Scheme of Teaching, Examination and Syllabus

B.E. COMPUTER SCIENCE & ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Batch: 2023-24

Second Year

(III and IV SEMESTER)

(Effective from the academic year 2024-2025)

SreeSiddaganga 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 Artificial Intelligence & Machine Learning — 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	3.5(48 hrs)	-	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)			
S3CCSA01	Project Management with GiT	S3CCSA03	PHP Programming
S3CCSA02	Data Analytics with Excel	S3CCSA04	Parallel Programming

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 Artificial Intelligence & Machine Learning — 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	S4CCSI01	Artificial Intelligence (I)	Dept.	3	0	2	3.5(50 hrs)	3	50	50	100	4
3.	IPCC	S4CCSI02	Data Science with Python (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)

S4CCSA01	Scala	S4CCSA03	R Programming
S4CCSA02	MERN Stack	S4CCSA04	Unix and Shell 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

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Batch: 2023-24

III SEMESTER

(Effective from the academic year 2024-2025)

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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	40hrs	Practical Hours	-

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

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – III

OPERATING SYSTEM

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	40hrs	Practical Hours	26hrs

Courseobjectives: The 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 analyse** 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

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

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		

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

(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

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

SEMESTER – III

DIGITAL CIRCUITS & COMPUTER ORGANIZATION

Course Code	S3CCSI02	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	4
Lecture Hours	40hrs	Practical Hours	26hrs

Course objectives: The course will enable students to

1. **Explain** the basic types of memory and cache mapping functions**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 signalsand 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
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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	40hrs	Practical Hours	26hrs

Course objectives: The 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 (08Lhrs)

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 (08Lhrs)

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 **(08L 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
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – III

DATA STRUCTURES AND APPLICATIONSLABORATORY

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

Course objectives: The 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.

<table>
<tr><th>Sl. no.</th><th>Programs</th></tr>
<tr><td>1</td><td>Develop a C program to create a sequential file for storing employee records with each record having following information:
<table><tr><th>Employee_Id</th><th>Name</th><th>Department</th><th>Salary</th><th>Age</th></tr><tr><td>Non-Zero Positive integer</td><td>25 Characters</td><td>25 Characters</td><td>Positive Integer</td><td>Positive integer</td></tr></table>
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.</td></tr>
<tr><td>2</td><td>Develop a C program to implement Stack of names to perform the push, pop and display operations.</td></tr>
<tr><td>3</td><td>Develop a C program to convert a valid infix expression to postfix.</td></tr>
<tr><td>4</td><td>Develop a C program to evaluate the given postfix expression.</td></tr>
<tr><td>5</td><td>Develop a C program to implement Linear Queue of characters to perform the insertion, deletion and display operations.</td></tr>
<tr><td>6</td><td>Develop a C program to implement Circular Queue of integers to perform the insertion, deletion and display operations.</td></tr>
<tr><td>7</td><td>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.</td></tr>
<tr><td>8</td><td>Develop a C program to add two two-variable polynomials using Singly Linked list.</td></tr>
<tr><td>9</td><td>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.</td></tr>
<tr><td>10</td><td>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</td></tr>
</table>

	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 bythe 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
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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	26hrs	Practical Hours	26hrs

Course objectives: The 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 (8hrs)

AN OVERVIEWOFJAVA: Object-OrientedProgramming,AFirstSimpleProgram,ASecondShortProgram

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 (8hrs)

Packages and Interfaces

Packages: Defining a Package, Finding Packages and CLASSPATH, A Short Package Example,Accessprotection: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 (8hrs)

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 hrs)

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 hrs)

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, 2007	Herbert Schildt	Tata McGrawhill	9th Edition,2014
Reference Books				
1	Object-Oriented Programming With JAVA Essentials and Applications	RajKumarBuyya, S Thamarai Selvi	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		

	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

JAVA Programming Laboratory-PLC3 LAB	
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.
3	USN, NAME, BRANCH, PHONE, PERCENTAGE
4	Write a JAVA program to create n Student objects and print the USN, Name, Branch, Phone, and percentage of these objects with suitable headings.
5	Write a JAVA program demonstrating Method overloading and Constructor overloading.
6	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.
7	Demonstrate dynamic dispatch using abstract class in JAVA.
8	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.
9	Write a java program to perform simple command line calculator with an exception handler.
10	Develop a Program to Launch the browser and open a specific URL
11	Write a Java program that reads a text file and displays the contents on the screen.
12	Write a Java program to check whether the given element is present in a given array or not using generimethod.

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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	26hrs	Practical Hours	26hrs

Course objectives: ThisCourse 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,UniformResourceLocators, 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)

UNIT-4: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, aElement,RelativeURLs,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)**

UNIT-5: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. CreateanXHTMLpage usingtags toaccomplishthefollowing:
   (i) Aparagraphcontainingtext"Allthat glittersisnot gold".Boldfaceanditalicizethis text
   (ii) Createequation: $x=1/3(y^2+z^2)$
   (iii) Putabackgroundimagetoapage anddemonstrate allattributesofbackgroundimage
   (iv) Createunorderedlistof5fruitsandorderedlistof3flowers
2. Using MathML, writethe mathematicalexpressions for the following. Use separate div for the equations. Insert suitable title, background colour, text colour for each div.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \overline{^{2}-4ac} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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.UseHTML5 forperformingfollowingtasks:
(i) Drawthe following shapes using HTML5SVG: Square: fillthesquarewithgreencolorandmake6pxbrownstrokewidth. Rectangle: Fillthe rectanglewithbluecolorandmake4pxblackstrokewidth. circleusingHTML5SVG. Ellipse:filltheellipsewithgreencolorandmake3pxbrownstrokewidth
(ii)Writethe followingmathematicalexpression byusingHTML5MathML.d=x^2-y^2
(iii)Redirectingcurrentpagetoanotherpageafter5 secondsusingHTML5 metatag

4.DemonstratethefollowingHTML5Semantictags-<article>,<aside>,<details>,<figcaption>, <figure>,<footer>,<header>,<main>,<mark>,<section>forawebpagethatgivesinformationabouttravel experience.

