Scheme of Teaching, Examination and Syllabus

B.E. COMPUTER SCIENCE AND ENGINEERING

Batch: 2023-24

Second Year
(III and IV SEMESTER)

(Effective from the academic year 2024-2025)

Sree Siddaganga Education Society®

Siddaganga Institute of Technology

(An Autonomous institute affiliated to Visvesvaraya Technological University, Belagavi)

(Approved by AICTE, New Delhi, Accredited by NAAC with 'A++' and ISO 9001-2015 certified)

B.H. Road, Tumakuru-572 103, Karnataka, India

Phone: Direct +91-816-2282696,Fax:+91-816-2282994

E-mail: principal@sit.ac.in web: www.sit.ac.in

SCHEME OF TEACHING AND EXAMINATION (160 Credits Scheme)

(Effective from the academic year 2024-2025)

III Semester

B.E. in Computer Science & Engineering — Batch:2023-2024

Sl. No.	Course and Course Code		Course Title	Teaching / Paper setting Dept.	Teaching hrs./week				Examination				Credits
					Lecture	Tutorial	Practical/ Drawing	Self-Study Component	Duration in hrs.	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1.	PCC / BSC	S3MATC	Statistics and Probability	Dept. / Maths	3	0	0	3.5(48 hrs)	3	50	50	100	3
2.	IPCC	S3CCSI01	Operating Systems(I)	Dept.	3	0	2	3.5(50 hrs)	3	50	50	100	4
3.	IPCC	S3CCSI02	Digital Circuits and Computer Organizations (I)	Dept.	3	0	2	3.5(50 hrs)	3	50	50	100	4
4.	PCC	S3CCS01	Data Structures and Applications	Dept.	3	0	0	3.5(48 hrs)	3	50	50	100	3
5.	PCCL	S3CCSL01	Data Structures and Applications Laboratory	Dept.	0	0	2		3	50	50	100	1
6.	ESC		ESC/ETC/PLC	Dept.	2	0	2	2.0(34hrs)	3	50	50	100	3
7.	UHV	SHS01	Social Connect and Responsibility **(Board: ME)**	Dept.	0	0	2		-	100	-	100	1
8.	AEC/ SEC		Ability Enhancement Course/ Skill Enhancement Course – III	Dept.	If offered as Theory Course				1½	50	50	100	1
					1	0	0	1.0(16 hrs)					
					If offered as Integrated Course				1½				
					0	0	2						
9.	NCMC	SMC01	National Service Scheme (NSS)	NSS CO	0	0	2			100	-	100	0
		SMC02	Physical Education (PE) (Sports and Athletics)	PED									
		SMC03	Yoga	PED									
			Total							**550**	**350**	**900**	**20**
		AAP	AICTE Activity Points (Applicable for both Regular and Lateral Entry students)	40 hours community service to be documented and produced for the examination									

Note: **PCC**: Professional Core Course, **IPCC**: Integrated Professional Core Course, **PCCL**: Professional Core Course laboratory, **UHV**: Universal Human Value Course, **NCMC**: Non Credit Mandatory Course, **AEC**: Ability Enhancement Course, **SEC**: Skill Enhancement Course, **ESC**: Engineering Science Course, **ETC**: Emerging Technology Course, **PLC**: Programming Language Course

L: Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation.

Engineering Science Course (ESC/ETC/PLC) (Offered by the Department)

S3CCSI03	Java Programming	S3CCSI05	Python Programming
S3CCSI04	Web Programming	S3CCSI06	Object Oriented Programming with C++

Ability Enhancement Course – III (Offered by the Department)

S3CSA01	Unix and Shell Programming	S3CSA03	Version controller with GiT
S3CSA02	MS Office and Latex	S3CSA04	Ethics and Public Policy for AI

Integrated Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical of the same course. Credit for IPCC can be 04 and its Teaching–Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor of Engineering (B.E.) 2022-23 may please be referred.

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG) with the concerned coordinator of the course during the first Week of III semesters. Activities shall be carried out between III semester to the VI semester (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the Degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the course is mandatory for the award of Degree.

SCHEME OF TEACHING AND EXAMINATION (160 Credits Scheme)

(Effective from the academic year 2024-2025)

IV Semester

B.E. in Computer Science & Engineering — Batch:2023-2024

Sl. No.	Course and Course Code		Course Title	Teaching / Paper setting Dept.	Teaching hrs./week				Examination				Credits
					Lecture	Tutorial	Practical/ Drawing	Self-Study Component	Duration in hrs.	CIE Marks	SEE Marks	Total Marks	
					L	T	P	S					
1.	PCC	S4CCS01	Design and Analysis of Algorithms	Dept.	3	0	0	3.5(48 hrs)	3	50	50	100	3
2.	IPCC	S4CSI01	Microcontroller and Embedded Systems (I)	Dept.	3	0	2	3.5(50 hrs)	3	50	50	100	4
3.	IPCC	S4CSI02	Theory of Computations (I)	Dept.	3	0	2	3.5(50 hrs)	3	50	50	100	4
4.	PCCL	S4CCSL01	Design and Analysis of Algorithms Lab	Dept.	0	0	2		3	50	50	100	1
5.	ESC		ESC/ETC/PLC	Dept.	3	0	0	3.5(48 hrs)	3	50	50	100	3
6.	BSC	S4CCA01	Biology for Engineers **(Board: BT)**	BT, CH, Phy, Che	3	0	0	3.5(48 hrs)	3	50	50	100	3
7.	UHV	SHS02	Universal Human Values Course **(Board: IEM)**	Dept.	1	0	0	1.0(16 hrs)	1½	50	50	100	1
8.	AEC/ SEC		Ability Enhancement Course/ Skill Enhancement Course – IV	Dept.	If offered as Theory Course				1½	50	50	100	1
					1	0	0	1.0(16 hrs)					
					If offered as Integrated Course				1½				
					0	0	2						
9.	NCMC	NS	National Service Scheme (NSS)	NSS CO	0	0	2			100	-	100	0
		PE	Physical Education (PE) (Sports and Athletics)	PED									
		YO	Yoga	PED									
			Total							**500**	**400**	**900**	**20**
		AAP	AICTE Activity Points (Applicable for both Regular and Lateral Entry students)	40 hours community service to be documented and produced for the examination									

Note: **PCC**: Professional Core Course, **IPCC**: Integrated Professional Core Course, **PCCL**: Professional Core Course laboratory, **UHV**: Universal Human Value Course, **NCMC**: Non Credit Mandatory Course, **AEC**: Ability Enhancement Course, **SEC**: Skill Enhancement Course, **ESC**: Engineering Science Course, **ETC**: Emerging Technology Course, **PLC**: Programming Language Course
L: Lecture, **T**: Tutorial, **P**: Practical **S= SDA**: Skill Development Activity, **CIE**: Continuous Internal Evaluation, **SEE**: Semester End Evaluation.

Engineering Science Course (ESC/ETC/PLC) (Offered by the Department)

S4CCS02	Discrete Mathematical Structures	**S4CCS04**	Linear Algebra
S4CCS03	Graph Theory	**S4CCS05**	Numerical Techniques

Ability Enhancement Course – IV (Offered by the Department)

S4CSA01	Java Script	**S4CSA03**	CUDA Programming
S4CSA02	Advanced Data Structures	**S4CSA04**	R Programming

Professional Core Course (IPCC): Refers to Professional Core Course Theory Integrated with practical of the same course. Credit for IPCC can be 04 and its Teaching–Learning hours (L : T : P) can be considered as (3 : 0 : 2) or (2 : 2 : 2). The theory part of the IPCC shall be evaluated both by CIE and SEE. The practical part shall be evaluated by only CIE (no SEE). However, questions from the practical part of IPCC shall be included in the SEE question paper. For more details, the regulation governing the Degree of Bachelor of Engineering (B.E.) 2022-23 may please be referred.

National Service Scheme /Physical Education/Yoga: All students have to register for any one of the courses namely National Service Scheme (NSS), Physical Education (PE)(Sports and Athletics), and Yoga(YOG) with the concerned coordinator of the course during the first Week of III semesters. Activities shall be carried out between III semester to the VI semester (for 4 semesters). Successful completion of the registered course and requisite CIE score is mandatory for the award of the Degree. The events shall be appropriately scheduled by the colleges and the same shall be reflected in the calendar prepared for the NSS, PE, and Yoga activities. These courses shall not be considered for vertical progression as well as for the calculation of SGPA and CGPA, but completion of the course is mandatory for the award of Degree.

Scheme of Teaching, Examination and Syllabus

B.E. COMPUTER SCIENCE & ENGINEERING

Batch: 2023-24

III SEMESTER

(Effective from the academic year 2024-2025)

B.E COMPUTER SCIENCE & ENGINEERING
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER - III

STATISTICS AND PROBABILITY

Course Code	S3MATC	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3	Exam Hours	03
Lecture Hours	40 Hrs	Practical Hour	-

Course objectives: The course will enable students to

1. Develop and conduct appropriate experimentation, analyze and interpret data and use engineering judgment to draw conclusion.
2. Understand the basic concepts and applications of probability in engineering.
3. Learn the random variable, random process and how to model the random processes in engineering.
4. Understand the multiple random variables and stochastic process.
5. Investigate the variability in sample statistics from sample to sample, measure of central tendency & dispersion of sample statistics and pattern of variability of sample.

UNIT-1 **(07 hrs)**

Introduction, Curve Fitting: Straight line, reducible to Linear and Quadratic form-parabola. Definition of Correlation and regression lines, formula for correlation coefficient, regression lines with proof and angle between the regression lines, Rank correlation..

UNIT-2 **(08 hrs)**

Basic terminology, Definition of probability, Probability and set notations, Types of events, Addition law of probability, conditional probability, multiplication law of probability, Baye's theorem.

UNIT-3 **(08 hrs)**

Definition **of** Random Variable, Discrete Probability distribution, expectation, Variance, Binomial distribution, Poisson distribution.

Continuous Probability distribution- expectation, Variance, Normal distribution and Exponential distributions.

UNIT-4 **(08 hrs)**

Joint probability distribution, Discrete and independent random variables, Expectation, Covariance, Correlation coefficient. Probability vectors, stochastic matrices, fixed point matrices, Regular stochastic matrices, Markov chains, Higher transition-probabilities, stationary distribution of regular Markov chains and absorbing states.

UNIT-5 **(09 hrs)**

Sampling Distribution: Introduction, Objectives, sampling distribution, testing of hypothesis, level of significance, confidence limits, simple sampling of attributes, test of significance of large samples, comparison of large samples, sampling of variables, central limit theorem, confidence limits for unknown mean, test of significance for means of two large samples, Sampling of variables – small samples, Student's t-distribution.

Course outcomes:

Upon completion of this course the student will be able to:

1. Apply least square method to fit a curve for the given data and evaluate the correlation coefficient and regression lines for the data. (L3).
2. Analyze the nature of the events and hence determine the appropriate probabilities of the events (L3).
3. Classify the random variables to determine the appropriate probability distributions and hence compute the associated probability. (L2).
4. Computes the joint probability and its parameters. Predicts the long run behavior of a

Markov chain using transition matrix (L3).
5. Estimate the parameters of a population and sample in testing of hypothesis (L2).

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	"Higher Engineering Mathematics"	B.S.Grewal	Khanna Publications	43rd edition 2015
2	Higher Engineering Mathematics	Ramana .B.V	Tata-McGraw Hill	latest edition 2016
Reference Books				
1	Advanced Engineering Mathematics	Erwin Kreyszig	Wiley Publications	10th Edition, 2015
2	Advanced Engineering Mathematics	C. Ray Wylie and Louis C. Barrett	Tata-McGraw Hill	6th Edition, 2005
3	Applied Mathematics for Engineers and Physicists	Louis A. Pipes and Lawrence R. Harvill	McGraw Hill	3rd edition 2014

Course Articulation matrix(CO-PO and CO-PSO mapping)

Course Outcomes	PROGRAM OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
CO1	2														
CO2		2	2												
CO3		2	2												
CO4		2	2												
CO5		2	2												
Overall CO	2	2	2												

B.E COMPUTER SCIENCE & ENGINEERING
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER - III

OPERATING SYSTEM (I)

Course Code	S3CCSI01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	03
Lecture Hours	40 Hrs	Practical Hour	26

Course objectives: This Course will enable students to:

1. **Define** fundamental OS abstractions such as processes, threads, files etc, (L1-knowlegde).
2. **Visualize** the intricate relationship between an operating system and its underlying hardware (L1-knowlegde).
3. **Explain** scheduling algorithms, deadlock detection algorithms and memory management strategies (L2-Comphrension).
4. A**pply** the principles of concurrency and synchronization, to write concurrent programs/ software (L3-Application).

UNIT-1 **(8L+2P)**

INTRODUCTION: What operating systems do - User view, System view, Defining operating systems, Operating System Structure, Operating System Operations – Dual mode and multi-mode operation, Timer, Process Management; Memory Management; Storage Management; Protection and Security. [1.1, 1.4 to 1.9]

SYSTEM STRUCTURES: Operating System Services; System calls; Types of system calls; System programs; Operating System Structure –Simple structure, Layered approach, Micro kernels, UNITs [2.1, 2.3 to 2.5, 2.7.1-2.7.4]

Self study : Hybrid Systems – Mac OS X, iOS, Android.[2.7.5]

UNIT-2 **(8L+6P)**

PROCESS: Process concept, Process state, Process control block, Process scheduling, Scheduling queues, Schedulers, Context switch, Operations on processes – Process creation and termination, Inter-process communication, Shared memory and message passing systems. [3.1 to 3.4]

PROCESS MANAGEMENT: Basic concepts, CPU scheduler, Preemptive and non-preemptive scheduling, Scheduling criteria, Scheduling algorithms – FCFS, SJF, Priority and Round robin scheduling, [Textbook 1: Chapters 5.1 to 5.3.4]

Self Study : Multi-level and multilevel feedback queue scheduling[5.3.5,5,3.6]

UNIT-3 **(8L+6P)**

THREADS: Overview, Benefits, Multi core Programming, Types of parallelism, Multi threading models. [4.1-4.3]

PROCESS SYNCHRONIZATION: Background, The Critical section problem, Peterson's solution, Synchronization hardware, Mutex locks, Semaphores, Classic problems of synchronization, Bounded buffer problem, Readers writer's problem, Dining philosopher's problem.
[Textbook 1: Chapters 6.1 to 6.7.3]

Self-Study : Monitors, Monitor Usage, Dining-Philosophers Solution Using Monitors. [6.8, 6.8.1, 6.8.2]

UNIT-4 **(8L+6P)**

DEADLOCKS: System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection and recovery from deadlock. [7.1-7.7]

MEMORY MANAGEMENT: Background, Basic hardware, Address binding, Logical and physical address, swapping, Dynamic loading and linking [8.1, 8.2]

UNIT-5 **(8L+6P)**

MEMORY MANAGEMENT: Contiguous memory allocation, Segmentation, Paging. [8.3, 8.4, 8.5]

VIRTUAL MEMORY MANAGEMENT: Basic concepts, Demand paging, Copy-on-write, Page

replacement – FIFO, LRU, Optimal [1:9.1-9.4] **Self-Study**: Structure of page table, Hierarchical paging, Hashed paging, Inverted paging. [8.6]
NOTE: Self-study topics are assessed only in CIE-Quizzes/ Assignment
Course outcomes: On successful completion of this course, students will be able to: 1. **Identify** the services, functions and structure of different operating systems. 2. **Apply and analyze** appropriate scheduling algorithm for process selection and execution. 3. **Identify and analyze** the techniques used to solve process synchronization issues. 4. **Apply and analyze** various deadlock prevention, avoidance, detection and recovery mechanisms to solve real world problems. 5. **Analyze** the performance of various memory management techniques and page replacement algorithms**.**

