS LAB [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2022-2023) SEMESTER – IV			
Course Code	**21CSL48**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**

Course Objectives:

1. Provide a strong foundation in database concepts, technology, and practice
2. Practice SQL programming through a variety of database problems.
3. Provide skills in retrieving and manipulating data stored in databases using SQL queries.
4. Design and build database applications for real world problems
5. Learn techniques for generating reports from databases, including the use of SQL queries and reporting tools.

Sl.No.	Title of the experiment
1.	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 based on 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.

2.	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.
3.	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 by 'Steven Spielberg' to 5.
4.	Consider the schema for College Database: STUDENT(USN, SName, Address, Phone, Gender) SEMSEC(SSID, Sem, Sec) CLASS(USN, SSID) COURSE(Subcode, Title, Sem, Credits) IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA) Write SQL queries to 1. List all the student details studying in fourth semester 'C' section. 2. Compute the total number of male and female students in each semester and in each section. 3. Create a view of Test1 marks of student USN '1BI15CS101' in all Courses. 4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students. 5. Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = 'Outstanding' If FinalIA = 12 to 16 then CAT = 'Average' If FinalIA< 12 then CAT = 'Weak' Give these details only for 8th semester A, B, and C section students
5.	Consider the schema for Company Database: EMPLOYEE(SSN, Name, Address, Sex, Salary, SuperSSN, DNo) DEPARTMENT(DNo, DName, MgrSSN, MgrStartDate) DLOCATION(DNo,DLoc) PROJECT(PNo, PName, PLocation, DNo) WORKS_ON(SSN, PNo, Hours) Write SQL queries to 1. Make a list of all project numbers for projects that involve an employee whose last name is 'Scott', either as a worker or as a manager of the department

	that controls the project. 2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise. 3. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department. 4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator). 5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs.6,00,000.
6.	Develop PL/SQL program using PROCEDURE.
7.	Develop PL/SQL program using FUNCTIONS.
8.	Develop PL/SQL program using CURSOR.
9.	Develop PL/SQL Programs using TRIGGERS.
10.	Develop PL/SQL programs using PACKAGES.

Conduct of Practical Examination:

• Experiment distribution

a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 50 marks

CO#	COURSE OUTCOMES
CO1	Demonstrate database concepts through series of queries.
CO2	Develop a program using MySQL.
CO3	Effectively debug and troubleshoot issues in DBMS programs, ensuring stable and performant database operations.
CO4	Examine data and query outputs.
CO5	Prepare a well-organized laboratory report.

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO2	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO3	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO4	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3
CO5	3	2	3	-	-	-	-	-	-	-	-	-	2	-	3

PROJECT-IV
[As per Choice Based Credit System (CBCS) scheme]
(Effective from the academic year 2022-2023)
SEMESTER – IV

Subject Code	**21PRJ49**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**

CREDITS – 01

Course Objectives:

- To develop interactive, communication, organisation, time management, and presentation skills.
- To impart flexibility and adaptability.
- To inspire independent and team working.
- To expand intellectual capacity, credibility, judgement, intuition.
- To adhere to punctuality, setting and meeting deadlines.
- To instil responsibilities to oneself and others.
- To train students to present the topic of project work in a seminar without any fear, face audience confidently, enhance communication skill, involve in group discussion to present and exchange ideas.

Project Guidelines:

The project should emphasize advanced topics and requires the students to implement more detailed system analysis, design, and coding practices.

Groups of 3 to 5 students are formed to tackle a moderately challenging project that aligns with the current curriculum.

The project journey begins with a detailed proposal, outlining the project's scope, objectives, technology stack, methodology, and expected timeline, which needs approval from a faculty mentor.

The development process is more structured, covering stages like detailed requirement analysis, advanced system design with UML diagrams, development using industry-standard coding practices, comprehensive testing (including unit, integration, and system testing), and deployment.

Students are encouraged to use version control systems and project management tools for efficient collaboration.

A well-documented report must be prepared, highlighting the system design, implementation details, testing processes, and conclusions.