6.Changethetag**li** to havethe followingproperties:

• A displaystatus of inline • Amedium,double-lined,blackborder • Noliststyletype Addthefollowingproperties to thestylefor **li**: • Marginof5px • Padding of 10px to the top, 20px to the right, 10px to the bottom, and 20px tothe left Alsodemonstrate liststyletypewithuserdefinedimagelogos
7.Createsign upwebpageusingHTMLand CSSwith tabular layout
8.Createcalculatorinterfacewith HTMLand 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 fromleftto right with asmalldelay
11. Create a webpage containing 3 overlapping images using HTML, CSS and JS. Further when themouseis over anyimage,it should beon the top and fullydisplayed.

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:TheCompleteR	ThomasA.Powell	TataMcGraw Hill	FifthEdition
2	WEBPROGRAMMINGwithHTML5,CSSandJavaScript	JohnDean,Jones&BartlettLearning	Jones & Bartlett Learning	FirstEdition
Reference Books				
1	ProgrammingtheWorldWideWeb	RobertW Sebesta	PearsonEducation	Seventh Edition 2017
2	HTML:ABeginner'sGuide	WendyWillard	McGraw-HillEducation	Fourth Edition, 2009
3	HTML&CSS:TheCompleteReference	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 & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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	3
Lecture Hours	26hrs	Practical Hours	26hrs

Course objectives: The course will enable students to

1. ImplementPythonprogramsusingPythonlanguageconstruct
2. Understandvarious datastructuresprovided byPython library
3. Usedifferentlibrariesforscientificanddataintensiveapplications
4. Buildreal-worldapplicationsusingOOP,filesandexceptionhandlingprovidedbyPython
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 Nongreedy 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.pathUNIT, 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 shutilUNIT, Walking a Directory Tree, Compressing Files with the zipfileUNIT, 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 webbrowser UNIT, Downloading Files from the Web with the requests UNIT, Saving Downloaded Files to the Hard Drive, HTML, Parsing HTML with the BeautifulSoupUNIT

Working with Excel Spreadsheets: Excel Documents, Installing the openpyxlUNIT, 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, ThejsonUNIT, 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, tuplesand 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 bythe 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.

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

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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	26hrs	Practical Hours	26hrs

Course objectives: The 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++	SouravSahay	Oxford University Press	1st edition, 2009

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**	

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	Developa 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	Developa 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	Developa vector class template for performing the scalar product of int type vectors as well as float type vectors.
8	Developa 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. Usedisplay( ) function to display the contents of the class. Create pointers to media class to accessthe 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 textinto the file and then read the text from the file.

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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 Hours	-	Practical Hours	26hrs

Course objectives: The 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:

LearningOutcomes: 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 Methodthat will be adopted by entire semester by agroup of students. They will also make an excerpt either as a documentary or a photoblogdescribingtheplant's origin,itsusageindailylife,anditsappearanceinfolkloreandliterature

UNIT-2 **(6Hrs)**

Heritagewalk andcrafts corner: Heritagetour,knowingthehistory and cultureofthe city,connecting to people around through their history, knowing the city and its craftsman, photoblogand documentaryonevolutionandpracticeof variouscraftforms.

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)**

FoodWalkCity'sculinarypractices,foodlore,andindigenousmaterialsoftheregionusedin 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 & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – III

PROJECT MANAGEMENT WITH GIT

Course Code	S3CCSA01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hours	-	Practical Hours	26hrs

Course objectives: The course will enable students to

1. TofamiliarwithbasiccommandofGit
2. Tocreateandmanagebranches
3. Tounderstandhowto collaborateandwork withRemoteRepositories.
4. Tofamiliarwithvirioncontrollingcommands

Sl. no.	Experiments
1	**SettingUpandBasicCommands** Initializeanew Git repositoryina directory. Createanewfileand addit tothestagingareaandcommit the changeswithan appropriatecommit message.
2	**CreatingandManagingBranches** Createanewbranchnamed"feature-branch."Switchtothe"master"branch.Mergethe"feature-branch"into "master."
3	**CreatingandManagingBranches** Writethecommandstostashyourchanges,switchbranches,andthenapplythestashedchanges.
4	**CollaborationandRemoteRepositories** ClonearemoteGitrepositorytoyourlocalmachine.
5	**CollaborationandRemoteRepositories** Fetchthelatestchangesfromaremoterepositoryandrebaseyourlocalbranchontotheupdatedremotebranch.
6	**CollaborationandRemoteRepositories** Writethecommandtomerge"feature-branch"into"master"whileprovidingacustomcommitmessageforthemerge.
7	**GitTagsandReleases** WritethecommandtocreatealightweightGittagnamed"v1.0"fora commitinyourlocalrepository.
8	**AdvancedGitOperations** Writethecommandtocherry-pickarangeofcommitsfrom"source-branch"tothecurrent branch.
9	**AnalysingandChangingGitHistory** GivenacommitID,howwouldyouuseGittoviewthedetailsofthatspecificcommit,includingthe author, date, and commit message?
10	**AnalysingandChangingGitHistory** Write the command to list all commits made by the author "JohnDoe" between "2023-01-01"and "2023-12-31."
11	**AnalysingandChangingGitHistory** Writethecommand todisplaythelast fivecommits intherepository'shistory.
12	**AnalysingandChangingGitHistory** WritethecommandtoundothechangesintroducedbythecommitwiththeID"abc123".