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Operating System Concepts	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne	Wiley-India,9th edition	2013
Reference Books				
1	Operating System - A Concept Based Approach,	D.M Dhamdhere	Tata McGraw-Hill	2nd Edition, 2002
2	Operating Systems,	P.C.P. Bhatt	PHI	4th Edition, 2013

Course Articulation matrix(CO-PO and CO-PSO mapping)

Course Outcomes	PROGRAM OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
CO1	2												2		
CO2		2	2										2		
CO3		2	2										2		
CO4		2	2										2		
CO5		2	2										2		
Overall CO	2	2	2										2		

Lab Syllabus:

Implementation of programs on the following Operating System concepts:

1. Threads
2. Process Scheduling.
3. Process Synchronization.
4. Deadlock Avoidance.
5. Memory allocation techniques.
6. Page Replacement Algorithms

Program articulation matrix

Course Outcomes	PROGRAMME OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
	2	2	2										2		

B.E COMPUTER SCIENCE & ENGINEERING
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER - III

DIGITAL CIRCUITS & COMPUTER ORGANIZATION (I)

Course Code	S3CCSI02	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	3
Lecture Hours	40 Hrs	Practical Hour	-

Course objectives: This Course will enable students to:

1. **Analyse** the basic structure of a computer and how computer programs are organized, stored and executed at the machine level
2. **Identify** the data path elements needed to implement single bus and three bus organization of a processor
3. **Design** control signal for of hardwired and micro programmed control
4. **Design** & implement different techniques used to perform arithmetic operations
5. **Illustrate** the basic types of memory and cache mapping functions

UNIT-1 **(8Hrs)**

Basic Structure of Computer: Functional Units, Basic Operational Concepts, Bus Structures, Performance - Processor Clock, Basic Performance Equation, Clock Rate, Performance Measurement. Machine Instructions and Programs: Numbers, Arithmetic Operations and Characters - Number Representation, Addition of Positive Numbers, Addition and Subtraction of Signed Numbers, Overflow in Integer Arithmetic, Characters, Memory Location and Addresses - Byte Addressability, Big-endian and Little-endian Assignments, Word Alignment, Accessing Numbers, Characters, and Character Strings, Memory Operations, Instructions and Instruction Sequencing - Register Transfer Notation, Assembly Language Notation, Basic Instruction Types, Instruction Execution and Straight-Line Sequencing, Branching, Condition Codes.

Chapter1:1.1, 1.2, 1.3, 1.4, 1.6, 1.61, 1.62, 1.64, 1.67. Chapter2:2.1, 2.2, 2.3, 2.4.1 to 2.4.6.

UNIT-2 **(8Hrs)**

Addressing Modes - Implementation of Variables and Constants, Indirection and Pointers, Indexing and Arrays, Relative Addressing, Additional Modes, Basic Input and Output Operations. Stacks and Queues, Subroutines - Subroutine Nesting and the Processor Stack, Parameter Passing, The Stack Frame , Basic Processing Unit: Some Fundamental Concepts – Single Bus Organization: Register Transfers, Performing an Arithmetic or Logic operation, Fetching a Word from Memory, Storing a word in Memory.

Chapter2: 2.5, 2.7, 2.8, 2.9. Chapter 7: 7.1.

UNIT-3 **(8Hrs)**

Basic Processing Unit: Execution of a Complete Instruction - Branch Instructions, Multiple Bus Organization, Hard wired Control - A Complete Processor, Micro programmed Control - Microinstructions. Arithmetic: Addition and Subtraction of Signed Numbers - Addition/Subtraction Logic Unit, Design of Fast Adders - Carry-Lookahead Addition.

Chapter7: 7.2-7.4, 7.5.1.

UNIT-4 **(8Hrs)**

Arithmetic: Multiplication of Positive Numbers, Signed Operand Multiplication - Booth Algorithm, Fast Multiplication - Bit-Pair Recoding of Multipliers, Carry-Save Addition of Summands, Integer Division, Floating-point Numbers and Operations - IEEE Standard for Floating-Point Numbers, Arithmetic Operations on Floating-Point Numbers - Addition and Subtraction Operations, Implementing Floating-Point Operations.

Chapter6: 6.1 -,6.7.

UNIT-5 (8Hrs)

Memory System: Some Basic Concepts, Semiconductor RAM Memories - Internal Organization of Memory Chips, Static Memories, Asynchronous DRAMs, Synchronous DRAMs, Structure of Larger Memories, Memory System Considerations, Read Only Memories - ROM, PROM, EPROM, EEPROM, Flash Memory, Speed, Size, and Cost, Cache Memories -Mapping Functions
Chapter5: 5.1-5.4, 5.5.1, 5.5.2.

LAB COMPONENT

Week 1 : Introduction to digital trainer kits & verification of basic gates

Week 2 onwards in every lab the instructions and design of the following experiments to be taught during the first one hour of the lab. The second hour is to be utilized in conducting the experiments and verification of truth tables.

1. Design and implementation of a Half- adder and a full adder using minimum number of 2 input NAND gates
2. Given any 4-variable logic expression, simplify using Entered Variable Map and realize the simplified logic expression using 8:1 or (2) 4:1 Multiplexer IC.
3. Design and implement Full Adder and Full Subtractor using 4:1 MUX.
4. Design and implement full-adder and full-subtractor using a 74138 DECODER.
5. Design and test one/ two-bit Magnitude Comparator and verify its true table.
6. Realize a J-K Master/Slave Flip-Flop using NAND gates and verify its truth table.
7. Design and implement a mod-n (n<8) Synchronous up Counter using J-K Flip-Flop Ics, display the result on discrete LEDs.
8. Design and implement the following using 4-bit Shift register IC.
   i) Left Shift ii) SIPOiii) SISO iv) PIPO v) Ring Counter vi) Johnson counter
9. Design and implement an Asynchronous Counter using Decade Counter IC to Count up from 0 to 9. Display the count value on 7 Segment LED display using BCD to 7 segment code converter IC.
10. Design and implement a 3 stage Asynchronous Counter using a J-K Flip Flops to count from 0 to n.

Course outcomes:

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

1. **Illustrate** the basic operational concepts of a computer system and **discuss** its performance parameters
2. **Interpret** various addressing modes and **apply** the same to **design** solution to a given problem
3. **Discuss** basic processing unit to generate control signals and to **design** the control sequence for execution of an instruction
4. **Explain** the various arithmetic algorithms and **apply** the same to solve a given problem
5. **Describe** memory organization and **design** the solution to the given problem

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Computer Organization	Carl Hamacher, Zvonko	TMH	2005
Reference Books				
1	Computer Organization & Architecture	William Stallings.	PHI	2006

2	Computer Systems Design and Architecture	Vincent P. Heuring & Harry F. Jordan	PEARSON	2004

COURSE ARTICULATION MATRIX (CO-PO AND CO-PSO MAPPING)

COs	POs												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	2												2		
CO2		2	1										2		
CO3	1		2										2		
CO4	1	2											2		
CO5	2		2										2		
Overall CO	**2**	**2**	**2**										**2**		

PROGRAM ARTICULATION MATRIX:

COs	POs												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO	2	2	2										2		

B.E COMPUTER SCIENCE & ENGINEERING
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER - III

DATA STRUCTURES AND APPLICATIONS			
Course Code	S3CCS01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3.0	Exam Hours	3
Lecture Hours	40 Hrs	Practical Hour	-

Course objectives: This Course will enable students to:

1. Discuss the concepts of structures, union, files and dynamic memory allocation techniques.
2. Describe the properties of various data structures such as Stacks, Queues, Lists, and Trees.
3. Implement the data structures such as Stacks, Queues, Lists, and Trees using C language.
4. Discuss the applications of various Data Structures.

UNIT-1 **(08L hrs)**

Structures and Unions: Defining a Structure, declaring Structure variables, accessing Structure members, Structure initialization, copying and comparing Structure variables, operations on individual members, array of Structures, array within Structure, Structure within Structure, Structures and Functions, Unions, size of structures.

File management in C: Defining and Opening a file, Closing a file, Input/Output operations on files - getc(), putc(), getw(), putw(), fscanf(), fprintf(), Error handling during I/O operations - feof(), ferror(), Random access to files - ftell(), rewind(), fseek(), Command line arguments.

(Text Book 1: 10, 12)

UNIT-2 **(08L hrs)**

The Stack: Definition and Examples, representing Stacks in C, Example: Infix, Postfix, and Prefix.

Recursion: Recursive Definition and Processes, Recursion in C, Writing recursive programs: The Towers of Hanoi Problem, Efficiency of Recursion.

Queues and Lists: The Queue and Its Sequential Representation: C implementation of Queues, Insertion, Deletion and Display operations, Types of Queues (Linear and Circular Queues)

Self-Study: Priority and Double Ended Queues (Only concepts).

(Text Book2: 2, 3.1, 3.2, 3.3(only the Towers of Hanoi Problem), 3.5. 4.1(excluding Queue as an ADT))

UNIT-3 **(08L hrs)**

Queues and Lists Continued

Dynamic memory allocation: malloc(), calloc(), realloc(), free().

(Text Book 1: 13.1-13.6)

Linked lists: Inserting and removing nodes from a list, linked implementation of stacks, getnode and freenode operations, linked implementation of queues, examples of list operation, list implementation of priority queues, header nodes.

Lists in C: allocating and freeing dynamic variables, linked lists using dynamic variables, queues as lists in C, examples of list operations in C, non-integer and non-homogeneous lists, Addition of two polynomials, implementing header nodes.

(Text Book2: 4.2, 4.3(except array implementation of list, Limitations of array implementation, comparing dynamic and array implementations of list))

UNIT-4 **(07L hrs)**

Other List Structures: Circular lists, stack as a Circular list, queue as a Circular list, primitive operations on circular lists, the Josephus problem, Doubly linked lists, Primitive operations on Doubly linked list.

(Text Book2: 4.5(except addition of long positive integers using circular and doubly linked list))

UNIT-5 **(09L hrs)**

Trees: Operations on Binary Trees, Applications of Binary Trees, Binary Tree Representations: Node representation of Binary Trees, Internal and External Nodes, Implicit array representation of Binary Trees, Binary Tree Traversals in C.
Trees and Their applications: C Representations of Trees, Tree Traversals, General Expressions as Trees, Evaluating an Expression Tree, Constructing a Tree.
Self-Study: Threaded Binary Trees - definition and types.
(Text Book2: 5.1, 5.2, 5.5(except choosing Binary Tree Representation, Traversal using a Father field, Heterogeneous Binary Trees))

Course outcomes:
On successful completion of this course, students will be able to:

1. **Apply** advanced C programming techniques like pointers, structures, union and files to **develop** solution for a given problem
2. **Discuss** and **implement** different linear data structures like stacks and queues using static memory allocation technique
3. **Discuss** different types of linked lists and **implement** using dynamic memory allocation technique
4. **Discuss** non-linear data structures like trees and **implement** using dynamic memory allocation technique.
5. **Apply** the knowledge of stacks, queues, linked lists and trees to **design** and **develop** solutions to given problems

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Programming in ANSI C	E. Balagurusamy	Tata McGraw-Hill Publications	7th Edition, 2017
2	Data structures using C and C++	YedidyahLangsam, Moshe J. Augenstein, Aaron M. Tenenbaum	PHI/Pearson	2nd Edition,2015
Reference Books				
1	Fundamentals of Data Structures in C	Horowitz, Sahni and Anderson-Freed	Universities Press Pvt. Ltd.	2nd Edition,2011
2	An Introduction to Data Structures with Applications	Jean- Paul Tremblay Paul G. Sorenson	McGraw-Hill International	2nd Edition,2007

COURSE ARTICULATION MATRIX (CO-PO AND CO-PSO MAPPING)

COs	POs												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	2	2	2											2	
CO2	2		2											2	
CO3	2	2	2											2	
CO4	2		2											2	
CO5	2	2	2											2	
Overall CO	**2**	**2**	**2**											**2**	

B.E COMPUTER SCIENCE & ENGINEERING
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER - III

DATA STRUCTURES AND APPLICATIONS LABORATORY

Course Code	S3CCSL01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	3
Lecture Hours	-	Practical Hour	26 Hrs

Course objectives: This Course will enable students to:

1. Design and implement different data structures.
2. Develop C programs for various applications of data structures.
3. Select appropriate data structure for a given problem.

Sl. no.	Programs
1	Develop a C program to create a sequential file for storing employee records with each record having following information: **Employee_Id**: Non-Zero Positive integer; **Name**: 25 Characters; **Department**: 25 Characters; **Salary**: Positive Integer; **Age**: Positive integer Write necessary functions to perform the following operations: a) Read the details of a record. b) Display all the records in the file. c) Search for a specific records based on Department. In case if the required record is not found, suitable message should be displayed.
2	Develop a C program to implement Stack of names to perform the push, pop and display operations.
3	Develop a C program to convert a valid infix expression to postfix.
4	Develop a C program to evaluate the given postfix expression.
5	Develop a C program to implement Linear Queue of characters to perform the insertion, deletion and display operations.
6	Develop a C program to implement Circular Queue of integers to perform the insertion, deletion and display operations.
7	Define a structure to represent a node in a Singly Linked List. Each node must contain following information: player name, team name and batting average. Develop a C program using functions to perform the following operations on a list of cricket players: a) Add a player at the end of the list. b) Search for a specific player and update his/her batting average if the player exists. c) Display the details of all the players.
8	Develop a C program to add two two-variable polynomials using Singly Linked list.
9	Develop a C program to construct two ordered singly linked lists using functions to perform following operations: a) Insert an element into a list. b) Merge the two lists. c) Display the contents of the list.
10	Define a structure to represent a node in a Linear Doubly Linked List. Each node must contain following information: Student name, USN, branch and year of admission. Develop a C program using functions to perform the following operations on a list of students: a) Add a student at the beginning of the list. b) Display the details of the students of a specified branch.

	c) Delete the student with specified USN. d) Display the details of all the students.
11	Develop a C program to implement Josephus problem using Circular Singly Linked List. Write necessary functions to perform the following operations: a) Add a soldier to the list. b) Delete a soldier from the list.
12	Develop a C program to perform the following operations: a) Construct a binary search tree of integers. b) Traverse the tree in Inorder. c) Delete a given node from the BST.
13	Develop a C program to construct an expression tree for a given postfix expression and evaluate the expression tree.

Open Ended Problems

These problems are introduced to make the students to apply the knowledge of Data Structures in solving real world problems. Following are the guidelines:

- Each team (3/4 students) from each batch should come up with the problem statement for an application of any of the data structures like files, stacks, queues, linked lists and trees.
- Faculty-incharge approves the problem based on the complexity of the problem chosen.
- Each team has to implement the problem statement within the deadline.

Implementation will be considered for Continuous Internal Evaluation (CIE) and it will be based on individual contribution of the students in each team.

Course outcomes:

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

1. **Design** and **develop** C programs by applying advanced C programming techniques like pointers, structures and files to solve a given problem
2. **Design** and **develop** C programs to implement linear data structures like stack, queue and explore its applications by **applying** the knowledge of static memory allocation technique
3. **Design** and **develop** C programs to implement linked lists and its types by applying the knowledge of dynamic memory allocation technique
4. **Apply** the knowledge of dynamic memory allocation technique to implement non-linear data structures like trees and to **design** and **develop** solutions for applications on trees

Conduct of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Breakup of marks and the instructions printed on the cover page of answer script to be strictly adhered by the examiners.
- Students can pick one experiment from the questions lot prepared by the examiners.
- Change of experiment is allowed only once and 20% Marks is to be deducted.