The final evaluation considers the project's technical complexity, design, testing rigor, innovation, teamwork, presentation quality, and the comprehensiveness of documentation.

Proper implementation of software engineering practices, including clean code, modular design, and effective use of databases or other relevant technologies, is crucial for successful project completion.

CO#	COURSE OUTCOMES
CO1	Identify the topic from various domains (example programming databases, networking) to real world problems.
CO2	Develop methodology for the problem.
CO3	Resolve issues that arise during the project.
CO4	Learn to assign and accept roles and responsibilities within a team and write a good technical reports.
CO5	Exhibit skills in presenting their project findings & progress orally

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3
CO2	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3
CO3	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3
CO4	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3
CO5	3	3	3	-	2	1	-	-	2	2	2	-	-	-	3

MAHADASOHIGALU (ªÀÄºÁzÁ¸ÀŠÃœnæÃ)			
[As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2022-2023) SEMESTER – IV			
Course Code	**21KANMD410**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**01**	**SEE Marks**	**50**
Total Number of Lecture Hours	**20**	**Exam Hours**	**02**
CREDITS - 01			
Course Objectives:			
1) a£ßq ¨ÁµÁeÁ£z9øv¸ vÀsr¸Ävz.			
2) a£ßq §gvsnaøeÄ wBÄv½a vsr¸vz.			
3) a£ßq £Áq£Ár, ¸o¸øwoCÄ §nî w½¸ÄvzÄ.			
4) a£ßq ¨ÁµÁ ¥9vÄv£Äß ¨B¸ÄvzÄ.			
MODULE I			
1) ±gs§¸v9±vgg §zÄaÄ (9gBnÄornoCÄº)			
2) ¸Áz£Áa9e. (=gÁz,¥eÁt¨Ázï vÄeÄÌ a®§ÄgRoCÄº)			
MODULE II			
3) zsq¥9¥ ºÁns ±gs§¸v¥9vg ¸o§oznBÄ (1 øoz 6£9 ¦9oÁø¥wnBÄ)			
4) vÄgÄB ±gs§¸¥ (z9vÁ®oCÄ evCÁts, zÁ¸sº vÄºÁvÄ£oCÄ ¨Bvsn)			
MODULE III			
5) ¥sC¸zsq¥9¥ (zÁ9Ätª ¸Áz£)			
6) ±ÊasªÁz£nBÄ			
MODULE IV			
7) ¥sCqÁ. ±gs§¸v¥9¥ (¸ÁvCÁ9a asqÄnnBÄ)			
8) ±ÊasªÁ asqÄnnBÄ			
MODULE V			
9) vÄºÁvÄ£oCÄ vÀºÁ vCÁeoCÄgÄ vsz® £Á®Ä ¥Äs¸¹9oCÄgÄ			
10) sz£oCÄ ¦9oÁø¥wn½oz 8£9 ¦9oÁø¥wnB ¥Äs¸¹9oCÄgÄ			

CO#	COURSE OUTCOMES
CO1	a£ßq ¸Áœe¸ §nî 9øeÄasBÄeÁg.
CO2	a£ßq ¨ÁµÁeÁ£z vÄºev£Äß w½zÄasBÄeÁg
CO3	¨ÁµÁ9vCÁ£v£Äß ¨B¹asBÄeÁg.
C04	a£ßq ¸Áœe¸ aÈwnB §nî D¸& vÄsqÄez

DzsÁg³n³.ox³:
ªÄºÁzÁ¸ŠÃœnæÄ:

1. ¥.zÁ£ ¸o¥ÁzagÄ: vCÁeš9&9 qÁ. zÁaÁoCÄš c¸ï. 9¥ ¸o¥ÁzagÄ. qÁ. co. c¸ï. ¥Án9®
2. ¥.¸ÁgÁon ±gš§¸v 9±9zÁ®oCÄ, a®§ÄgR