Course outcomes:

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

1. Usethebasicscommands relatedtogitrepository
2. Createandmanagethebranches
3. ApplycommandsrelatedtoCollaborationandRemoteRepositories
4. Usethecommandsrelated to Git Tags, Releases and advanced git operations
5. Analyse and change the git history

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – III

DATA ANALYTICS WITH EXCEL

Course Code	S3CCSA02	CIE Marks	50
TeachingHours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hours	-	Practical Hours	26hrs

Course objectives: The course will enable students to

- ToApplyanalysis techniquesto datasetsin Excel
- LearnhowtousePivotTablesandPivotChartstostreamlineyourworkflowinExcel
- UnderstandandIdentifytheprinciplesofdata analysis
- BecomeadeptatusingExcelfunctionsandtechniquesforanalysis
- Buildpresentationreadydashboards inExcel

Sl. no.	Experiments
1	**GettingStartedwithExcel**:Creationofspreadsheets, Insertionof rowsandcolumns,Drag&Fill, use ofAggregatefunctions.
2	**WorkingwithData** : Importingdata, DataEntry&Manipulation,Sorting&Filtering.
3	**Workingwith Data:**Data Validation,Pivot Tables &PivotCharts.
4	**DataAnalysisProcess**:ConditionalFormatting,What-If Analysis,Data Tables, Charts& Graphs.
5	**CleaningDatawithTextFunctions**:useofUPPERandLOWER,TRIMfunction,Concatenate.
6	**CleaningDataContainingDateandTimeValues:**useofDATEVALUEfunction,DATEADDandDATE DIF, TIMEVALUE functions.
7	**ConditionalFormatting**:formatting,parsing,andhighlightingdatainspreadsheetsduringdataanalysis.
8	**WorkingwithMultipleSheets**:workwithmultiplesheetswithinaworkbookiscrucialfororganizingandmanagingdata,performcomplexcalculationsandcreatecomprehensive reports.
9	Createworksheetwithfollowingfields:Empno,Ename,BasicPay(BP),TravellingAllowance(TA), Dearness Allowance(DA), House Rent Allowance(HRA), Income Tax(IT),ProvidentFund(PF),NetPay(NP).Useappropriateformulastocalculatetheabovescenario. Analysethedatausingappropriate chartandreportthedata.
10	CreateworksheetonInventoryManagement:SheetshouldcontainProductcode,Productname,Product type,MRP,Costafter%ofdiscount,Dateofpurchase.Useappropriate formulastocalculatetheabovescenario.Analysethedatausingappropriatechartandreportthedata.
11	CreateworksheetonSalesanalysisof MerchandiseStore:dataconsistingofOrderID,Customer ID, Gender, age, date of order, month, online platform, Category of product, size,quantity,amount,shippingcityandotherdetails.Useofformulato segregatedifferent categoriesandperforma comparativestudyusingpivottables anddifferentsortofcharts.
12	Generationofreport&presentationusingAutofilter&macro.

Course outcomes:

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

1. Useadvancedfunctions andproductivitytoolstoassistindevelopingworksheets.
2. ManipulatedatalistsusingOutlineandPivotTables.
3. UseConsolidationtosummariseandreportresultsfrommultipleworksheets.
4. Apply Macros and Autofiltertosolvethegiven realworldscenario.

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – III

PHP PROGRAMMING

Course Code	S3CCSA03	CIE Marks	50
TeachingHours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hours	-	Practical Hours	26hrs

Course objectives: The course will enable students to

- TointroducethePHPsyntax,elements,andcontrolstructures
- TomakeuseofPHPFunctionsandFilehandling
- ToillustratetheconceptofPHParraysandOOPs