COURSE ARTICULATION MATRIX (CO-PO AND CO-PSO MAPPING)

COs	POs												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	2	2	2											2	
CO2	2		2											2	
CO3	2	2	2											2	
CO4	2		2											2	
CO5	2	2	2											2	
Overall CO	**2**	**2**	**2**											**2**	

B.E COMPUTER SCIENCE & ENGINEERING
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER - III

JAVA PROGRAMMING

Course Code	S3CCSI03	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:2)	SEE Marks	50
Credits	3	Exam Hours	3
Lecture Hours	26 Hrs	Practical Hour	26Hrs

Course objectives: This Course will enable students to:

1. Understand the fundamentals of object-oriented programming in Java, including defining classes, Objects, invoking methods
2. Set up Java JDK environment to create, debug and run simple Java programs.
3. Understand the principles of inheritance, packages and interfaces.
4. Understand generic programming and implement generic classes and methods.
5. Design and develop reliable Object oriented programs

UNIT-1 **(8 hours)**

AN OVERVIEW OF JAVA: Object-Oriented Programming, A First Simple Program, A Second Short Program

Classes, Objects and Methods; Inheritance Classes, Objects and Methods-Introduction, Defining a Class, Fields Declaration, Methods Declaration, Creating Objects, Accessing Class Members, Constructors, Methods Overloading, Static Members. **Inheritance:** Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance.

UNIT-2 **(8 hours)**

Packages and Interfaces

Packages: Defining a Package, Finding Packages and CLASSPATH, A Short Package Example, Access protection: An Access Example, Importing Packages.

Interfaces: Defining an Interface, Implementing Interfaces, Nested Interfaces, Applying Interfaces, Variables in Interfaces, Default Interface Methods: A More Practical Example, Multiple Inheritance Issues, and Use Static Methods in an Interface.

UNIT-3 **(8 hours)**

Exception Handling and Generics

Exception Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch: Displaying a Description of an Exception, Multiple Catch Clauses, throw, throws, finally.

What Are Generics? A Simple Generics Example, The General Form of a Generic Class, Creating a Generic Method, Generic Constructors, Some Generic Restrictions.

UNIT-4 **(8 hours)**

Programming with I/O, Applets

I/O Basics, Streams, Byte Streams and Character Streams, The Predefined Streams, Reading Console Read the values, Reading Characters, Reading Strings, Writing Console Output, The PrintWriter Class, Reading and Writing Files, Automatically Closing a File. Applet Fundamentals

The Applet Class:- Two Types of Applets, Applet Basics, The Applet Class, An Applet Skeleton, Requesting Repainting, Using the Status Window, The HTML APPLET Tag

UNIT-5 **(8 hours)**

Event Handling, Introducing the AWT: Working with Windows, Graphics, and Text

Event Handling: Using the Delegation Event Model- Handling Mouse Events

Introducing the AWT: Working with Windows, Graphics, and Text: AWT Classes: Window Fundamentals, Component, Container, Panel, Window, Frame, Canvas, Working with Frame Windows, Creating a Frame Window in an Applet, Handling Events in a Frame Window. Working with Graphics: Drawing Lines, Rectangles etc, Working with Color, working with Fonts.

Course outcomes:

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

1. Discuss the Object Oriented Programming concepts and apply the same to design programs.
2. Design and implement object oriented solutions involving multiple objects, packages & Interfaces.
3. Develop simpler, reliable and reusable programs using exception handling and Generics
4. Illustrate the versatility of I/O Operations in programs
5. Design and develop Web applications using Java AWT packages

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2009	Herbert Schildt	Tata McGrawhill	9th Edition,2009
Reference Books				
1	Object-Oriented Programming With JAVA Essentials and Applications	RajKumar Buyya,	McGraw Hill	2009
2	WEB LINKS AND VIDEO LECTURES (e-RESOURCES) :	https://onlinecourses.nptel.ac.in/noc22_cs47/preview http://www.mhhe.com/buyya/oopj		

Sl. No.	Experiments
1	Write a JAVA program to sort list of elements in ascending and descending order using bubble sort.
2	Create a JAVA class called Student with the following details as variables within it. USN, NAME, BRANCH, PHONE, PERCENTAGE Write a JAVA program to create n Student objects and print the USN, Name, Branch, Phone, and percentag of these objects with suitable headings.
3	Write a JAVA program demonstrating Method overloading and Constructor overloading.
4	Design a super class called Staff with details as Staff ID, Name, Phone, and Salary. Extend this class by writing three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a JAVA program to read and display at least 3 staff objects of all three categories.
5	Demonstrate dynamic dispatch using abstract class in JAVA.
6	Create two packages P1 and P2. In package P1, create class A, class B inherited from A, class C. In package P2, create class D inherited from class A in package P1 and class E. Demonstrate working of access modifiers (private, public, protected, default) in all these classes using JAVA.
7	Write a java program to perform simple command line calculator with an exception handler.
8	Develop a Program to Launch the browser and open a specific URL
9	Write a Java program that reads a text file and displays the contents on the screen.
10	Write a Java program to check whether the given element is present in a given array or not using generic method.
11	Create an AWT-based application that creates a frame window and responds to mouse clicks and keystrokes
12	Develop an applet that constructs several colors and draws various objects using these colors and also Display various Fonts

COURSE ARTICULATION MATRIX (CO-PO AND CO-PSO MAPPING)

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

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

WEB PROGRAMMING

Course Code	S3CCSI04	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:2)	SEE Marks	50
Credits	3	Exam Hours	03
Lecture Hours	26 Hrs	Practical Hour	26 Hrs

Course objectives: This Course will enable students to:

1. To use the syntax and semantics of HTML and XHTML
2. To develop different parts of a web page
3. To understand how CSS can enhance the design of a webpage.
4. To create and apply CSS styling to a webpage
5. To understand the JavaScript fundamentals

UNIT-1 **(5L+ 4P)**

Fundamentals of WWW: A Brief Introduction to the Internet, TheWorld Wide Web, WebBrowsers,WebServers,Uniform ResourceLocators, client-server architecture, difference between static and dynamic web pages.

Traditional HTML and XHTML:

First Look at HTML and XHTML, Hello HTML and XHTML World, HTML and XHTML: VersionHistory, HTML and XHTML DTDs: The Specifications Up Close, (X)HTML Document Structure,Browsers and (X)HTML, The Rules of (X)HTML, **TextBook1:Chapter1**

UNIT-2 **(5L+ 6P)**

HTML5:

Hello HTML5, Loose Syntax Returns, XHTML5, HTML5: Embracing the Reality of Web Markup,PresentationalMarkupRemovedandRedefined,HTML5DocumentStructureChanges,AddingSemantics,HTML5'sOpenMediaEffort,Client-SideGraphicswith<canvas>,HTML5FormChanges,EmergingElements and Attributes to SupportWeb Applications

TextBook1:Chapter2

UNIT-3 **(5L+ 6P)**

Cascading Style Sheets(CSS):

Introduction, CSS Overview , CSS Rules, Example with Type Selectors and the Universal Selector,CSSSyntaxandStyle,ClassSelectors,IDSelectors,spananddivElements,Cascading,styleAttribute, style Container, External CSS Files, CSS Properties, Color Properties, RGB Values forColor,OpacityValuesforColor,HSLandHSLAValuesforColor,FontProperties,line-height Property, TextProperties, BorderProperties, ElementBox, paddingProperty,marginProperty,CaseStudy:DescriptionofaSmallCity'sCoreArea.

Bootstrap: Introduction to Bootstrap, Why use Bootstrap, Bootstrap Examples-Tables, forms, nav menu,
Breakpoints, poppers.

TextBook2-: Chapter3, https://getbootstrap.com/

UNIT-4 **(5L+ 4P)**

TablesandCSS,LinksandImages

Table Elements, Formatting a Data Table: Borders, Alignment, and Padding, CSS Structural Pseudo-Class Selectors, thead and tbody Elements, Cell Spanning, Web Accessibility, CSS display Property withTable Values, a Element,Relative URLs,NavigationWithina Web Page, CSSforLinks , Responsive Images, Positioning Images,ShortcutIcon, iframeElement .

TextBook2:5.2to 5.8, 6.2,6.3, 6.6., 6.7, 6.9,6.10, 6.12, 7.2 to 7.4

UNIT-5 (6L+6 P)

IntroductiontoJavaScript:Functions,DOM,Forms,andEventHandlers

AssignmentStatements and Objects, Document Object Model, Forms and How They're Processed: Client-SideVersus Server-Side, form Element, Controls, Text Control, Accessing a Form's Control Values, reset and focus Methods, handling errors in javascript.

TextBook2:8.2 to8,13,8.15, 8.16

Form Element, Controls, Text Control, Email Address Generator web page, Event Handler Attributes, onchange, onmouseover, onmouseevent, Using noscript to Accomodate Disabled javascript (Chapter 8.11 to8.18, 8.20)

LAB COMPONENT

1. Create an XHTML page using tags to accomplish the following:
   (i) A paragraph containing text "All that glitters is not gold". Bold face and italicize this text
   (ii) Create equation: $x = 1/3(y_1^2 + z_1^2)$
   (iii) Put a background image to a page and demonstrate all attributes of background image
   (iv) Create unordered list of 5 fruits and ordered list of 3 flowers

2. Using MathML, write the mathematical expressions for the following. Use separate div for the equations. Insert suitable title, background colour, text colour for each div.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \overline{\qquad} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$^{2} - 4ac$$

$$l$$

3. Create following table using XHTML tags. Properly align cells, give suitable cell padding and cell spacing, and apply background color, bold and emphasis necessary

Department	**Sem1**	*SubjectA*
		SubjectB
		SubjectC
	Sem2	*SubjectE*
		SubjectF
		SubjectG
	Sem3	*SubjectH*
		SubjectI
		SubjectJ

3.Use HTML5 for performing following tasks:

(i) Draw the following shapes using HTML5 SVG: Square: fill the square with green color and make 6px brown stroke width. Rectangle: Fill the rectangle with blue color and make 4px black stroke width. circle using HTML5 SVG. Ellipse: fill the ellipse with green color and make 3px brownstroke width

(ii)Write the following mathematical expression by using HTML5 MathML.d=x^2-y^2

(iii)Redirecting current page to another page after 5 seconds using HTML5 meta tag

4.Demonstrate the following HTML5 Semantic tags- <article>, <aside>, <details>, <figcaption>, <figure>, <footer>, <header>, <main>, <mark>, <section> for a webpage that gives information about travel experience.

5.Create a class called **income**, and make it a background color of #0ff. Create a class called **expenses**, and make it a background color of #f0f. Create a class called **profit**, and make it a background color of #f00.

Throughout the document, any text that mentions income, expenses, or profit, attach theappropriate class to that piece of text. Further create following line of text in the same document:
~~The current price is 50₹~~ and new price is 40₹

6.Change the tag **li** to have the following properties:

- A display status of inline
- A medium, double-lined, black border
- No list style type

Add the following properties to the style for **li**:

- Margin of 5px
- Padding of 10px to the top, 20px to the right, 10px to the bottom, and 20px tothe left

Also demonstrate list style type with user defined image logos

7.Create sign up web page using HTML and CSS with tabular layout

8.Create calculator interface with HTML and CSS

9. Design a BMI calculator using HTML, CSS and Javascript. Inputs are Height and Weight.

10.Write a Java Script program that on clicking a button, displays scrolling text which moves from left to right with a small delay

11.Create a webpage containing 3 overlapping images using HTML, CSS and JS. Further when the mouse is over any image, it should be on the top and fully displayed.

Course outcomes:

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

1. To use the syntax and semantics of HTML and XHTML
2. To apply HTML5 Tags, forms, and graphics in the web application design
3. To apply CSS attributes and properties to a webpage
4. To design website using Bootstrap components and apply Pseudo-Class Selectors
5. Implement core constructs and event handling mechanisms of JavaScript.

.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	TextBook-1: HTML & CSS: The Complete Reference	Thomas A. Powell	TataMcGraw Hill	Fifth Edition
2	WEB PROGRAMMING with HTML5, CSS and JavaScript	John Dean, Jones & Bartlett Learning	Jones & Bartlett Learning	First Edition
Reference Books				
1	Programming the World Wide Web	Robert W Sebesta	Pearson Education	Seventh Edition 2017

2	HTML:A Beginner's Guide	WendyWillard	McGraw-HillEducation	Fourth Edition, 2009
3	HTML & CSS: The Complete Reference	ThomasA.Powell	TataMcGraw Hill,	Fifth Edition, 2010

Course Articulation matrix (CO-PO and CO-PSO mapping)

Course Outcomes	PROGRAMME OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
CO1	2		2		2				2	2				2	
CO2			2		2				2	2				2	
CO3		2	2		2				2	2				2	
CO4		2	2		2				2	2				2	
CO5		2	2		2				2	2				2	
Overall CO	2	2	2		2				2	2				2	

Program articulation matrix:

Course Outcomes	PROGRAMME OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
CO	2	2	2		2				2	2				2	

Degree of compliance **1**: Low **2**: Medium **3**: High

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

PYTHON PROGRAMMING

Course Code	S3CCSI05	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:2)	SEE Marks	50
Credits	3.0	Exam Hours	03
Lecture Hours	26 Hrs	Practical Hour	26 Hrs

Course objectives: This Course will enable students to:

1. Implement Python programs using Python language construct
2. Understand various data structures provided by Python library
3. Use different libraries for scientific and data intensive applications
4. Build real-world applications using OOP, files and exception handling provided byPython
5. Determine the need for scraping websites and working with CSV, JSON and other file formats.

UNIT-1 **(5L+4P)**

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 UNITs, 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, A Short Program: Guess the Number

Chapter1, Chapter2, Chapter3 (Automate the Boring Stuff with Python by Al Sweigart)

UNIT-2 **(5L+4P)**

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,

Manipulating Strings: Working with Strings, Useful String Methods

Chapter4, Chapter5, Chapter6 (Automate the Boring Stuff with Python by Al Sweigart)

UNIT-3 **(6L+6P)**

Pattern Matching with Regular Expressions: Finding Patterns of Text Without Regular Expressions, Finding Patterns of Text with Regular Expressions, More Pattern Matching with Regular Expressions, Greedy and Non greedy Matching, The findall() Method, Character Classes, Making Your Own Character Classes, The Caret and Dollar Sign Characters, The Wildcard Character, Review of Regex Symbols, Case-Insensitive Matching, Substituting Strings with the sub() Method, Managing Complex Regexes, Combining re .IGNORECASE, re .DOTALL, and re .VERBOSE,

Reading and Writing Files: Files and File Paths, The os.path UNIT, The File Reading/Writing Process, Saving Variables with the shelve UNIT,Saving Variables with the pprint.pformat() Function, Project: Generating Random Quiz Files,

Organizing Files: The shutil UNIT, Walking a Directory Tree, Compressing Files with the zipfile UNIT, Project: Renaming Files with American-Style Dates to European-Style Dates

Debugging: Raising Exceptions, Getting the Traceback as a String, Assertions, Logging, IDLE's Debugger.