KANNADA KALI-IV [As per Choice Based Credit System (CBCS) scheme] (Effective from the academic year 2022-2023) SEMESTER – III			
Course Code	**21KANKK410**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**01**	**SEE Marks**	**50**
Total Number of Lecture Hours	**20**	**Exam Hours**	**02**
CREDITS - 01			
Course Objectives:			
1) 9£¸¨Á¶a 9zÁytn½n a£ßq vCÁe£ÁqÄvçzÄ §goCÄÄvaË±®¸ a°¸Ävçz Ä. 2) a£ßq ¨ÁµÁeÁ£z9øvç vÄsr¸Ävçz Ä. 3) a£ßq §gvsnaÄøeÄ 4) wвÄv½a vÄsr¸Ävçz Ä. 5) a£ßq £Áqä £År, ¸o¸øwoCÄ §nî w½¸Ävçz Ä. 6) a£ßq ¨ÁµÁ ¥9vÄv£Äß ¨в¸Ävçvçz Ä.			
MODULE I			
Lesson 1. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises. **Lesson 2.**Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.			
MODULE II			
Lesson 3. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises. **Lesson 4.**Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.			
MODULE III			
Lesson 5. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises. **Lesson 6.**Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.			
MODULE IV			
Lesson 7. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises. **Lesson8.**Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.			
MODULE V			
Lesson 9. Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises. **Lesson 10.**Conversation 1, Conversation 2, Conversation 3, Vocabulary, Exercises.			

CO#	COURSE OUTCOMES
CO1	To understand the necessity of local language for comfortable life.
CO2	To speak, read write Kannada language as per requirement.
CO3	To communicate [converse] in Kannada language in their daily life with Kannada speakers.
CO4	To listen and understand the Kannada language properly.

DzsÁg³nÖbxÀhÆÄ:

1. vÇÁeÁqÄ a¦£ßq¦ - a¦£ßq¦ ¸Áœe¸ ¥ømeÄ¦ - ¨onBSgÄ

SCRIPTING LANGUAGES
[As per Choice Based Credit System(CBCS)scheme]
(Effective from the academic year 2022-2023)
SEMESTER – IV

Course Code	**21AEC4111**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**

Course Objectives: This course will enable students to:

- Use JavaScript for dynamic effects
- To prepare PHP scripts
- Use JavaScript & PHP to validate form input entry

Sl.No.	Title of the experiment
1.	Write a JavaScript to design a simple calculator to perform the following operations: sum, product, difference and quotient
2.	Write a JavaScript that calculates the squares and cubes of the numbers from 0 to 10and outputs HTML text that displays the resulting values in an HTML table format.
3.	Write JavaScript to validate the following fields of the Registration page. a. First Name (Name should contains alphabets and the length should not be less than 6 characters). b. Password (Password should not be less than 6 characters length). c. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com) d. Mobile Number (Phone number should contain 10 digits only).e.Last Name and Address (should not be Empty).
4.	Write an HTML page including any required JavaScript that takes a number from text field in the range of 0 to 999 and shows it in words. It should not accept four and above digits, alphabets and special characters
5.	Write a JavaScript code that displays text “TEXT-GROWING” with increasing fontsize in the interval of 100ms in RED COLOR, when the font size reaches 50pt it displays “TEXT-SHRINKING” in BLUE color. Then the font size decreases to 5pt.
6.	a. Install and configure PHP, web server, MYSQL b. Write a program to print “Welcome to PHP” c. Write a simple PHP program using expressions and operators.
7.	Develop and demonstrate PHP Script for the following problems: a) Write a PHP Script to find out the Sum of the Individual Digits. b) Write a PHP Script to check whether the given number is Palindrome or not
8.	Write a PHP program to keep track of the number of visitors visiting the web page andto display this count of visitors, with proper headings.
9.	Write a PHP Program to display current Date, Time and Day.
10.	Write the PHP programs to do the following: a. Implement simple calculator operations. b. Find the transpose of a matrix. c. Multiplication of two matrices. d. Addition of two matrices

11.	Write a PHP program named states.py that declares a variable states with value"Mississippi Alabama Texas Massachusetts Kansas". write a PHP program that does the following: a. Search for a word in variable states that ends in xas. Store this word in element 0of a list named statesList. b. Search for a word in states that begins with k and ends in s. Perform a case insensitive comparison. Store this word in element1 of statesList. c. Search for a word in states that begins with M and ends in s. Store this word inelement 2 of the list. d. Search for a word in states that ends in a. Store this word in element 3 of the list.
12.	Write a PHP program to sort the student records which are stored in the databaseusing selection sort.
13.	Write a PHP program for sending and receiving plain text message (email).