Sl. no.	Experiments
	AIM: IntroductiontoHTML/PHPenvironment,PHPDataTypes,Variables,Literals,andoperators
1	a. DevelopaPHPprogramtocalculateareasofTriangleandRectangle. DevelopaPHPprogramtocalculateCompoundInterest.
2	Demonstratingthevariousformstoconcatenatemultiplestrings Developp rogram(s)todemonstrateconcatenationofstrings: (i) Stringsrepresentedwithliterals(singlequoteordoublequote) (ii) Stringsasvariables (iii) Multiplestringsrepresentedwithliterals(singlequoteordoublequote)andvariables (iv) Stringsandstringvariables containingsinglequotesaspartstringcontents StringscontainingHTMLsegmentshavingelementswithattributes
3	a. DevelopaPHPProgram(s)tocheckgivennumberis: (i) Oddoreven (ii) Divisiblebyagivennumber(N) (iii) Squareofaanothernumber DevelopaPHPProgramtocomputetherootsofaquadraticequationbyacceptingthecoefficients.Printtheappropriate messages.
4	a. DevelopaPHPprogram tofindthesquareroot ofanumberbyusingthenewton'salgorithm. DevelopaPHPprogramtogenerateFloyd'striangle.
5	a. DevelopaPHPapplicationthatreadsalistofnumbersandcalculatesmeanandstandarddeviation. Develop a PHP application that reads scores between 0 and 100 (possibly including both 0 and 100)and creates a histogram array whose elements contain the number of scores between 0 and 9, 10 and19, etc. The last "box" in the histogram should include scores between 90 and 100. Use a function togeneratethe histogram.
6	a. DevelopPHPprogramtodemonstratethedate()withdifferentparameteroptions. DevelopaPHPprogramtogeneratetheFibonacciseriesusingarecursivefunction.
7	DevelopaPHPprogramtoacceptthefileandperformthefollowing (i) PrintthefirstNlinesofa file Update/Addthecontentofa file
8	DevelopaPHPprogramtoreadthecontentofthefileandprintthefrequencyofoccurrenceofthewordaccepted bythe userinthe file
9	DevelopaPHPprogramtofiltertheelementsofanarraywithkeynames. SampleInputData: 1st array: ('c1' => 'Red', 'c2' => 'Green', 'c3' => 'White', c4 => 'Black') 2ndarray:('c2','c4')

	Output: Arra y([c1] => Red[c3]=> White)
10	DevelopaPHPprogramthatillustratestheconceptofclassesandobjectsbyreadingandprintingemployeedata,including Emp_Name,Emp_ID, Emp_Dept,Emp_Salary,andEmp_DOJ.
11	a. DevelopaPHPprogramtocounttheoccurrencesofAadhaarnumberspresentinatext. DevelopaPHPprogramtofindtheoccurrencesofagivenpatternandreplacethemwithatext.
12	DevelopaPHPprogram toreadthecontentsofaHTMLformanddisplaythecontentsona browser.
NOTE:NecessaryHTMLelements(andCSS)canbeusedfordesigningtheexperiments.	

Course outcomes:

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

1. Apply basic concepts of PHP to develop web program
2. Develop programs in PHP involving control structures
3. Develop programs to handle structured data (object) and data items (array)
4. Develop programs to access and manipulate contents of files
5. Use super-global arrays and regular expressions to solve real world problems.

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – III

PARALLEL PROGRAMMING

Course Code	S3CCSA04	CIE Marks	50
TeachingHours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hours	-	Practical Hours	26hrs

Course objectives: The 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

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 bythe 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 detucted.

Scheme of Teaching, Examination and Syllabus

B.E. COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE &MACHINE LEARNING)

Batch: 2023-24

IV SEMESTER

(Effective from the academic year 2024-2025)

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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 Hours	40hrs	Practical Hours	-

Course objectives: The 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.	AnanyLevitin	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	

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – IV

ARTIFICIAL INTELLIGENCE(I)

Course Code	S4CCSI01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	3
Lecture Hours	40hrs	Practical Hours	26hrs

Course objectives: The course will enable students to

1. Understand and apply intelligent agents and searching algorithms to solve problems.
2. Understand knowledge based agents and solve knowledge based engineering problems using first order logic.
3. Compare propositional and first order inference, forward and backward chaining.
4. Understand stochastic approaches to solve Planning and uncertainty problems.

UNIT-1 (8L+6P hrs)

What is AI? Acting humanly and thinking humanly, thinking rationally and acting rationally, Intelligent Agents: Agents and Environments, Good Behavior: The concept of Rationality: Rationality, Omniscience, Learning and autonomy, The nature of Environments: specifying the task environment, properties of task environments, The structure of Agents: Agent Programs, simple reflex agents, Model-based reflex agents, Goal-based agents, Utility-based agents, Learning agents, How the components of agents programme work. Solving problems by Searching: problem-solving agents, well-defined problems and solutions, Example problems.Searching for Solutions: infrastructure for search algorithms, measuring problem-solving performance

Text book 1: Chapter 1:1.1, Chapter 2, Chapter 3: 3.1-3.3

UNIT-2 (8L+8P hrs)

Uninformed Search strategies: BFS, uninform-cost search, DFS, depth-limited search, iterative deepening depth-first search, bidirectional search, comparing uniformed search strategies, Informed search strategies: Greedy best-first search, A* search, Memory-bounded heuristic search, learning to search better.

Adversarial search: Games, Optimal Decisions in Games- The minimax algorithm, Optimal decisions in multiplayer games, Alpha–Beta Pruning.

Text book 1: Chapter 3.4, 3.5, chapter 5: 5.1, 5.2, 5.3

UNIT-3 (8L+6P hrs)

Constraint satisfaction problems: Example problem: Map coloring, Example problem: Job-shop scheduling, Variations on the CSP formalism, Constraint Propagation: Inference in CSPs:Node consistency, Arc consistency, Path consistency, *K*-consistency, Global constraints, Backtracking search for CSPs; Variable and value ordering, Interleaving search and inference, Intelligent backtracking: Looking backward.

Knowledge-based agents; The wumpus world as an example world, Logic; propositional logic: a Very Simple logic A simple knowledge base, A simple inference procedure, Propositional theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining. First order logic: Syntax and semantics of first-order logic, Using first-order logic, Knowledge engineering infirst-orderlogic.