Chapter7, Chapter9, Chapter10 & Chapter11 (Automate the Boring Stuff with Python by Al Sweigart)

UNIT-4 **(5L+6P)**
Classes and objects: Programmer-defined types, Attributes, Rectangles, Instances as return values, Objects are mutable, Copying, **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 Chapter11 (Introduction to Python Programming by Gowrishankar S, Veena A)
UNIT-5 **(5L+6P)**
Web Scraping: Project: MAPIT.PY with the web browser UNIT, Downloading Files from the Web with the requests UNIT, Saving Downloaded Files to the Hard Drive, HTML, Parsing HTML with the BeautifulSoup UNIT **Working with Excel Spread sheets:** Excel Documents, Installing the open pyxl UNIT, Reading Excel Documents, Project: Reading Data from a Spread sheet, Writing Excel Documents **Working with CSV files and JSON data:** The csv UNIT, Project: Removing the Header from CSV Files, JSON and APIs, The json UNIT, Project: Fetching Current Weather Data Chapter12, Chapter13 & Chapter16 (Automate the Boring Stuff with Python by Al Sweigart)

Course outcomes:

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

1. Demonstrate proficiency in handling of loops and creation of functions.
2. Identify the methods to create and manipulate lists, tuples and dictionaries.
3. Discover the commonly used operations involving regular expressions and file system.
4. Interpret the concepts of Object-Oriented Programming as used in Python.
5. Determine the need for scraping websites and working with CSV, JSON and other file formats.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Automate the Boring Stuff with Python	Al Sweigart	No Starch Press	1st Edition & 2015
Reference Books				
1	Introduction to Python Programming	Gowrishankar S, Veena A	CRC Press/Taylor & Francis	1st Edition & 2018
2	Introduction to Computer Science Using Python	Charles Dierbach	Wiley India Pvt Ltd	1st Edition & 2015

Programming Assignments:
Programs on basic concepts of python.
Programs on Strings
Programs on lists, tuples and dictionaries.
Programs on regular expressions.
Programs on exception handling.
Programs on files operations.
Programs on Classes and objects.
Programs on Web-Scrapping
Programs to work with CSV
Programs to work with JSON and other file formats

Conduct of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Breakup of marks and the instructions printed on the cover page of answer script to be strictly adhered by the examiners.
- Students can pick one experiment from the questions lot prepared by the examiners.
- Change of experiment is allowed only once and 20% Marks is to be deducted.

Course Articulation matrix (CO-PO and CO-PSO mapping)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2
CO 1	3		3											2
CO 2	3		3											2
CO 3	3		3											2
CO 4	3		3											2
CO 5	2		2											2

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

OBJECT ORIENTED PROGRAMMING WITH C++

Course Code	S3CCSI06	CIE Marks	50
Teaching Hours/Week (L:T:P)	(2:0:2)	SEE Marks	50
Credits	3	Exam Hours	3
Lecture Hours	26 Hrs	Practical Hour	26 Hrs

Course objectives: This Course will enable students to:

- The course provides the basic principles of object-oriented programming using C++.
- The course introduces the following topics such as classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, templates.
- The course briefly covers C++ implementation and object-oriented considerations for software design and reuse.

UNIT-1 **(6L+4P Hrs)**

PRINCIPLES OF OBJECT ORIENTED PROGRAMMING: A look at procedure Oriented programming, Object Oriented Programming paradigm, Basic concepts of OOP, Benefits of OOP, A sample program, structure of C++ program.

TOKENS, CONTROL STRUCTURES: Tokens, keywords, identifiers & constants, symbolic constants, reference variables, operators in C++, Scope Resolution Operator, Memory management operators, manipulators.

FUNCTIONS IN C++: The main(), function prototyping, Inline function, Default arguments, const arguments, function overloading

Book1:[1.3,1.4,1.5,1.6,2.3,2.5,2.6,3.2,3.3,3.4,3.93.13,3.15,3.17,3.18,4.2,4.3,4.6,4.7,4.10]

UNIT-2 **(6L+6P Hrs)**

CLASSES AND OBJECTS: C structures, specifying class, member functions, Inline functions, nesting of member function, private member functions, arrays within a class, memory allocation for objects, static data members and member functions, arrays of objects, objects as function arguments, Friendly functions, returning objects.

CONSTRUCTORS AND DESTRUCTORS: Introduction, constructors, parameterized constructors, multiple constructors in a class, constructors with default arguments, copy constructors, and destructors**.**

Book1:[5.3,5.4,5.6,5.7,5.8,5.9,5.10,5.11,5.12,5.13,5.14,5.15,5.16,6.1,6.2,6.4,6.7,6.11]

UNIT-3 **(5L+4P Hrs)**

OPERATOR OVERLOADING: Defining operator overloading, overloading unary and binary operators, overloading using friends, Rules for overloading operators.

TYPE CONVERSIONS: Basic to Class type, class to basic type, one class to another class type, A data conversion example.

INHERITANCE: Introduction, defining derived classes, single inheritance, making private member inheritable, multilevel, multiple, hierarchical, hybrid inheritance, virtual base classes

Book1:[7.2,7.3,7.4,7.5,7.7,7.8,7.9,8.1,8.2,8.3,8.4,8.5,8.6,8.7,8.8,8.9]

UNIT-4 **(5L+6P Hrs)**

VIRTUAL FUNCTIONS AND POLYMORPHISM: 'this' pointer, Pointer to derived classes, virtual function, pure virtual functions.

TEMPLATES: class templates, class templates with multiple parameters, function templates,

function templates with multiple parameters, overloading of template functions, member function templates, Non-type template arguments.
Book1:[9.4,9.5,9.6,9.7,12.1,12.2,12.4,12.5,12.6,12.7]

UNIT-5 (6L+6P Hrs)

MANAGING CONSOLE I/O OPERATIONS:
C++ stream classes, unformatted I/O operations, Formatted console I/O operations.

WORKING WITH FILES: Opening and Closing a File, detecting EOF, More about Open(): File modes, File pointers and their manipulations, sequential and random access.

EXCEPTION HANDLING: Introduction, Basics of Exception handling, Exception Handling Mechanism, Throwing Mechanism, Catching Mechanism.
Book1:[10.3,10.4,10.5,11.3,11.4,11.5,11.6,11.7,13.1,13.2,13.3,13.413.5,13.6]

Course outcomes:
On successful completion of this course, students will be able to:

1. **Apply** the object-oriented programming concepts to solve real world problems
2. **Develop** and **demonstrate** the different overloading techniques.
3. **Develop** solutions for real world problems using inheritance and polymorphism concepts.
4. Develop generic programming skills using templates and programs to perform I/O operations using file handling.
5. **Apply** the exception handling methodology for handling errors.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbook				
1	Object Oriented Programming with C++,	E Balagurusamy	Tata McGraw Hill	5th edition, ISBN: 9781259029936
Reference Books				
1	The Complete reference C++,	Herbert Schildt	Tata McGraw Hill	4th Edition
2	Object Oriented Programming with C++	Robert Lafore	SAMS Pearson Education	4th Edition
3	C++ Primer	Stanley B. Lippman,	Addison Wesley	4th edition, 2005
4	Object- Oriented Programming with C++	Sourav Sahay	Oxford University Press	1st edition, 2009

Sl. no.	Programming Assignments:
1	Develop a program to swap two numbers using reference variable and function swap ().
2	Develop a C++ program to compute the area of circle, rectangle and triangle (given with 3 sides) by overloading the area( ) function.
3	Develop a C++ program to create a class FLOWER with following characteristics: Name, Colour, Price. Display the names of all flower costing more than 25 rupees.
4	Develop a C++ program to create a class POINT with two floating point data members and illustrate the concept of default constructor, parameterized constructor and copy constructor for initializing the objects of POINT type
5	Develop a program to overload unary prefix –-(Pre-decrement) and binary + operators using friend function.
6	Develop a C++ program to create a class STUDENT with data members USN, name and age. Using inheritance create a class UGSTUDENT having fields semester, fees and stipend. Enter data for at least 5 students and compute the semester wise average age for UG students.
7	Develop a vector class template for performing the scalar product of int type vectors as well as float type vectors.
8	Develop a C++ program using function template called bubbleSort( ) to sort the given array elements.
9	Develop a C++ program to define media class with suitable data members and member functions. Define Book class and tape class which derives the properties of media class. Use display( ) function to display the contents of the class. Create pointers to media class to access the functions of derived class.
10	Develop a program in C++ to illustrate the divide by zero exception handling.
11	Develop a program that has multiple catch statements to handle various types of exceptions.
12	Develop a C++ program to create a text file, check file created or not, if created it will write some text into the file and then read the text from the file.

COURSE ARTICULATION MATRIX (CO-PO AND CO-PSO MAPPING)

COs	POs												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	2	1												2	
CO2			2											2	
CO3			2											2	
CO4			2											2	
CO5			2											2	
Overall CO	**2**	**1**	**2**											**2**	

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

SOCIAL CONNECT & RESPONSIBILITY

Course Code	SHS01	CIE Marks	100
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	1	Exam Hours	-
Lecture Hour	-	Practical Hours	26 Hrs

Course objectives: This Course will enable students to:

- Enable the student to do a deep drive into societal challenges being addressed by NGO(s), social enterprises & The government and build solutions to alleviate these complex social problems through immersion, design & technology.
- Provide a formal platform for students to communicate and connect with their surroundings.
- Enable to create of a responsible connection with society.

Contents:

The course is mainly activity-based that will offer a set of activities for the student that enables them to connect with fellow human beings, nature, society, and the world at large. The course will engage students in interactive sessions, reading groups and semester-long activities conducted by faculty mentors. In the following a set of activities planned for the course have been listed:

Learning Outcomes: The students are expected to have the ability to:

1. Understand social responsibility
2. Practice sustainability and creativity
3. Showcase planning and organizational skills

UNIT-1 **(6Hrs)**

Plantation and adoption of a tree: Plantation of a tree by Miyawaki Method that will be adopted by entire semester by a group of students. They will also make an excerpt either as a documentary or a photoblog describing the plant's origin, its usage in daily life, and its appearance in folklore and literature

UNIT-2 **(6Hrs)**

Heritage walk and crafts corner: Heritage tour, knowing the history and culture of the city, connecting to people around through their history, knowing the city and its craftsman, photoblog and documentary on evolution and practice of various craft forms.

UNIT-3 **(4Hrs)**

Organic farming: Definition of organic farming, Organically grown crops in India, Differentiate between conventional farming and organic farming, Necessity of organic farming, Key characteristics of organic farming, Four principles of organic farming(principle of Health, principle of ecology, principle of fairness and principle of care),Types of organic farming: 1) Pure organic farming, 2) Integrated farming (Integrated nutrient management and Integrated pest management), objectives of organic farming, benefits of organic farming, Basic steps in organic farming and limitations of organic farming.

UNIT-4 **(6Hrs)**

Water Conservation: Global Water Scarcity - Global water crisis and its implications; Rainwater Harvesting - Concept and benefits of rainwater harvesting; Water Audit – An approach to water conservation; Efficient Water Use - Optimizing water consumption in daily life .

UNIT-5 **(4Hrs)**

Food Walk City's culinary practices, food lore, and indigenous materials of the region used in cooking.

Activities:
Plantation and adoption of a tree: Select suitable species in consultation with horticulture, forest or agriculture department. Interact with NGO/Industry and community to plant Tag the plant for continuous monitoring
Heritage walk and crafts corner: Survey in the form of questioner by connecting to the people and asking. Questions during survey can be asked in local language but report language is English.
Organic farming: Collect data on organic farming in the vicinity. Like types of crop, methodology etc.,
Water Conservation: Report on traditional water conservation practices (to minimize wastage)
Food Walk: Survey local food centres and identify its specialty, Identify and study the food ingredients, Report on the regional foods, Report on Medicinals values of the local food grains, and plants.

PEDAGOGY
The pedagogy will include interactive lectures, inspiring talks by various departments, field visits, social immersion. Applying and synthesizing information from these sources to define the social problem with your group. Social immersion with NGOs/social sections will be a key part of the course.

COURSE TOPICS:
The course will introduce social context and various players in the social space, and present approaches to discovering and understanding social needs. Social immersion and inspiring conversional will culminate in developing an actual, idea for problem-based intervention, based on an in-depth understanding of a key social problem.
A total of 26 hrs engagement per semester for this course in 3rd semester of the B.E. program. The students will be divided into 1 group of 60 each. Each group will be handled by one ffaculty mentor.

Guideline for Assessment Process:
Continuous Internal Evaluation (CIE)
Student shall keep a separate dairy and prepare report in consultation with the mentor/s to indicate what he has observed and learned in the social connect period.
Report shall be handwritten or blog with paintings, sketches, poster, video and/or photograph with Geo tag.
The report should be signed by the mentor.
The report shall be evaluated on the basis of the following criteria (see Table below) and/or other relevant criteria pertaining to the activity completed.
Each UNIT is evaluated for 35 Marks and final presentation will be for 15 marks.

Sl. No.	Particulars (for each UNIT)	Maximum Marks
1	Planning and scheduling the social connect	10
2	Information/Da ta collected during the social connect	10
3	Report writing	15
4	Final Presentation from the group	15
	Total	50

B.E COMPUTER SCIENCE & ENGINERRING Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – III			
UNIX & SHELL PROGRAMMING			
Course Code	S3CSA01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	100
Credits	1	Exam Hours	1.5
Lecture Hour	-	Practical Hours	26 Hrs

Course objectives: This Course will enable students to:

1. Understand the basic UNIX commands using bash shell.
2. Illustrate the advanced UNIX commands and their options to manipulate the file system.
3. Understand the basics of Shell program to write a shell script.
4. Exercise to build Software using Linux environment variables.

Experiments:

1.	How do you achieve the following using basic UNIX commands and vi editor: i) create a file, identify its attributes ii) edit the file contents using vi editor: insert-lines, words, copy-lines, words, delete-lines words, cut-lines words, append, search, navigating across the file iii) change the permissions of the files – both octal and symbolic notations iv) create a new user and change the ownership of the file v) record your login session vi) change any 3 terminal characteristics vii) create a directory structure, remove the current working directory and navigate across the file system – absolute and relative paths viii) create hard link and symbolic link for a file and identify the same in the file system ix) Identify no. of processes and explore any three options.
2.	Create a database file using space as a delimiter. How do you achieve the following? i) Display first 6 lines of the file ii) Display last 6 lines of the file iii) Display lines from 5 to 8 of the file iv) Display specified columns from a file v) Combine two files vertically vi) Sort the file based on field attributes vii) Search a given file viii) Count the number of characters, words and lines in a file ix) demonstrate to zip and unzip the files.
3.	Given a file, achieve the following operations: i) Redirect the file contents to both terminal and a new file ii) Enter a wrong command and redirect the error to a error file iii) Rectify the command and append the output to the same error file iv) Execute, cat file1.c nofile , Redirect the output of successful command to a file and error to error file v) Given two files, compare them using different filters

	vi) Redirect the output of a command to /dev/null . What is your observation? vii) Search a file based on a criteria viii) Identify suitable command for input redirection ix) Use system control command and run the job in background. x)Illustrate ps, sig, kill, system commands
4.	Create a text file, How do you achieve the following using GREP: i) Remove the blank lines from the file ii) List the 5 character palindromes iii) Select lines that have exactly 5 characters iv) Select the lines with leading or trailing zeros v) Number the above lines of text vi) Select lines that do not start with A to K. vii) List the dates available in mon/dd/yyyy viii) Select lines that contain floating point nos. ix) Select the lines that contain only one hex number x) Simulate wc -l, cat f1 f2.
5.	Create a text file, how do you achieve the following using sed : i. Replace all Read with Retrieve ii. Delete the blank line that follows the line that starts with an alphabet. iii. Double space the file iv. Extract the first word of each line v. Extract the year from the date in mm/dd/yyyy format vi. Print the line following a pattern match vii. Merge the odd numbered line and even numbered line. Eg. Merge 1 st and 2 nd line, 3 rd and 4 th line , …… viii. Delete any integer in each line. ix. Insert header info " Summary sheet" available in the file new.txt x. Simulate copy, head and tail
6.	Develop a Menu driven shell script that accepts two real numbers from the user to simulate a simple calculator. Display the result with suitable messages. Also, the program must take care of handling divide by zero error and the precision of the result must be 4. [Hint : To perform modulo operation, typecast the values].
7.	Develop a shell script that computes the Gross Salary and Net Salary of *'n'* employees according to the following: a) if basic salary is <1500 then HRA 10% of the basic, DA =90% of the basic and PF= 12% of the basic. b) if basic salary is > =1500 then HRA 500, DA =98% of the basic and PF=15% of the basic. The basic salary and no of employees 'n' must be entered interactively through the keyboard. The salary details(Sl. No, Employee name, Basic Salary, HRA, DA, PF, Gross Salary and Net Salary) must be displayed in tabular format with suitable message.
8.	Develop a shell script that accepts a list of filename as its arguments and perform the following : - counts and reports the occurrence of each word that is present in the first argument file on other argument files. - Checks every argument specified is a file or a directory and report accordingly. Whenever the argument is a file, the number of lines on it is also reported.
9.	Develop a shell scripts using functions to perform the following :

	- To check the given string is palindrome or not. Display the input string, reversed string and the result with suitable messages. - To find the substring in a given string. Input to the shell script must be accepted from the user and display the resultant string(s) along with input string(s) with suitable messages.
10.	Develop a shell script to check the permission of a file, print file line contents along with line numbers and copy the contents of files to another file.
11.	Develop a makefile to build executable. The build should be created using multiple .h and multiple .c source files

Course Outcomes:
Upon completion of this course the student will be able to:

CO1	To execute basic unix commands using bash shell
CO2	To program using shell scripting
CO3	To program in C to utilize unix services using filters, process and files

COURSE ARTICULATION MATRIX (CO-PO AND CO-PSO MAPPING)

Program Articulation Matrix															
Course Outcomes	**Program Outcomes**												**Program Specific Outcome**		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1			2											2	
CO2			2											2	
CO3			2											2	
over all			2											2	

Degree of compliance 1: Slight (Low) 2: Moderate (Medium) 3: Substantial (High)

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

MS OFFICE AND LATEX

Course Code	S3CSA02	CIE Marks	50
TeachingHours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hour	-	Practical Hours	26 Hrs

Course objectives: This Course will enable students to:

1. Understand the basics of MS Office to prepare documents and small presentations,
2. Illustrate the features of spreadsheet/worksheet with various options.
3. Create simple presentations using templates with various options available.
4. Demonstrate the ability to apply application software in an office environment.
5. Usage of Latex tool to prepare quality documents.