Conduct of Practical Examination:

• Experiment distribution

a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 5 0 marks

CO#	COURSE OUTCOMES (COs)
CO1	Design and develop dynamic, interactive web applications using HTML, CSS, JavaScript, and form validation techniques.
CO2	Implement full-stack web applications by integrating client-side technologies with server-side logic using PHP.
CO3	Debug and troubleshoot software issues effectively
CO4	Analyze the data and interpret the results
CO5	Prepare a well-organized laboratory report

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3
CO2	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3
CO3	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3
CO4	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3
CO5	2	2	3	-	-	-	-	-	-	2	-	-	-	-	3

XML Programming
[As per Choice Based Credit System(CBCS)scheme]
(Effective from the academic year 2022-2023)
SEMESTER – IV

Course Code	**21AEC4112**	**CIE Marks**	**50**
Number of Lecture Hours/Week	**02**	**SEE Marks**	**50**
Total Number of Lecture Hours	**30**	**Exam Hours**	**03**

Course O bjectives: This course will enable students to:

- To learn the purpose and advantages of using XML
- To learn XML syntax
- To apply XML syntax to data and interactive web development
- To learn to work with DTD, CSS, XSLT and Schemas

List of Experiments

1. Create a simple XML document with attributes
2. Develop an XML file using the Internal DTD
3. Create an external DTD and implement it in XML file
4. Write an XML file which will display the Book information which includes the following: 1) Title of the book 2) Author Name 3) ISBN number 4) Publisher name 5) Edition 6) Price.
5. Write a Document Type Definition (DTD) to validate the above XML file in program 5
6. Design an XML document to store information about a student in an engineering college affiliated to SUK. The information must include USN, Name, Name of the College, Brach, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
7. Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.
8. Design an XML document to store information about patients in a hospital. Information about patients must include name (in 3 parts, first name, middle name, last name), social security number (SSN), age, room number, primary insurance company – including member identification number, group number and address – secondary insurance company (in the same sub parts as for the primary insurance company), known medical problems, and known drug allergies. Both attributes and nested tags must be included. Make up sample data of atleast 4 patients. Create a CSS style sheet for the above XML document and use it to create a display of that document.
9. Create the XSLT style sheet to format all the patient elements of the XML, document of program 8 and use it to create a display of whole element
10. Create an XML document that contains 10 users information. Write a Java Program, which takes User Id as input and returns the user details by taking the user information from XML document using DOM parser or SAX parser.

Conduct of Practical Examination:

• Experiment distribution

a) For laboratories having only one part: Students are allowed to pick one experiment from the lot with equal opportunity.

b) For laboratories having PART A and PART B: Students are allowed to pick one experiment from PART A and one experiment from PART B, with equal opportunity.

• Change of experiment is allowed only once and marks allotted for procedure to be made zero of the changed part only.

• Marks Distribution

SEE are mentioned here, writeup-15%, Conduction procedure and result in -70%, Viva-voce 15% of maximum marks. SEE for practical shall be evaluated for 50 marks

CO#	COURSE OUTCOMES
CO1	Illustrate fundamental concepts of XML using DTDs, CSS, and XSLT to model and display structured data for various scenarios.
CO2	Design and implement XML documents
CO3	Debug and troubleshoot software issues effectively
CO4	Analyze the data and interpret the results
CO5	Prepare a well-organized laboratory report

CO-PO-PSO mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3	-	2	-	-	-	-	2	-	2	2	-	3
CO2	2	2	3	-	2	-	-	-	-	2	-	2	2	-	3
CO3	2	2	3	-	2	-	-	-	-	2	-	2	2	-	3
CO4	2	2	3	-	2	-	-	-	-	2	-	2	2	-	3
CO5	2	2	3	-	2	-	-	-	-	2	-	2	2	-	3