Text book 1: Chapter 6:6.1-6.3, Chapter 7:7.1-7.5, Chapter 8: 8.2-8.4

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

Inference In First-Order Logic: Propositional vs. First-order Inference, Unification and Lifting, Forward chaining: First-order definite clauses, A simple forward-chaining algorithm, Efficient forward chaining, backward chaining: A backward-chaining algorithm, Resolution: Conjunctive normal form for first-order logic, The resolution inference rule, Example proofs.

Knowledge Representation: Ontological Engineering, Categories and Objects, Events, Mental events and Mental Objects. Reasoning systems for Categories, Reasoning with default information.

Text book 1: Chapter 9, Chapter 10 (12 in 3rd edition)

UNIT-5 **(7L+2P hrs)**

Classical Planning: Definition of Classical Planning, Algorithms For Planning as State-Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning Approaches

QuantifyingUncertainty: Acting under Uncertainty, Basic Probability Notation, Inference using Full Joint Distributions, Independence, Bayes' Rule and its Use, The Wumpus World Revisited

Text book 1: Chapter 11 (10 in 3 rded), 12 (13 in 3rdedition)

Course outcomes:

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

1. Analyse and develop artificial intelligent agents for simple applications.
2. Apply searching algorithms to develop artificial intelligence agents.
3. Develop knowledge base sentences with propositional logic and first order logic.
4. Design AI representational system for reasoning about simple applications
5. Apply and analyse the stochastic approaches to Planning and uncertainty.

Sl. No	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Artificial Intelligence - A Modern Approach	Stuart Russel Peter Norvig	Pearson Education	3rd Edition 2013/4th Edition 2021
Reference Books				
1	Artificial Intelligence	Elaine Rich	Tata McGraw Hill	3rd Edition, 2019
2	Introduction to Artificial Intelligence	Wolfgang Ertel	Springer	2nd Edition, 2017
3	Prolog Programming for Artificial Intelligence	Ivan Bratko	Addison-Wesley	3rd Edition 2012
4	Artificial intelligence: Structures and Strategies for Complex Problem Solving	George F Luger	Pearson Education	6th Edition, 2011

Sl No.	Experiments
1	Write a program to find the possible solutions for given n-queen's problem.
2	Write a program to find the possible solutions for given n-Puzzle problem.
3	Solve any problem using depth first search.

4	Solve any problem using breadth first search.
5	Solve any problem using iterative deepening search.
6	Solve any problem using Uniform cost search
7	Solve the problem using best first search.
8	Solve any searching problem using A* algorithm using Straight line distance heuristics.
9	Find the appropriate solution by implementing backtracking search to solve Constraint Satisfaction Problem.
10	Write a program to implement Resolution algorithm for first order logic

Course outcomes for lab:

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

1. Apply AI techniques to solve search problems.
2. Design and implement search problem using Python/PROLOG.
3. Design and implement CSP problem using Python/PROLOG.
4. Design and implement learning algorithms using Python/PROLOG.

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		
CO3			2										2		
CO4			2										2		
CO5		1											1		
Overall CO		2	2										1		

Program articulation matrix:

Course Outcomes	PROGRAM 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
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – IV

DATA SCIENCE WITH PYTHON(I)

Course Code	S4CCSI02	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	3
Lecture Hours	40hrs	Practical Hours	26hrs

Course objectives: The course will enable students to

1. Describe the concept of data science, its scope in business and explain the available techniques.
2. Understand Predictive modeling, explain supervised segmentation and select (through solving) the attribute for segmentation using the available techniques.
3. Explain the concept of Classification and classify (solve) a given data set.
4. Understand and describe the concept of similarity, neighbors and clustering and apply it for any real-world data.
5. Explain the concepts of evaluating the model performance.
6. Describe the concepts of association rule mining and ensemble modeling.

UNIT-1 (8hrs)

Introduction: Data-Analytic Thinking: The Ubiquity of Data Opportunities, Example: Hurricane Frances, Predicting Customer Churn. Data Science, Engineering, and Data-Driven Decision Making, Data Processing and "Big Data", Data and Data Science Capability as a Strategic Asset, Data-Analytic Thinking.

Business Problems and Data Science Solutions: From Business Problems to Data Mining Tasks, Supervised Versus Unsupervised Methods, Data Mining and Its Results, The Data Mining Process, Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, Deployment, Other Analytics Techniques and Technologies: Statistics, Database Querying, Data Warehousing, Regression Analysis, Machine Learning and Data Mining.

Textbook 1: Chapter 1,2

UNIT-2 (8hrs)

Introduction to Predictive Modeling: From Correlation to Supervised Segmentation Models, Induction, and Prediction, Supervised Segmentation, Selecting Informative Attributes Example: Attribute Selection with Information Gain, Supervised Segmentation with Tree- Structured Models, Visualizing Segmentations, Trees as Sets of Rules, Probability Estimation, Example: Addressing the Churn Problem with Tree Induction.

Textbook 1: Chapter 3

UNIT-3 (8hrs)

Fitting a Model to Data: Classification via Mathematical Functions: Linear Discriminant Functions, Optimizing an Objective Function, An Example of Mining a Linear Discriminant from Data, Linear Discriminant Functions for Scoring and Ranking Instances, Support Vector Machines briefly, Regression via Mathematical Functions, Class Probability Estimation and Logistic "Regression". Logistic Regression: Some Technical Details. Example: Logistic Regression versus Tree Induction, Non-Linear Functions, Support vector machines and Neural Networks

Over fitting and Its Avoidance: Fundamental Concepts, Exemplary Techniques, Regularization, Generalization, Overfitting, Overfitting Examined.