Sl. no.	Experiments
1	Create a document using MS-Word with at least three paragraphs and perform the following operations: • With left and right margin of 0.75” and Top and Bottom margin of 1”. • Insert page number in every page. • Centre the heading and make it bold, increase the font size. • Underline the specified words in the document and change them to italics. • Conduct the spell check and correct them suitably. • Exchange paragraphs 2 and 3 using cut and paste facility. • Put suitable headers and footers. • Find and replace the text. • To demonstrate Watermarking technique using ima • Make your documents protected by a password so that nobody changes it.
2	Create a document using MS-Word with at least two paragraphs and also create a power point presentation with minimum two slides. Include each slide a minimum of 4 to 5 lines of text. Then perform the following operations: • Count the number of words and lines in word document. • Using mail merge facility of MS-Word send a birthday invitation to 10 of your friends. • Insert page border and insert gutter of 1” in word document. • Create a time table of your class in the following format in MS-Word.

			Tea Break			Lunch Break			

<table>
<tr><td></td><td>

Create a hyperlink from the Power Point slide to the above time table created.
Create a hyperlink from word to the power point document.

</td></tr>
<tr><td>3</td><td>Following is the table of scores of basketball players using MS Excel.
<table>
<tr><td>Players</td><td>Team A</td><td>Team B</td></tr>
<tr><td>A</td><td>10</td><td>11</td></tr>
<tr><td>B</td><td>10</td><td>9</td></tr>
<tr><td>C</td><td>11</td><td>12</td></tr>
<tr><td>D</td><td>13</td><td>12</td></tr>
<tr><td>E</td><td>15</td><td>15</td></tr>
<tr><td>F</td><td>20</td><td>18</td></tr>
<tr><td>Average Score</td><td></td><td></td></tr>
<tr><td>Grade</td><td></td><td></td></tr>
</table>
Using the table, perform the following operations:

Calculate the average-score of each team and assign grade for the team - if the average score of Team A is greater than Team B, then assign Grade ‘A’ to Team A or assign Grade ‘A’ to Team B.
Draw a Bar Chart.
Demonstrate Auto Format.
Demonstrate relative reference and absolute reference formulation.

</td></tr>
<tr><td>4</td><td>Using MS Excel create a table of five students for their marks scored in five subjects (max. marks 25).
<table>
<tr><td>Players</td><td>Team A</td><td>Team B</td></tr>
<tr><td>A</td><td>10</td><td>11</td></tr>
<tr><td>B</td><td>10</td><td>9</td></tr>
<tr><td>C</td><td>11</td><td>12</td></tr>
<tr><td>D</td><td>13</td><td>12</td></tr>
<tr><td>E</td><td>15</td><td>15</td></tr>
<tr><td>F</td><td>20</td><td>18</td></tr>
<tr><td>Average Score</td><td></td><td></td></tr>
<tr><td>Grade</td><td></td><td></td></tr>
</table>
Using the table, perform the following operations for the above data:

Calculate the percentage of each student.
Highlight the data cell with green color if a student scores above 20 and highlight with red color, if a student scores less than 10 using formatting toolbar.
Draw a Column Bar chart.
Demonstrate the filter option.

</td></tr>
<tr><td>5</td><td>You are asked to make a presentation about your organization to the general public. Develop the presentation as specified below:

Make a power point presentation consisting of the following 3 slides.
Slide 1: Your organization Name and Mission. Insert a picture either from the clipart gallery or from the library of scanned images of your organization.
Slide 2: Minimum 5 lines of text describing different departments in your organization. Create a word document describing the departments and give the link to the word document from this slide.

</td></tr>
</table>

	• Slide 3: Create organization chart of SIT with suitable data. • Give transition effects to each slide with automatically advancing to next slide (5 Sec.). • Demonstrate custom animation and action buttons for the slides. • Show presentation.
6	Make a Power point presentation on any topic of your interest. Create Master Slide with suitable slide design, slide layout, background color, slide number & date. Use this Master slide to demonstrate the following operations: • Give transition effects to each slide with automatically advancing to next slide (5 Sec). • Demonstrate how to use appropriate fonts, pictures and colors. • Demonstrate how to add charts and hyperlinks. • Demonstrate custom animations & action buttons for the slides. • Demonstrate organization chart with suitable data. • Show presentation.
7	Create a database using MS-Access to keep track of products purchased which has the attributes as pid, pname, price, city, country. Demonstrate the following : • Creating the table. • Inserting suitable values into the table. • Creating the forms. • Querying the database. • Generating suitable reports.
8	Create the database of a member of the Library using MS Acess having the fields like member¬_id, member_name, account_open_date, max_books_allowed, penalty_amount. Demonstrate the following • Creating the table. • Inserting suitable values into the table. • Creating the forms. • Querying the database.
9	Create a document using LATEX typesetting system with the following contents : • title page with author name • table of contents • header and footer • section containing text in paragraphs having certain words in bold ,italics, colouring & foot notes • section containing formulae for the following - Roots of a quadratic equation - Euler's number definition - Taylors series
10	Create a document using LATEX typesetting system with the following contents : • section containing lists • numbered lists • unordered lists • section containing image with a title

	• section containing table to display marks of 5 students in 5 subjects

Course outcomes:

On successful completion of this course, stuents will be able to:

1. Understand the basics features of MS Office tools such as MS Word, Excel, Power Point & Access and Latex tools.
2. Apply the knowledge of MS Office to solve given problems.
3. Apply the knowledge of Latex tool to prepare quality documentation.

Conduct of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Breakup of marks and the instructions printed on the cover page of answer script to be strictly adhered by the examiners.
- Students can pick one experiment from the questions lot prepared by the examiners.
- Change of experiment is allowed only once and 20% Marks is to be deducted.

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – III

ETHICS & PUBLIC POLICY FOR AI

Course Code	S3CSA04	CIE Marks	100
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	-
Credits	1	Exam Hours	1.5
Lecture Hour	-	Practical Hours	26 Hrs

Course objectives: This Course will enable students to:

- To understand Ethical Framework for a Good AI Society, establishing Rules for trustworthy AI
- To Designing ethics for good society
- To familiar with Tools, methods and practices for designing AI for social good
- To familiar with Innovation and future AI
- To understand the Case Study: Ai in health care, knowing Regulation and Governance of AI ethics

Teaching-Learning Process (General Instructions)
These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes. 1. Chalk and Talk 2. Real time Examples 3. Natural Approaches

UNIT-1

An Ethical Framework for a Good AI Society: opportunities, Risks, principles and Recommendations. Establishing the rules for building trustworthy AI Textbook1: Chapter 3, chapter 4

UNIT-2

How to design AI for social good: seven essential factors From What to How: An Initial Review of publicly available AI Ethics tools, Methods and Research to Translate principles into Practices Textbook1: Chapter 9, Chapter 10

UNIT-3

How to design AI for social good: seven essential factors From What to How: An Initial Review of publicly available AI Ethics tools, Methods and Research to Translate principles into Practices Textbook1: Chapter 9, Chapter 10

UNIT-4

Innovating with Confidence: Embedding AI Governance and fairness in financial Services Risk management framework, What the near future of AI could be. Textbook1: Chapter 20, chapter 22

UNIT-5 **(4Hrs)**

Human-AI Relationship, AI and Workforce, Autonomous Machines and Moral Decisions, AI in HealthCare: balancing Progress and Ethics,
Regulation and Governance of AI Ethics Textbook2 : Chapter 5,Chapter 8, Chapter 9

Course outcome (Course Skill Set)
At the end of the course, the student will be able to :

1. Describe Ethical Framework for a Good AI Society, establishing Rules for trustworthy AI
2. Explain ethics for good society
3. Illustrate various Tools, methods and practices for designing AI for social good
4. Describe the Innovation and future AI
5. Illustrate Regulation and Governance of AI ethics in Healthcare domain.

Assessment Details (both CIE and SEE) The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Continuous Internal Evaluation:

- For the Assignment component of the CIE, there are 25 marks and for the Internal Assessment Test component, there are 25 marks.
- The first test will be administered after 40-50% of the syllabus has been covered, and the second test will be administered after 85-90% of the syllabus has been covered
- Any two assignment methods mentioned in the 22OB2.4, if an assignment is project-based then only one assignment for the course shall be planned. The teacher should not conduct two assignments at the end of the semester if two assignments are planned.
- For the course, CIE marks will be based on a scaled-down sum of two tests and other methods of assessment.

Internal Assessment Test question paper is designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.

Semester-End Examination: Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the course (duration 03 hours). 1. The question paper will have ten questions. Each question is set for 20 marks. 2. There will be 2 questions from each UNIT. Each of the two questions under a UNIT (with a maximum of 3 sub-questions), should have a mix of topics under that UNIT. 3. The students have to answer 5 full questions, selecting one full question from each UNIT. 4. Marks scored shall be proportionally reduced to 50 marks.

Suggested Learning Resources:

Books

1. "Ethics, governance and Policies in Artificial Intelligence", Author-Editor : Luciano Floridi, Springer, 1st Edition 2021, vol 144, Oxford Internet Institute, University of ixford, UK, ISSN 0921-8599, e-ISSN 2542- 8349 Philosophical Studies series, ISBN 978-3-030-81906-4 e-ISBN 978-3-030-81907-1, ://doi.orghttps/10.1007/978-3-030-81907-1, 2021.
2. "Ethics and AI: Navigating the Moral Landscape of Digital Age", Author: Aaron Aboagye,

Scheme of Teaching, Examination and Syllabus

B.E. COMPUTER SCIENCE & ENGINEERING

Batch: 2023-24

IV SEMESTER

(Effective from the academic year 2024-2025)

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

DESIGN AND ANALYSIS OF ALGORITHMS

Course Code	S4CCS01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3	Exam Hours	3
Lecture Hour	40Hrs	Practical Hours	-

Course objectives: This Course will enable students to:

1. An introduction to the design and analysis of algorithms. (Synthesis)
2. Expose students to prove the correctness and analyse the running time of the basic algorithms. (Analysis)
3. To compare the running time of sorting and searching algorithms. (Comprehension)
4. Create an awareness of applying the algorithms and design techniques to solve problems. (application)

UNIT-1 **(08hrs)**

Introduction: Notion of algorithm, Fundamentals of Algorithmic Problem Solving, Fundamentals of the Analysis of Algorithm Efficiency: Analysis frame work, Asymptotic Notations and Basic Efficiency Classes, Mathematical Analysis of Non-recursive and Recursive Algorithms. [Chapters: 1.1, 1.2, 2.1-2.4]

UNIT-2 **(07 hrs)**

Brute Force: Selection Sort, Brute-Force String Matching, Exhaustive Search: Travelling Salesman problem, Knapsack Problem, Assignment Problem.
Divide and Conquer: Mergesort, Quicksort, Binary Search. [Chapters: 3.1, 3.2, 3.4, 4.1- 4.3]

UNIT-3 **(08hrs)**

Decrease and Conquer: Insertion Sort, Depth First Search, Breadth First Search, Topological Sorting, Algorithms for Generating Combinatorial Objects.
Transform and Conquer: Presorting, Balanced Search Trees: AVL Tree, Heaps and Heapsort. [Chapters: 5.1 – 5.4, 6.1, 6.3 (only AVL Trees),6.4]
Self Study : Algorithms for Generating Combinatorial Objects.

UNIT-4 **(08hrs)**

Dynamic Programming: Computing a Binomial Coefficient, Warshall's and Floyd's Algorithms, The Knapsack Problem.
Greedy Technique: Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm. [Chapters: 8.1, 8.2, 8.4, 9.1-9.3]

UNIT-5 **(09hrs)**

Space and Time Tradeoffs: Sorting by Counting, Input Enhancement in String Matching: Horspool's Algorithm.
Limitations of Algorithm Power: P, NP and NP-Complete Problems.
Coping with the Limitations of Algorithm Power: Backtracking: N-Queens, Hamiltonian Circuit Problem, Subset-Sum Problem. **Branch and Bound:** Assignment Problem, Travelling Salesman Problem**.** [Chapters: 7.1,7.2, 11.3, 12.1, 12.2]
Self Study : Limitations of Algorithm Power: P, NP and NP-Complete Problems.

Course outcomes:

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

1. Discuss the fundamental principles of analysis and design of algorithms.
2. Apply design techniques such as Brute -Force, Divide-and-Conquer, Decrease-and-Conquer, Transform-and-Conquer, Greedy, Dynamic programming, space & time trade-off and Backtracking to solve a given problem.
3. Design/Outline algorithms classified under different design techniques.
4. Analyse the complexity of a given algorithm.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbook				
1	Introduction to The Design & Analysis of Algorithms.	Anany Levitin	Pearson Education	Ed2 2007. ISBN: 81-7808-984-X
Reference Book				
1	Fundamentals of Computer Algorithms.	Ellis Horowitz, SatrajSahni and Rajasekharan.	University Press Pvt. Ltd,	2nd Edition, 2009

Course Articulation Matrix (CO-PO and CO-PSO mapping)

Course Outcomes	PROGRAMME OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
CO1	2													2	
CO2			2											2	
CO3			2											2	
CO4		2												2	
Overall CO	2	2	2											2	

Program articulation matrix:

Course Outcomes	PROGRAMME OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
	2	2	2											2	

Degree of compliance **1**: Low **2**: Medium **3**: High

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

MICROCONTROLLER AND EMBEDDED SYSTEMS (I)

Course Code	S4CSI01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4.0	Exam Hours	3
Lecture Hour	40Hrs	Practical Hours	26 Hrs

Course objectives: This Course will enable students to:

1. Introduce the concept of architecture and programming of advanced embedded microcontrollers.
2. ARM family of microcontrollers that are widely used in design of real time sophisticated embedded systems like tablets, hand held devices, automation and industrial control systems.
3. It also covers writing Embedded C programming of LPC2148 for GPIO, ADC, DAC, UART, LCD, Timers.
4. It also explains the concepts of embedded system and its components.

UNIT-1 **(08L+4P)**

CHAPTER 1-EMBEDDED SYSTEM COMPONENTS: Embedded v/s General computing system, Classification of Embedded systems, Major applications and purpose of Embedded systems, Core of an Embedded System including all types of Processors, Controller and Memory.

CHAPTER 2-ARM EMBEDDED SYSTEMS: The RISC Design Philosophy, The ARM Design Philosophy, Embedded System Hardware, Embedded System Software.

Text book 2, Chapter1: 1.1, 1.2, 1.4, 1.5, 1.6. Chapter2: 2.1, 2.2.

Text book 1: Chapter1: 1.1- 1.4.

UNIT-2 **(8L+4P)**

CHAPTER 3-ARM PROCESSOR FUNDAMENTALS: Registers, Current Program Status Register, Pipeline, Exceptions, Interrupts, and Vector Table Core Extensions, Architecture Revisions, ARM Processor Families, LPC2148 Microcontroller Architecture, LPC2148 Microcontroller Architecture, Memory Mapping, Register Description.

Text book 1: Chapter 2: 2.1-2.7.

UNIT-3 **(8L + 6P)**

CHAPTER 4-INTRODUCTION TO THE ARM INSTRUCTIONS SET: Data Processing Instructions and examples, Branch Instructions and examples, Load-Store Instructions and examples, Software Interrupt Instructions and examples, Program Status Register Instruction and examples. Programs: Addition, Multiplication, division and Subtraction of 16, 32-bit data, Example Programs: Looping, conditional programs like sum of memory elements, Loading Constants, Conditional Execution, and Example Programs.

Text book1, Chapter3: 3.1-3.5.

UNIT-4 **(8L+6P)**

CHAPTER 5-INTERFACING Sensors, Actuators, GPIO, LED interfacing and programming in C,7 segment display interfacing and C program, stepper motor interfacing, Keyboard interfacing, Push button switch interfacing and programming in C.

Text book 2, Chapter2:2.3- 2.3.1, 2.3.2, 2.3.3-2.3.3.1, 2.3.3.2, 2.3.3.3, 2.3.3.4, 2.3.3.7, 2.3.3.8.

UNIT-5 **(8L+6P)**

INTERFACING Cont… Data Conversions (ADC, DAC) interfacing and C program, Timers, Counters, Communication Protocols: UART, I2C (on board) SPI, CAN Programs using C

Course outcomes:

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

1. Analyze the fundamental concept of Embedded System Architecture.
2. Understand and analyze the ARM processor architecture and its family.
3. Develop assembly language programs to perform specific tasks using ARM instructions.
4. Design and develop embedded C programs to interface external hardware with LPC214x microcontroller.
5. Design and develop the solutions for a problem using embedded system and demonstrate.

Question paper pattern:

- The question paper will have ten full questions carrying equal marks.
- Each full question will be for 20 marks.
- There will be two full questions (with a maximum of four sub- questions) from each UNIT.
- Each full question will have sub- question covering all the topics under a UNIT.
- The students will have to answer five full questions, selecting one full question from each UNIT.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	ARM Systems Developer's Guide Designing and Optimizing System Software. (Chapters 1, 2, 3)	Andrew N. Sloss, Dominic Symes, Chris wright, Morgan Kaufmann	Elsevier Inc,	2004
2	Introduction to Embedded Systems	Shibu K V	Tata McGraw Hill Education Private Limited	Second edition , 2017
Reference Books				
1	ARM System on Chip Architecture	Steve Furber	Pearson Education Limited,	Second Edition, 2000
2	ARM ASSEMBLY LANGUAGE Fundamentals and Techniques	William Hohl, Christopher	CRC Press	Second Edition, 2015
3	ARM Assembly Language an Introduction	Gibson	John Wiley & Sons	Second Edition, 2007

Integrated Lab Syllabus

Course objectives:

This Course will enable students to:

1. Develop and test assembly language programs (ALP) using ARM7TDMI/LPC2148
2. Conduct the experiments on ARM7TDMI/LPC2148 evaluation board using evaluation version of Keil µvision tool.

Sl. no.	Programs

PART A

Conduct the following experiments by developing Assembly Language Program (ALP) for LPC2148 using Keil µvision-5 tool.

1	Develop an ALP to add two 64-bit numbers.
2	Develop an ALP to find the sum of first *n* natural numbers.
3	Develop an ALP to find the factorial of a non-zero number using subroutine.
4	Develop an ALP to add an array of *n* numbers.
5	Develop an ALP to find the square of a number (0-10) using Look-up table.
6	Develop an ALP to find the largest number in an array of 32-bit signed integers.
7	Develop an ALP to arrange a series of 32-bit unsigned numbers in ascending order.
8	Develop an ALP to count the number of ones and zeroes in two consecutive memory locations.

PART B

Conduct the following experiments on LPC2148 evaluation board by developing embedded C program using Keil µvision-5 and Flash Magic tools.

1	Develop an embedded C program to display the Hex digits 0 to F on a Seven-Segment display interface with an appropriate delay in between.
2	Develop an embedded C program to drive a Stepper motor Interface to rotate the motor in clockwise direction.
3	Develop an embedded C program to read the status of 8 input bits from the Logic Controller Interface and display 'FF' if it is even parity bits otherwise display 00. Also display number of 1's in the input data.
4	Develop an embedded C program to display "Hello World" message on a 2x16 character LCD interface.
5	Develop an embedded C program to generate triangular and square waveforms using the DAC Interface. (The output of the DAC is to be displayed on the CRO).
6	Develop an embedded C program to display the message "dEPt OF CSE" from right to left and left to right on a Seven Segment display interface.
7	Develop an embedded C program to drive a Stepper motor Interface to rotate the motor in anti-clockwise direction.
8	Develop an embedded C program to generate Half-rectified Sine waveform using DAC interface.

Course outcomes:

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

1. Summarize ARM instruction sets and comprehend the knowledge of how assembly language works.
2. Design and develop ARM assembly language programs using Keil tool.
3. Infer functioning of hardware devices and interfacing them to ARM microcontroller.
4. Design and develop embedded C code for interfacing different UNITs with ARM microcontroller using Keil and Flash Magic tools.

COURSE ARTICULATION MATRIX (CO-PO AND CO-PSO MAPPING)

COs	POs												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	2												2		
CO2	1												2		
CO3	2	2	2										2		
CO4	2	2	2										2		
CO5	2	2	2										2		
Overall CO	**2**	**2**	**2**										**2**		

PROGRAM ARTICULATION MATRIX:

COs	POs												PSOs		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO	**2**	**2**	**2**										**2**		

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

THEORY OF COMPUTATIONS(I)

Course Code	S4CSI02	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	3
Lecture Hour	40Hrs	Practical Hours	26Hrs

Course objectives:
This Course will enable students to:
1. **Explain** the concepts of automata theory and formal languages. (Comprehension)
2. **Identify** different formal language classes like regular and context free and their relationships. (Knowledge)
3. **Design** grammars and recognizers for different formal languages. (Synthesis)
4. **Analyze** and modify the CFGs to normal forms. (Analysis)
5. **Demonstrate** the equivalence between various language recognizers. (Application)
6. **Describe** Turing machine and its variants. (Comprehension)

UNIT-1 **(8L hrs)**

Introduction to Finite Automata: Why study automata theory? The central concepts of Automata theory; Deterministic finite automata; Nondeterministic finite automata; Finite automata with Epsilon-transitions. (Text Book: 1.1.1, 1.5, 2.2, 2.3.1-2.3.5, 2.5)

UNIT-2 **(8L hrs)**

Regular expressions and Languages: Regular expressions, Finite Automata and Regular Expressions- Converting DFA's to Regular Expressions by Eliminating States, converting regular expressions to automata; **Properties of Regular languages**- Equivalence and Minimization of Automata - Testing equivalence of states, Testing equivalence of regular languages, Minimization of DFA's. (Text Book: 3.1, 3.2.2, 3.2.3,4.4.1-4.4.3)

UNIT-3 **(8L hrs)**

Context-Free Grammars and Languages: Context-free grammars; Parse trees- Constructing Parse Trees, The yield of a parse tree; Ambiguity in grammars and languages- Ambiguous grammars, Leftmost derivation as a way to express ambiguity, Inherent ambiguity. Normal forms for CFGs, Eliminating useless symbols, Computing reachable symbols, eliminating € productions and unit productions, Chomsky normal form.(Text Book:5.1, 5.2.1, 5.2.2,5.4.1,5.4.3,5.4.4,7.1)

UNIT-4 **(8L hrs)**

Pushdown Automata: Definition of the Pushdown automata; The languages of a PDA, Acceptance by Final State, Acceptance by Empty Stack, From Empty Stack to Final State, From Final State to Empty Stack, Equivalence of PDA's and CFG's: From Grammars to Pushdown Automata, From PDA's to Grammars, Deterministic Pushdown Automata- Definition of a DPDA. (Textbook:6.1, 6.2, 6.3, 6.4.1)

UNIT-5 **(8L hrs)**

Introduction to Turing Machine: The Turing Machine: Notation for the TM, Instantaneous Descriptions for the TM, Transition diagrams for the TM, The Language of a TM, TM and Halting, Multitape Turing machines, Restricted Turing machines: Turing machines with Semi-infinite tapes, Multistack machines . (Text Book:8.2.2 - 8.2.6, 8.4.1, 8.5.1,8.5,2)

Course outcomes:
After the completion of this course, students will be able to:

- **Describe** and **Design** the various forms of finite automata for a given regular language.
- **Describe** and **Design** an equivalent regular expression for a given regular language and also **Identify** the equivalence between the various representations of the finite automata.
- **Describe** and **Design** the context free grammars for a given context free language and also **Apply** Chomsky normal form to normalize them to a standard form.
- **Describe** and **Design** various types of push down automata for a given language and **Show** their equivalence.
- **Describe** and **Design** Turing machine for a given language and also **Explain** its variants like Multitape and Multistack turing machines.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Introduction to Automata Theory, Languages and Computation	John E. Hopcroft, Rajeev Motwani, Jeffrey D.Ullman	Pearson education	3rd Edition, 2007
Reference Books				
1	An Introduction to Formal Languages and Automata	Peter Linz	Narosa publication	4th edition
2	Introduction to Languages and Automata Theory	John C Martin	Tata McGraw-Hill	3rd Edition, , 2007

	LAB
	PART-A
	Design and Implementation of various automata for a given language using C- language coding. **20hrs**
	PART-B
	Simulation of various automata for a given language using JFLAP/any equivalent simulator. **6 Hrs**

Course Articulation matrix(CO-PO and CO-PSO mapping)

Course Outcomes	PROGRAM OUTCOMES												PROGRAM SPECIFIC OUTCOMES		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
CO1	2		3										2		
CO2	2		3										2		
CO3	2	2	3										2		
CO4	2	2	3										2		
CO5	2	1	3										2		
Overall CO	2	2	3										2		

Program articulation matrix:

Course Outcomes	PROGRAMME OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
	2	2	3										2		

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

Course Code	S4CCSL01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	3 Hours
Lecture Hour	-	Practical Hours	26Hrs

Course objectives: This Course will enable students to:

1. An introduction to the design and analysis of algorithms. (Synthesis)
2. Expose students to prove the correctness and analyse the running time of the basic algorithms. (Analysis)
3. To compare the running time of sorting and searching algorithms. (Comprehension)
4. Create an awareness of applying the algorithms and design techniques to solve problems. (application)

Sl. no.	Experiments
	Note: C/C++ language must be used to develop the following programs:
1	Sort a given set of elements using the Merge sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2	Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
3	Print all the nodes reachable from a given starting node in a given digraph using Depth First Search method.
4	Print all the nodes reachable from a given starting node in a digraph using Breadth First Search method.
5	Obtain the Topological ordering of vertices in a given digraph using source removal method.
6	Sort a given set of elements using the Heap sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
7	Implement Horspool algorithm for String Matching.
8	Implement 0/1 Knapsack problem using dynamic programming.
9	a. Implement All Pair Shortest paths problem using Floyd's algorithm. b. Compute the transitive closure of a given directed graph using Warshall's algorithm.
10	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.
11	Find Minimum Cost Spanning Tree of a given undirected graph using Prims algorithm.
12	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm
13	Find a subset of a given set S={sl,s2,.....,sn} of n positive integers whose sum is equal to a

	given positive integer d. For example, if S= {1, 2, 5, 6, 8} and d = 9 there are two solutions{1,2,6}and{1,8}.A suitable message is to be displayed if the given problem instance doesn't have a solution.
14	Implement N Queen's problem using Back Tracking.

Course outcomes:

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

1. **Design/Develop** a solution for the given problem using appropriate design techniques such as brute-force, greedy, dynamic programming, divide and conquer, decrease and conquer, transform and conquer and backtracking.
2. **Analyse** the efficiency of sorting algorithms with respect to time and space complexity.
3. **Apply** various algorithmic design techniques to solve real world problems.

Conduct of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Breakup of marks and the instructions printed on the cover page of answer script to be strictly adhered by the examiners.
- Students can pick one experiment from the questions lot prepared by the examiners.
- Change of experiment is allowed only once and 20% Marks is to be deducted.

Course Articulation Matrix (CO-PO and CO-PSO mapping)

Course Outcomes	PROGRAMME OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
CO1	2		2											2	
CO2	2	2	2											2	
CO3	2		2											2	
Overall CO	2	2	2											2	

Degree of compliance **1**: Low **2**: Medium **3**: High

B.E COMPUTER SCIENCE & ENGINEERING

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

SEMESTER - IV

DISCRETE MATHEMATICAL STRUCTURES

Course Code	S4CCS02	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3	Exam Hours	03
Lecture Hour	40Hrs	Practical Hours	-

Course objectives:

The course will enable students to

1. Illustrate the domain and range of a relation and their properties.
2. Explain the basics of groups and its associated concepts.
3. Demonstrate the theory of Boolean algebra and normalize a switching circuit.
4. Identify types of graphs, outline properties of graphs
5. Illustrate tree structure and its properties.

UNIT-I: Relations and Function 8 **Hours**

Relations, Properties of Relations, Computer Recognition- Zero-One Matrices and Digraphs, Partial order relation -Poset and Hasse-Diagrams, Equivalence Relation and Partitions, Extremal elements of a Poset, Lattice.

UNIT-II: Groups 8 **Hours**

Binary Operations and Properties, Definition of a Group, Examples and Elementary properties, Abelian Group, Homomorphism, Isomorphism and Cyclic Groups, Cosets and Lagrange's Thoerem, Normal subgroups.

UNIT-III Boolean Algebra and Switching Functions: **7 Hours**

Introduction, Definition of Boolean algebra and Boolean function, Laws of Boolean functions and problems Switching functions: Disjunctive and conjunctive normal forms. Structure of Boolean Algebra.

UNIT-IV Introduction to Graph Theory: 8 **Hours**

Definitions and Examples, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits, Planar Graphs, Hamilton Paths and Cycles.

UNIT-V Graph Coloring and Trees: 8 **Hours**

Graph Coloring, and Chromatic Polynomials. **Trees:** Definitions, Properties, and Examples, Rooted Trees, Trees and Sorting, Weighted Trees and Prefix Codes. Minimal spanning Tree, Transport Networks: Max-Flow Min-cut Theorem.