Textbook 1: Chapter 4,5

UNIT-4 **(8 hrs)**

Similarity, Neighbors, and Clusters: Similarity and Distance, Nearest-Neighbor Reasoning, Example: Whiskey Analytics, Nearest Neighbors for Predictive Modeling, How Many Neighbors and How Much Influence? Geometric Interpretation, Overfitting, and Complexity Control. Issues with Nearest-Neighbor Methods. Some important Technical Details Relating to Similarities and neighbors. Clustering, Example: Whiskey Analytics Revisited, Hierarchical Clustering, Nearest Neighbors Revisited: Clustering Around Centroids. Understanding the Results of Clustering.

Textbook 1: Chapter 6

UNIT-5 **(8 hrs)**

Decision Analytic Thinking: What is a Good Model?: Evaluating Classifiers Plain Accuracy and its Problems, The confusion matrix, Problems with unbalanced Classes, Problems with Unequal Costs and Benefits.

Other Data Science Tasks and Techniques: Co-occurrences and Associations: Finding Items That Go Together, Measuring Surprise: Lift and Leverage, Example: Beer and Lottery Tickets, Associations

Ensemble methods: Rationale for Ensemble Method, Methods for Constructing an Ensemble Classifier, Bagging, Boosting, Random Forests

Textbook 1: Chapter 7,10,12

Textbook 2: Chapter 5.6

Course outcomes:

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

1. Apply the knowledge of mathematics to explain the concept of data science, the available techniques in data science and its scope in business.
2. Develop a Decision tree based on supervised segmentation and predict the class for a given data.
3. Analyze the given data set and develop linear models to classify the given data.
4. Develop solutions to group entities in data set and apply it for the given real-world data using the basic knowledge of similarity, neighbour's and clustering
5. Analyze the given data and formulate the association rules based on market basket analysis.

Sl. No.	Title of the Book	Name of the Author/s	Name of the Publisher	Edition and Year
Textbooks				
1	Data Science for Business	Foster Provost and Tom	O'Reilly Media	2013
2	Introduction to Data Mining	Pang-Ning Tan Michael Steinbach Vipin Kumar	Pearson Education Limited	2014
Reference Books				
1	Doing Data Science, ,	Rachel Schutt& Cathy	O'Reilly Media	2013
2	Python Data Science Handbook Essential Tools for Working with Data	Jake VanderPlas	O'Reilly Media	2022

Data Science lab experiments:

Week 1: Students need to explore the basic packages such as Pandas, NumPy, matplotlib and basic operation in python related to data science.

The following list of programs should be implemented using Python from week 2.

1. Write a program to develop decision tree to classify the given dataset. Print the confusion matrix and accuracy of the model developed.

2. Develop a program to develop Simple Linear Regression and Support Vector Regression models. Apply the models to the given dataset. Plot the actual vs predicted values. Compare the model through the regression error metrics namely MAE, MAD, MAPE, RMSE and R2 score.

3. Develop a program for Logistic Regression. Apply the regression model to the given dataset. Print the confusion matrix and plot the ROC and AUC curves.

4. Develop a program to implement k-Nearest Neighbour algorithm to predict the value of the target variable in the data set. Use distance-based weights of the neighbors. Identify the best ***k*** for the given dataset through *elbow* method and plot the same.

5. Develop k-Means algorithm for clustering. Identify the best k through *Silhouette* method and plot related graphs to justify your k selection.

6. Develop a program to develop Agglomerative clustering and hierarchical clustering models. Print the dendrogram of each method and compare the results of these two algorithms also comment on the quality of clustering.

7. Write a program to demonstrate the working of Apriori-algorithm for the given transactions. Print all the rules based on support and confidence.

8. Write a program to demonstrate the working of Random forests for classification. For the given data, fine tune the parameters through grid search and print the accuracy of the model.

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														2
CO2	2	2	2												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:

Course Outcomes	PROGRAM 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
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – IV

DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

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

Course objectives: The 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. b. Compute the transitive closure of a given directed graph using Warshall's algorithm.
9	Implement All Pair Shortest paths problem using Floyd'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	Implement N Queen's problem using Back Tracking.
14	Implement Travelling Salesman Problem using branch and bound technique.

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 bythe 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	

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – IV

DISCRETE MATHEMATICAL STRUCTURES

Course Code	S4CCS03	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3	Exam Hours	03
Lecture Hours	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 Function8 **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: Groups8 **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**,** understandisomorphism and applyGraph 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

		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**											

Mapping of Course outcomes to Program outcomes

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

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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 Hours	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 th

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.

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

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

B B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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 Hours	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).

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

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

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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 Hours	40hrs	Practical Hours	-

Course objectives:

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 problemsFinite 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

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
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
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 Hours	40hrs	Practical Hours	-

Course objectives: The 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
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – IV

SCALA

Course Code	S4CCSA01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(1:0:0)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hours	13hrs	Practical Hours	-

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

Course objectives: The 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				
Overal l 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**

Question Paper Pattern

CIE: CIE pattern may be hybrid type with MCQs and descriptive questions.

- 10 Marks MCQs
- 3 descriptive questions of 5 marks each
- All the questions are compulsory

SEE: SEE pattern may also be hybrid type with MCQs and descriptive questions.

- 20 Marks MCQs
- 1 descriptive question of 10 marks from each of the units (total 3 questions)

  or

  2 questions of 5 marks from each of the units (total 6 questions)
  - All the questions are compulsory

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – IV

MERN STACK

Course Code	S4CCSA02	CIE Marks	50
TeachingHours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hours	-	Practical Hours	26hrs

Course objectives: The 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 bythe 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 & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – IV

R PROGRAMMING

Course Code	S4CCSA03	CIE Marks	50
TeachingHours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hours	-	Practical Hours	26hrs

Course objectives: The course will enable students to

1. ToexploreandunderstandhowRandRStudiointeractiveenvironment.
2. Tounderstandthedifferent dataStructures,datatypesinR.
3. TolearnandpracticeprogrammingtechniquesusingRprogramming.
4. ToimportdataintoRfromvariousdatasourcesandgeneratevisualization.
5. Todrawinsightsfromdatasetsusingdataanalyticstechniques.

Sl. no.	Experiments
1	DemonstratethestepsforinstallationofRandRStudio. Performthefollowing: a) Assign different type of values to variables and display the type of variable. Assign different typessuch as Double, Integer, Logical, Complex and Character and understand the difference betweeneachdatatype. b) DemonstrateArithmeticandLogicalOperationswithsimpleexamples. c) Demonstrategenerationofsequencesandcreationofvectors. d) DemonstrateCreationofMatrices e) DemonstratetheCreationofMatricesfromVectorsusingBindingFunction. f) Demonstrateelementextractionfromvectors,matricesandarrays **Suggested Reading** – Text Book 1 – Chapter 1 (What is R, Installing R, Choosing an IDE – RStudio, How toGetHelpinR,InstallingExtraRelatedSoftware),Chapter2(MathematicalOperationsandVectors,AssigningVariables,SpecialNumbers,LogicalVectors),Chapter3(Classes,DifferentTypesofNumbers, OtherCommonClasses,CheckingandChangingClasses,ExaminingVariables)
2	Assess the Financial Statement of an Organization being supplied with 2 vectors of data: Monthly RevenueandMonthlyExpensesfortheFinancialYear.Youcancreateyourownsampledatavectorforthisexperiment)Calculatethefollowingfinancial metrics: a. Profitforeachmonth. b. Profitaftertaxforeachmonth(TaxRateis30%). c. Profitmarginforeachmonthequalstoprofitaftertaxdividedbyrevenue. d. GoodMonths– wheretheprofitaftertaxwasgreaterthanthemeanfortheyear. e. BadMonths– wheretheprofitaftertax waslessthanthemeanfortheyear. f. Thebestmonth –wherethe profitaftertaxwasmaxfortheyear. g. Theworstmonth–wheretheprofitaftertaxwasminfortheyear. **Note:** a. AllResultsneedtobepresentedasvectors b. ResultsforDollarvaluesneedtobecalculatedwith$0.01precision,butneedtobepresentedinUnitsof$1000(i.e1k)with nodecimal points c. Resultsfortheprofitmarginrationeedtobepresentedinunitsof%withnodecimalpoint. d. Itisokayfortaxtobenegativeforanygivenmonth(deferredtaxasset) e. GenerateCSVfileforthedata. **SuggestedReading**– TextBook1–Chapter4(Vectors,CombiningMatrices)
3	Developaprogramtocreatetwo3X3matricesAandBandperformthefollowingoperationsa)Transposeofthematrixb)additionc)subtractiond)multiplication **SuggestedReading**– TextBook1–Chapter4(MatricesandArrays–ArrayArithmetic)
4	Developaprogramtofindthefactorialofgivennumberusingrecursivefunctioncalls. **SuggestedReading**–ReferenceBook1–Chapter5(5.5 –RecursiveProgramming)