Course outcomes:

Upon completion of this course the student will be able to:

1. Compute zero-one matrix, composition of relations and draw Hasse diagram. (L3).
2. Apply the concept of groups and subgroup to verify Lagrange's theorem.. (L2).
3. Apply the theory of Boolean algebra to minimize switching functions. (L3).
4. Recognize types of graphs, outline properties of graphs**,** understand isomorphism and apply Graph theory tools in solving real world problems. (L2/ L3).
5. Colour the vertices/ edges of a graph, understand tree structure, its properties, importance of minimal spanning tree and hence the shortest path using algorithms. (L2/L3).

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Discrete and Combinatorial Mathematics	Ralph P. Grimaldi	Pearson Education	5th edition 2012
2	Discrete Mathematical Structures	Bernard Kolman, Robert Busby and Sharon C. Ross	Pearson Education	6th edition 2012
3	Elementary Number Theory	David M Burton	McGraw Hill	7th Edition 2013
Reference Books				
1	Discrete Mathematical and its Applications	Kenneth H. Rosen	Tata-McGraw Hill	7th Edition, 2011
2	Discrete Mathematical Structures with Applications to computer science	J.P.Tremblay and R. Manohar	Tata-McGraw Hill	2010
3	Problems Algebraic number theory	M. Ram Murthy and Jody Esmonde	Springer	2006
4	Advanced Engineering Mathematics	Erwin Kreyszig	Wiley Publications,	10th edition 2015

Mapping of Course outcomes to Program outcomes

		PROGRAM OUTCOMES											
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
COs	CO1	**3**											
	CO2	**3**											
	CO3	**3**	1										
	CO4	**3**											
	CO5	**3**											

1: Low association, 2: Moderate association, 3: High association

B.E COMPUTER SCIENCE & ENGINERRING Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – IV			
GRAPH THEORY			
Course Code	S4CCS03	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3	Exam Hours	3
Lecture Hour	40Hrs	Practical Hours	-

UNIT-1	**(08Hrs)**
Introduction to Graph Theory: Definitions and Examples, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits, Planar Graphs, Hamilton Paths and Cycles	
UNIT-2	**(08Hrs)**
Introduction to Graph Theory contd.: Graph Colouring, and Chromatic Polynomials. **Trees:** Definitions, Properties, and Examples, Rooted Trees, Trees and Sorting, Weighted Trees and Prefix Codes. Minimal spanning Tree, Transport Networks: Max-Flow Min-cut Theorem, Matching	
UNIT-3	**(08Hrs)**
Fundamental Principles of Counting: The Rules of Sum and Product, Permutations(linear, circular, identical objects), Combinations – The Binomial Theorem, Combinations with Repetition, The Catalon Numbers. **The Principle of Inclusion and Exclusion**: The Principle of Inclusion and Exclusion, Generalizations of the Principle.	
UNIT-4	**(08Hrs)**
Derangements – Nothing is in its Right Place, Rook Polynomials. **Generating Functions**: Introductory Examples, Definition and Examples – Calculational Techniques, Partitions of Integers. The Exponential Generating Function, The Summation Operator.	
UNIT-5	**(08Hrs)**
Recurrence Relations: First Order Linear Recurrence Relation, The Second Order Linear Homogeneous, Recurrence Relation with Constant Coefficients, The Non-homogeneous Recurrence Relation, The Method of Generating Functions.	

Course outcomes:

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

1. Identify types of graphs, outline properties of graphs, describe when the graphs are said to be same even though the shapes are different (isomorphism) and apply to some practical problems like seven bridge problem, traveling sales man problem.
2. Describe how to color the vertices/ edges of a graph, apply graph coloring in map coloring, describe what is a tree and its properties and apply the concept of trees in constructing optimal prefix codes. Determine the shortest path between two vertices, write algorithms for finding minimal spanning trees and apply the concepts in transport network.
3. Apply the techniques of counting to identify the number of ways in which a given task can be accomplished without list all the possibilities explicitly.
4. Identify the different physical situations in which principle of inclusion and exclusion can be used for counting.
5. Derive the generating function for the given situation and evaluate the required coefficient. Solve the recurrence relation and interpret the solution.

Question paper pattern:

- The question paper will have ten full questions carrying equal marks.
- Each full question will be for 20 marks.
- There will be two full questions (with a maximum of four sub- questions) from each UNIT.
- Each full question will have sub- question covering all the topics under a UNIT.
- The students will have to answer five full questions, selecting one full question from each UNIT.

Sl. No	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbook				
1	Discrete and Combinatorial Mathematics	Ralph P. Grimaldi	Pearson Education,	5th edition 2012
2	Applied Combinatorics	Alan Tucker	Wiley-India	5th edition 2011
Reference Book				
1	Graph Theory and Combinatorics	Dr.D.S.Chandrasekharaiah	Prism	2005
2	Introductory Combinatorics	Richard A. Brualdi	Pearson Prentice Hall	5th edition 2014
3	Graph Theory Modeling, Applications, and Algorithms	Geir Agnarsson & Raymond Geenlaw	Pearson Prentice Hall	2008

Mapping of Course outcomes to Program outcomes

		PROGRAM OUTCOMES											
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
COs	CO1	**3**											
	CO2	**3**											
	CO3	**3**	1										
	CO4	**3**											
	CO5	**3**											

1: Low association, 2: Moderate association, 3: High association

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

LINEAR ALGEBRA

Course Code	**S4CCS04**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3	Exam Hours	3
Lecture Hour	40Hrs	Practical Hours	-

UNIT-1 **(08Hrs)**

Linear equations: Systems of linear equations, row reduction and Echelon form, vector equations, Matrix equation, solution sets of linear systems, **Applications of Linear system**.

UNIT-2 **(08Hrs)**

Matrix Algebra: Introduction to linear transformations, Matrix of a linear transformation. Matrix operations, Inverse of a matrix, characterization of invertible matrices, partitioned matrices, matrix factorizations

UNIT-3 **(08Hrs)**

Eigen values, Eigen vectors: Introduction, characteristic equation, **Complex Eigen values and Eigen vectors** diagonalization, Eigen vectors and linear transformations

UNIT-4 **(08Hrs)**

Orthogonality and least squares: Inner product, length, and orthogonality, orthogonal sets, orthogonal projections Gram-Schmidt process, Q-R factorization, least squares problems

UNIT-5 **(08Hrs)**

Symmetric Matrices and Quadratic Forms: Diagonalization of symmetric matrices, quadratic forms, Constrained optimization, the singular Value Decomposition

Course outcomes:

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

1. Apply the numerical methods to solve Systems of linear equations, row reduction and Echelon form, vector equations, Matrix equation, solution sets of linear systems, Linear independence (L3).
2. Solve the linear transformations, Matrix of a linear transformation. Matrix operations, Inverse of a matrix, characterization of invertible matrices, partitioned matrices, matrix factorizations, Determinants: Introduction, Properties, volume and linear transformations(L3).
3. Determine and describe characteristic equation, diagonalization, Eigen vectors and linear transformations, Complex Eigen values. Orthogonality- Inner product, length, and orthogonality, orthogonal sets, orthogonal projections (L1, L3).
4. Determine and Describe Gram-Schmidt process, least squares problems, Inner product spaces.
5. Diagonalization of symmetric matrices, quadratic forms and Constrained optimization, the singular Value Decomposition (L1, L3).

Question paper pattern:

- The question paper will have ten full questions carrying equal marks.
- Each full question will be for 20 marks.
- There will be two full questions (with a maximum of four sub- questions) from each UNIT.
- Each full question will have sub- question covering all the topics under a UNIT.
- The students will have to answer five full questions, selecting one full question from each UNIT.

Sl. No	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbook				
1	Linear algebra and its applications	David C. Lay	Pearson Education,	5th edition 2014
Reference Book				
1	Linear algebra and its applications	Gilbert Strang	Thomson Asia Pvt. ltd	4th edition 2007
2	Linear algebra	Kenneth Hoffman, Ray Kunze	Prentice-Hall of India Pvt. Ltd	2nd edition 2002

Mapping of Course outcomes to Program outcomes

	PROGRAM OUTCOMES												
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
COs	CO1	**3**											
	CO2	**3**											
	CO3	**3**	1										
	CO4	**3**											
	CO5	**3**											

1: Low association, 2: Moderate association, 3: High association

B.E COMPUTER SCIENCE & ENGINERRING
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER – IV

NUMERICAL TECHNIQUES

Course Code	**S4CCS05**	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3	Exam Hours	3
Lecture Hour	40Hrs	Practical Hours	-

Course objectives: This Course will enable students to:

1. To develop an ability to use algorithms for approximation problems w.r.t Differentiation and Integration
2. To develop an ability to use algorithms for approximation problems w.r.t Differential equations
3. To develop an ability to use algorithms for approximation problems w.r.t partial differential equations
4. To develop an ability to use algorithms for approximation problems w.r.t Linear Algebraic equations
5. To develop an ability to use algorithms for approximation problems w.r.t Finite element methods

UNIT-1 **(08L)**

Numerical Differentiation and Integration Introduction, Numerical Differentiation, Numerical Integration, Euler-Maclaurin Formula, Adaptive Quadrature Methods, Gaussian Integration, Singular Integrals, Fourier Integrals, Numerical Double Integration

UNIT-2 **(07L)**

Numerical Solution of Ordinary Differential Equations Introduction, Solution by Taylor's Picard's Method, Euler's Method, Runge-Kutta Methods, Predictor-Corrector Methods, the Cubic Spline Method, Simultaneous and Higher Order Equations, Boundary Value Problems: Finite-Difference Method, The Shooting Method

UNIT-3 **(08L)**

Numerical Solution of Partial Differential Equations Introduction, Finite-Difference Approximations, Laplace's Equation: Jacobi's Method, Gauss-Seidel Method, SOR Method, ADI Method, Parabolic Equations, Iterative Methods, Hyperbolic Equations. .

UNIT-4 **(08L)**

System of Linear Algebraic Equations Introduction, Solution of Centro-symmetric Equations, Direct Methods, LU- Decomposition Methods, Iterative Methods, III-conditioned Linear Systems.

UNIT-5 **(08L)**

The Finite Element Method: Functionals- Base Function Methods of Approximation- The Rayleigh – Ritz Method –The Galerkin Method, Application to two dimensional problems Finite element Method for one and two dimensional problems.

Course outcomes:

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

1. Assess the approximation techniques to formulate and apply appropriate strategy to solve real world problems.
2. Evaluate the accuracy of numerical methods for Differentiation and Integration
3. Evaluate the accuracy of numerical methods for Differential equation
4. Evaluate the accuracy of numerical methods for Partial Differential equation
5. Evaluate the accuracy of numerical methods for Linear Algebraic Equations
6. Evaluate the accuracy of numerical methods for Finite Element Method

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbook				
1	Numerical Methods	Balagurusamy, E.,	Tata McGraw –Hill	978-0074633113 Standard Edition, July 2017
Reference Book				
1	Numerical Analysis and Algorithms	Niyogi, Pradip	Tata McGraw –Hill	978-0070494930 2003

Mapping of Course outcomes to Program outcomes

Course Outcomes	PROGRAM OUTCOMES											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	2				2							
CO2	2				2							
CO3	2				2							
CO4	2				2							
CO5	2				2							
CO6	2				2							
Overall CO	2				2							

B.E COMPUTER SCIENCE & ENGINEERING
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER - IV

BIOLOGY FOR ENGINEERS

Course Code	S4CCA01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3	Exam Hours	03
Lecture Hour	40Hrs	Practical Hours	-

Course objectives: This Course will enable students to:

1. To familiarize the students with the basic biological concepts and their engineering applications.
2. To enable the students with an understanding of biodesign principles to create novel devices and structures.
3. To provide the students an appreciation of how biological systems can be re-designed as substitute products for natural systems.
4. To motivate the students to develop interdisciplinary vision of biological engineering..

Teaching-Learning Process (General Instructions)

These are sample Strategies, which teachers can use to accelerate the attainment of the various course outcomes.

- Explanation via real life problem, situation modelling, and deliberation of solutions, hands-on sessions, reflective and questioning /inquiry-based teaching.
- Instructions with interactions in classroom lectures (physical/hybrid).
- Use of ICT tools, including YouTube videos, related MOOCs, AR/VR/MR tools.
- Flipped classroom sessions (~10% of the classes).
- Industrial visits, Guests talks and competitions for learning beyond the syllabus.
- Students' participation through audio-video based content creation for the syllabus (as assignments).
- Use of gamification tools (in both physical/hybrid classes) for creative learning outcomes.
- Students' seminars (in solo or group) /oral presentations.

UNIT-1 **(08 hrs)**

INTRODUCTION TO BIOLOGY:

The cell: the basic unit of life, Structure and functions of a cell. The Plant Cell and animal cell, Prokaryotic and Eukaryotic cell, Stem cells and their application. Biomolecules: Properties and functions of Carbohydrates, Nucleic acids, proteins, lipids. Importance of special biomolecules; Enzymes (Classification (with one example each), Properties and functions), vitamins and hormones.

UNIT-2 **(08 hrs)**

BIOMOLECULES AND THEIR APPLICATIONS (QUALITATIVE):

Carbohydrates (cellulose-based water filters, PHA and PLA as bioplastics), Nucleic acids (DNA Vaccine for Rabies and RNA vaccines for Covid19, Forensics – DNA fingerprinting), Proteins (Proteins as food – whey protein and meat analogs, Plant based proteins), lipids (biodiesel, cleaning agents/ detergents), Enzymes (glucose-oxidase in biosensors, lignolytic enzyme in bio-bleaching).

UNIT-3 **(08 hrs)**

HUMAN ORGAN SYSTEMS AND BIO DESIGNS (QUALITATIVE):

Brain as a CPU system (architecture, CNS and Peripheral Nervous System, signal transmission, EEG, Robotic arms for prosthetics. Engineering solutions for Parkinson's disease). Eye as a Camera system (architecture of rod and cone cells, optical corrections, cataract, lens materials,

bionic eye). Heart as a pump system (architecture, electrical signalling - ECG monitoring and heart related issues, reasons for blockages of blood vessels, design of stents, pace makers, defibrillators). Lungs as purification system (architecture, gas exchange mechanisms, spirometry, abnormal lung physiology - COPD, Ventilators, Heart-lung machine). Kidney as a filtration system (architecture, mechanism of filtration, CKD, dialysis systems).

UNIT-4 **(08 hrs)**

NATURE-BIOINSPIRED MATERIALS AND MECHANISMS (QUALITATIVE):
Echolocation (ultrasonography, sonars), Photosynthesis (photovoltaic cells, bionic leaf). Bird flying (GPS and aircrafts), Lotus leaf effect (Super hydrophobic and self-cleaning surfaces), Plant burrs (Velcro), Shark skin (Friction reducing swim suits), Kingfisher beak (Bullet train). Human Blood substitutes-hemoglobin-based oxygen carriers (HBOCs) and perflourocarbons (PFCs).

UNIT-5 **(08 hrs)**

TRENDS IN BIOENGINEERING (QUALITATIVE):
Muscular and Skeletal Systems as scaffolds (architecture, mechanisms, bioengineering solutions for muscular dystrophy and osteoporosis), scaffolds and tissue engineering, Bioprinting techniques and materials, 3D printing of ear, bone and skin. 3D printed foods. Electrical tongue and electrical nose in food science, DNA origami and Biocomputing, Bioimaging and Artificial Intelligence for disease diagnosis. Self- healing Bioconcrete (based on bacillus spores, calcium lactate nutrients and biomineralization processes) and Bioremediation and Biomining via microbial surface adsorption (removal of heavy metals like Lead, Cadmium, Mercury, Arsenic).