<table>
<tr><td></td><td>TextBook1–Chapter8(FlowControlandLoops–IfandElse,VectorizedIf,whileloops,forloops),Chapter6(Creatingand CallingFunctions, PassingFunctionsto and fromotherfunctions)</td></tr>
<tr><td>5</td><td>DevelopanRProgramusingfunctionstofindalltheprimenumbersuptoaspecifiednumberbythemethodof Sieveof Eratosthenes.
SuggestedReading–ReferenceBook
1-Chapter5(5.5–RecursiveProgramming)
TextBook1–Chapter8(FlowControlandLoops–IfandElse,VectorizedIf,whileloops,forloops),Chapter6(Creatingand CallingFunctions, PassingFunctionsto and fromotherfunctions)</td></tr>
<tr><td>6</td><td>Thebuilt-indatasetmammalscontaindata onbodyweightversusbrainweight.DevelopRcommandsto:
a) FindthePearsonandSpearmancorrelationcoefficients.Aretheysimilar?
b) Plotthedatausingtheplotcommand.
c) Plotthelogarithm(log)ofeachvariableandseeifthatmakesadifference.
SuggestedReading–TextBook1–Chapter12–(Built-inDatasets)Chapter14 –(Scatterplots)ReferenceBook2–13.2.5(CovarianceandCorrelation)</td></tr>
<tr><td>7</td><td>DevelopRprogramtocreateaDataFramewithfollowingdetailsanddothefollowing operations.
<table>
<tr><th>itemCode</th><th>itemCategory</th><th>itemPrice</th></tr>
<tr><td>1001</td><td>Electronics</td><td>700</td></tr>
<tr><td>1002</td><td>DesktopSupplies</td><td>300</td></tr>
<tr><td>1003</td><td>OfficeSupplies</td><td>350</td></tr>
<tr><td>1004</td><td>USB</td><td>400</td></tr>
<tr><td>1005</td><td>CDDrive</td><td>800</td></tr>
</table>
a) SubsettheDataframeanddisplaythedetailsofonlythoseitemswhosepriceisgreaterthanorequalto350.
b) SubsettheDataframeanddisplayonlytheitemswherethecategoryiseither“OfficeSupplies”or “DesktopSupplies”
c) CreateanotherDataFramecalled“item-details”withthreedifferentfieldsitemCode,ItemQtyonHandandItemReorderLvl andmerge the twoframes
SuggestedReading–Textbook1:Chapter5(Listsand DataFrames)</td></tr>
<tr><td>8</td><td>Let us use the built-in dataset air quality which has Daily air quality measurements in New York, May toSeptember1973.DevelopRprogramtogeneratehistogrambyusingappropriateargumentsforthefollowing statements.
a) Assigningnames,usingtheairqualitydataset.
b) ChangecolorsoftheHistogram
c) RemoveAxisandAddlabelstoHistogram
d) ChangeAxislimits ofaHistogram
e) AddDensitycurvetothehistogram
Suggested Reading–Reference Book 2–Chapter 7 (7.4–The ggplot2 Package), Chapter 24 (SmoothingandShading)</td></tr>
<tr><td>9</td><td>Design a data frame in R for storing about 20 employee details. Create a CSV file named “input.csv” thatdefinesalltherequiredinformationabouttheemployeesuchasid,name,salary,start_date,dept.ImportintoRanddoth e followinganalysis.
a) Findthetotalnumberrows&columns
b) Findthemaximumsalary
c) Retrievethedetailsoftheemployeewithmaximumsalary
d) Retrieveallthemployeesworkingin theITDepartment.
e) RetrievetheemployeesintheITDepartmentwhosesalaryisgreaterthan20000andwritethese detailsintoanotherfile “output.csv”
SuggestedReading–TextBook1–Chapter12(CSVandTabDelimitedFiles)</td></tr>
<tr><td>10</td><td>Using the built in dataset mtcars which is a popular dataset consisting of the design and fuel consumptionpatternsof32differentautomobiles.Thedata wasextractedfromthe1974MotorTrendUSmagazine,andcomprises fuel consumption and 10 aspects of</td></tr>
</table>

	automobile design and performance for 32 automobiles(1973-74models).FormatA dataframewith32observationson11variables:[1]mpgMiles/(US)gallon, [2] cyl Number of cylinders [3] disp Displacement (cu.in.), [4] hp Gross horsepower [5] drat Rear axleratio,[6]wtWeight(lb/1000)[7]qsec1/4miletime,[8]vsV/S,[9]amTransmission(0=automatic,1=manual),[10]gear Numberofforwardgears,[11]carbNumberofcarburetors DevelopRprogram,tosolve thefollowing: a) Whatisthetotalnumberof observationsandvariablesinthedataset? b) Findthecarwiththelargest hpandtheleasthpusingsuitablefunctions c) Plothistogram/densityforeachvariableanddeterminewhethercontinuousvariablesarenormallydistri butedor not. Ifnot,whatistheirskewness? d) Whatistheaveragedifferenceofgrosshorsepower(hp) betweenautomobileswith3and4numberofcylinders(cyl)?Alsodeterminethedifferenceintheirstanda rddeviations. e) WhichpairofvariableshasthehighestPearsoncorrelation? **References(Weblinks):** 1. https://cran.r-project.org/web/packages/explore/vignettes/explore_mtcars.html 2. https://www.w3schools.com/r/r_stat_data_set.asp https://rpubs.com/BillB/217355
11	Demonstratetheprogressionofsalarywithyearsofexperienceusingasuitabledataset(Youcancreateyour own dataset). Plot the graph visualizing the best fit line on the plot of the given data points. Plot acurveofActualValuesvs. Predictedvaluestoshowtheircorrelationandperformanceofthemodel. Interpret the meaning of the slope and y-intercept of the line with respect to the given data. Implementusing lm function. Save the graphs and coefficients in files. Attach the predicted values of salaries as a newcolumntothe originaldatasetandsave the dataasanewCSV file. **SuggestedReading**–ReferenceBook2–Chapter20(GeneralConcepts,StatisticalInference,Prediction)

Course outcomes:

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

1. Explain the fundamental syntax of R data types, expressions and the usage of the R-Studio IDE
2. Develop a program in R with programming constructs: conditionals, looping and functions.
3. Apply the list and data frame structure of the R programming language.
4. Use visualization packages and file handlers for data analysis..

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

B.E COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)
Outcome Based Education (OBE) And Choice Based Credit System (CBCS)
SEMESTER – IV

UNIX & SHELL PROGRAMMING

Course Code	S4CCSA04	CIE Marks	50
TeachingHours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	1.5
Lecture Hours	-	Practical Hours	26hrs

Course objectives: The 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 number 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 thline , …… 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 executables. The build should be created using multiple .h and .c source file.

Course Outcomes:

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

CO1	Demonstrate the basic and advanced UNIX commands.
CO2	Apply changes in the file system using UNIX commands and their options.
CO3	Develop Shell Script using Shell programming constructs.
CO4	Demonstrate Software execution skills using the Linux environment.

Mapping of Course outcomes (COs) to Program outcomes (Pos):

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)