Course outcomes:
On successful completion of this course, students will be able to:

1. Elucidate the basic biological concepts via relevant industrial applications and case studies.
2. Evaluate the principles of design and development, for exploring novel bioengineering projects.
3. Corroborate the concepts of biomimetics for specific requirements.
4. Think critically towards exploring innovative biobased solutions for socially relevant problems.

Assessment Details (both CIE and SEE)
The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Continuous Internal Evaluation:

- For the Assignment component of the CIE, there are 25 marks and for the Internal Assessment Test component, there are 25 marks.
- The first test will be administered after 40-50% of the syllabus has been covered, and the second test will be administered after 85-90% of the syllabus has been covered
- Any two assignment methods mentioned in the 22OB2.4, if an assignment is project-based then only one assignment for the course shall be planned. The teacher should not conduct two assignments at the end of the semester if two assignments are planned.
- For the course, CIE marks will be based on a scaled-down sum of two tests and other methods of assessment.
- Internal Assessment Test question paper is designed to attain the different levels of Bloom's taxonomy as per the outcome defined for the course.

Semester-End Examination:

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the course (duration 03 hours).

1. The question paper will have ten questions. Each question is set for 20 marks.
2. There will be 2 questions from each UNIT. Each of the two questions under a UNIT (with a maximum of 3 sub-questions), should have a mix of topics under that UNIT.
3. The students have to answer 5 full questions, selecting one full question from each UNIT.
4. Marks scored shall be proportionally reduced to 50 marks

Suggested Learning Resources:

Books

- Biology for Engineers, Rajendra Singh C and Rathnakar Rao N, Rajendra Singh C and Rathnakar Rao N Publishing, Bengaluru, 2023.
- Human Physiology, Stuart Fox, Krista Rompolski, McGraw-Hill eBook. 16th Edition, 2022
- Biology for Engineers, Thyagarajan S., Selvamurugan N., Rajesh M.P., Nazeer R.A., Thilagaraj W., Barathi S., and Jaganthan M.K., Tata McGraw-Hill, New Delhi, 2012.
- Biology for Engineers, Arthur T. Johnson, CRC Press, Taylor and Francis, 2011
- Biomedical Instrumentation, Leslie Cromwell, Prentice Hall 2011.
- Biology for Engineers, Sohini Singh and Tanu Allen, Vayu Education of India, New Delhi, 2014.
- Biomimetics: Nature-Based Innovation, Yoseph Bar-Cohen, 1st edition, 2012, CRC Press.
- Bio-Inspired Artificial Intelligence: Theories, Methods and Technologies, D. Floreano and C. Mattiussi, MIT Press, 2008.
- Bioremediation of heavy metals: bacterial participation, by C R Sunilkumar, N Geetha A C Udayashankar Lambert Academic Publishing, 2019.
- 3D Bioprinting: Fundamentals, Principles and Applications by Ibrahim Ozbolat, Academic Press, 2016.
- Electronic Noses and Tongues in Food Science, Maria Rodriguez Mende, Academic Press, 2016

Web links and Video Lectures (e-Resources):

- https://nptel.ac.in/courses/121106008
- https://freevideolectures.com/course/4877/nptel-biology-engineers-other-non-biologists
- https://ocw.mit.edu/courses/20-020-introduction-to-biological-engineering-design-spring-2009
- https://ocw.mit.edu/courses/20-010j-introduction-to-bioengineering-be-010j-spring-2006
- https://www.coursera.org/courses?query=biology
- https://onlinecourses.nptel.ac.in/noc19_ge31/preview
- https://www.classcentral.com/subject/biology
- https://www.futurelearn.com/courses/biology-basic-concepts

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Group Discussion of Case studies.
- Model Making and seminar/poster presentations.
- Design of novel device/equipment like Cellulose-based water filters, Filtration system.

CORRELATION BETWEEN COURSE OUTCOMES WITH PROGRAM OUTCOMES

Program articulation matrix

Course	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
BE	2	2	3			2	2								3

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes (PSOs)

		POs												PSOs		
		1	2	3	4	5	6	7	8	9	10	11	12	1	2	3
COs	CO1	2	2													3
	CO2	2	2	3												3
	CO3	2	2	3												3
	CO4	2	2													3
	CO5	2	2				2	2								3

1: Low, 2: Medium, 3: High

B.E COMPUTER SCIENCE & ENGINEERING
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER - IV

UNIVERSAL HUMAN VALUES

Course Code	SHS02	CIE Marks	50
Teaching Hours/Week (L:T:P)	(1:0:0)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hour	-	Practical Hours	-

Pre-requisites: Universal Human Values (conducted during induction programme)

Course objectives: This Course will enable students to:

1. Understanding of self-exploration about themselves (human beings), family, society and nature/existence.
2. Appreciating the harmony in the human being, family, society and nature/existence
3. Strengthening holistic perception of co-existence and mutual fulfilment among the four orders of nature.

UNIT-1 **(03 hrs)**

Understanding Harmony in the Human Being - Harmony in self

Understanding human being as a co-existence of the sentient 'I' and the material 'Body'; Understanding the needs of Self ('I') and 'Body' - happiness and physical facility; Understanding the Body as an instrument of 'I' (I being the doer, seer and enjoyer); Understanding the characteristics and activities of 'I' and harmony in 'I'.

UNIT-2 **(02 hrs)**

Understanding Harmony in self and body

Understanding the harmony of 'I' with the Body: Sanyam and Health, correct appraisal of Physical needs, meaning of Prosperity in detail, Include discussions to differentiate between i) Prosperity and accumulation. ii) Ensuring health vs dealing with disease.

UNIT-3 **(03 hrs)**

Understanding Harmony in the Family - Harmony in Human-Human Relationship

Understanding values in human - human relationship, meaning of Justice (nine universal values in relationships) and program for its fulfilment to ensure mutual happiness, Trust and Respect as the foundational values of relationship; Understanding the meaning of Trust, Difference between intention and competence; Understanding the meaning of Respect, Difference between respect and differentiation; the other salient values in relationship.

UNIT-4 **(02 hrs)**

Understanding Harmony in Society and Nature

Understanding the harmony in the society (society being an extension of family)- Resolution, Prosperity, fearlessness (trust) and co-existence as comprehensive Human Goals. Understanding the harmony in the Nature; Interconnectedness and mutual fulfilment among the four orders of nature-recyclability and self-regulation in nature.

UNIT-5 **(03 hrs)**

Understanding Harmony in all levels of Existence

Understanding Existence as Co-existence of mutually interacting units in all-pervasive space; Holistic perception of harmony at all levels of existence. Include discussions on-human being as cause of imbalance in nature (film "Home" can be used), pollution, depletion of resources and role of technology etc.

Course Outcomes:

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

1. Become more aware of themselves, and their surroundings (family, society, nature)
2. Become more responsible in life, and value human relationships and human society

3. Have better critical ability in handling problems and in finding sustainable solutions

Text Book:

1. Gaur, R.R. & Sangal R – 'Foundation Course in Human Values and Professional Ethics; Presenting a universal approach to value education through self-exploration', Excel Books, Bangalore, 2016, ISBN: 978-8-174-46781-2

Reference Book:

2. Tripathi A.N. – 'Human Values', New Age International Publisher, 2003, ISBN: 81-224-1426-5

Web Resource:

1. Story of Stuff, http://www.storyofstuff.com
2. https://www.youtube.com/channel/UCQxWr5QB_eZUnwxSwxXEkQw
3. https://fdp-si.aicte-india.org/8dayUHV_download.php
4. https://www.youtube.com/watch?v=8ovkLRYXIjE
5. https://www.youtube.com/watch?v=OgdNx0X923I

Mapping of COs with POs:

POs / COs	1	2	3	4	5	6	7	8	9	10	11	12
CO1						1		3				
CO2						1		3				
CO3						1		3				
Overall level						**1**		**3**				

Evaluation Pattern:

Two Tests of 25 marks each and 45 minutes duration

SEE for 50 marks and examination duration is 90 minutes

Description	Schedule	Duration (min)	Conducted for	Reduced to
Test-1	7th Week	45	25 marks	25 marks
Test-2	14th Week	45	25 marks	25 marks
CIE			50 marks	50 marks
SEE		90	50 marks	50 marks
Total			**100 marks**	**100 marks**

B.E COMPUTER SCIENCE & ENGINERRING

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

SEMESTER – IV

JAVA SCRIPT

Course Code	S4CSA01	CIE Marks	50 Marks
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50 Marks
Credits	01	Exam Hours	1.5
Lecture Hour	-	Practical Hours	26 Hrs

Course objectives: This Course will enable students to:

1. Implement java script programs to understand core concepts and object-oriented programming.
2. Implement standalone programs using Functions, and rollovers.
3. Develop programs on Java script for building smarter forms.
4. Develop programs on Java script for handling events.

Sl. No.	Experiments
1	Design and develop Real-time applications in JavaScript using • Control structures • Error handling
2	Design and develop Real-time applications in JavaScript using • Functions • Recursion
3	Design and develop Real-time applications in JavaScript using • Arrays • Iterables
4	Design and develop Real-time applications in JavaScript using • Class and Objects • Object Oriented Programming
5	Design and develop Real-time applications in JavaScript using • Constructors • Strings
6	Design and develop Real-time applications for Handling Events. • Event handling • Event delegation
7	Design and develop Real-time applications for Handling Events. • Forms • Dynamic web page creation

Course outcomes:

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

1. Design and develop applications of Object-oriented programming concepts using Core Java script.
2. Design and develop applications using Java script rollovers and functions.
3. Design and develop applications for interacting with users.
4. Design and develop smarter forms through efficient event handling mechanism.

Course Articulation matrix (CO-PO and CO-PSO mapping)

Course Outcomes	PROGRAM OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
CO1	1	1												2	
CO2	1	2												2	
CO3		2	2											2	
CO4		2	2											2	

B.E COMPUTER SCIENCE & ENGINEERING

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

SEMESTER – IV

ADVANCED DATA STRUCTURES

Course Code	S4CSA02	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hour	-	Practical Hours	26 Hrs

Course objectives:

This Course will enable students to:

1. Understand the abstract data types such as List and Vector.
2. Understand the variety of ways that linearly ordered data can be stored, accessed, and manipulated using Hash tables.
3. Explore various sorting algorithms.
4. Understand concepts of Heaps and Binomial Queues and Tree data structures.

Sl. no.	Experiments
1	Develop a program to implement Stack of strings using vectors from STL.
2	Develop a program to implement an Ordinary Queue and Circular Queue using Lists in STL. (Any one to be asked in Examination)
3	Develop a program to store and add two polynomials using lists in STL.
4	Develop a program to merge two unordered Vectors/Lists in STL into a ordered(sorted) Vector/List without using built in functions for sorting.
5	Develop a program to implement Separate Chaining hash table along with rehashing to store a set of integers for a table of prime size.
6	Develop a program to implement Linear Probing and Quadratic Probing along with rehashing to store a set of integers for a table of prime size. (Any one to be asked in Examination)
7	Develop a program to implement Double hashing hash table along with rehashing to store a set of integers for a table of prime size.
8	Develop a program to implement a Binary Heap as a maximum Priority Queue to store a set of strings.
9	Given a Priority Queue of size N, Identify the Binomial Trees present in the forest when implemented as minimum priority queue
10	Develop a program to implement Shell Sort to sort a set of given strings in descending order.
11	Develop a program to implement Quick Sort using median of three as a pivot to sort a set of integers in ascending order.
12	Develop a program to implement a B-Tree of order M = 3 to perform the following operations on set of elements {10, 20, 5, 6, 12, 30, 7}. 1. Insert keys to B – Trees. 2. Display the B – Trees keys.

Note: All the programs are to be implemented using C/C++

Course Outcome:

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

1. Apply various data structures for different applications in computer science.
2. Identify how data structures are implemented using C++ standard template library.
3. Identify and design algorithms and techniques of implementing data structures using STL
4. Analyze the working and applications of various data structures.

Conduct of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Breakup of marks and the instructions printed on the cover page of answer script to be strictly adhered by the examiners.
- Students can pick one experiment from the questions lot prepared by the examiners.
- Change of experiment is allowed only once and 20% Marks is to be deducted.

Course Articulation matrix (CO-PO and CO-PSO mapping)

Course Outcomes	PROGRAM OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
CO1	1	1												2	
CO2	1	2												2	
CO3		2	2											2	
CO4		2	2											2	

Program articulation matrix:

Course Outcomes	PROGRAMME OUTCOMES												PSO		
	1	2	3	4	5	6	7	8	9	10	11	12	PSO1	PSO2	PSO3
	2	2	2										2		

Degree of compliance **1**: Low **2**: Medium **3**: High

B.E. COMPUTER SCIENCE & ENGINEERING

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

SEMESTER – IV

CUDA PROGRAMMING

Course Code	S4CSA03	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hour	-	Practical Hours	26Hrs

Course objectives: This Course will enable students to:

1. To program using Message Passing Paradigm
2. To program using shared address space
3. To program for GPUs using CUDA

Sl. no.	Experiments
	Message Passing Interface
1	Establish communication between nodes.
2	Receive selective messages.
4	Factorial of a huge number.
5	Sorting
6	Vector operation
7	Matrix operation
	OpenMP
1	One dimensional array
2	Two dimensional array
3	Synchronization among threads
4	Scheduling of threads
5	Workload sharing
	CUDA
1	Basic image processing operation
2	Text analysis
3	One dimensional array
4	Two dimensional array
5	Query device properties and handling errors

Course outcomes:

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

1. To implement and debug program using Message Passing Interface (MPI)
2. To implement and debug program using OpenMP to use shared address space
3. To implement and debug programs on GPU

Conduct of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Breakup of marks and the instructions printed on the cover page of answer script to be strictly adhered by the examiners.
- Students can pick one experiment from the questions lot prepared by the examiners.
- Change of experiment is allowed only once and 20% Marks is to be deducted.

B.E COMPUTER SCIENCE & ENGINERRING

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

SEMESTER – IV

R PROGRAMMING

Course Code	S4CSA04	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hour	-	Practical Hours	26Hrs

Course objectives: This Course will enable students to:

1. Learn the basic programming concepts of R
2. Develop coding strategies for Data handling, sampling, pre processing and boot strapping using R
3. Use the logics with object oriented programming concepts to compute the program using R
4. Perform data analysis

Sl. no.	Experiments
1	Installations and Environment Set up, Introduction to R basics,R usage & applications (usecases)
2	R as calculator, Data Vectors, Built in Commands
3	Operations with Matrices, Data Handling, Strings etc
4	Sampling , Probabilities of selection of samples, etc
5	Random Sampling, Estimations etc
6	BootStrap methodology
7	Object Oriented Programming using R
8	Data Handling - Importing data, Accessing data, Exploring data
9	Data Visualization using R
10	Data Pre processing using R
11	Data Manipulation using R
12	Statistical Analysis – Simple use cases
13	Statistical Analysis – Advanced usecases

Course outcomes:

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

1. To **Understand** , **Learn** and **Apply** the basic programming concepts of R
2. To **Synthesize** Data handling, sampling, pre processing and boot strapping using R
3. To **Develop** the logics with object oriented programming concepts using R
4. To **Compute** and **Analyse** statistical data using R

Conduct of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Breakup of marks and the instructions printed on the cover page of answer script to be strictly adhered by the examiners.
- Students can pick one experiment from the questions lot prepared by the examiners.
- Change of experiment is allowed only once and 20% Marks is to be deducted.

Course Articulation matrix (CO-PO and CO-PSO mapping)

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2
CO 1	2		2											2
CO 2	3		3											
CO 3	3		3											2
CO 4	3	2	3											2
CO 5	2		2											2