SCHEME OF TEACHING AND EXAMINATION (160 Credits Scheme)

(Effective from the academic year 2024-2025)

III Semester

B.E. in Artificial Intelligence & Data Science — 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	42	0	0	48	3	50	50	100	3
2.	IPCC	S3CCSI01	Operating Systems(I)	Dept.	42	0	28	50	3	50	50	100	4
3.	IPCC	S3CCSI02	Digital Circuits and Computer Organizations (I)	Dept.	42	0	28	50	3	50	50	100	4
4.	PCC	S3CCS01	Data Structures and Applications	Dept.	42	0	0	48	3	50	50	100	3
5.	PCCL	S3CCSL01	Data Structures and Applications Laboratory	Dept.	0	0	28	2	3	50	50	100	1
6.	ESC		ESC/ETC/PLC	Dept.	28	0	28	34	3	50	50	100	3
7.	UHV	SHS01	Social Connect and Responsibility **(Board: ME)**	Dept.	0	0	28	48	-	100	-	100	1
8.	AEC/ SEC		Ability Enhancement Course/ Skill Enhancement Course – III	Dept.	If offered as Theory Course				1½	50	50	100	1
					14	0	0	16					
					If offered as Integrated Course				1½				
					0	0	28						
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									
		SMC04	National Cadet Corps (NCC)	NCC CO									
			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	S3CCSA04	Parallel Programming
S3CCSA03	PHP Programming	S3CCSA05	Devops

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 & Data Science — 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.	42	0	0	48	3	50	50	100	3
2.	IPCC	S4CCSI03	Artificial Intelligence (I)	Dept.	42	0	28	50	3	50	50	100	4
3.	IPCC	S4CCSI04	Data Science with Python (I)	Dept.	42	0	28	50	3	50	50	100	4
4.	PCCL	S4CCSL01	Design and Analysis of Algorithms Lab	Dept.	0	0	28		3	50	50	100	1
5.	ESC		ESC/ETC/PLC	Dept.	42	0	0	48	3	50	50	100	3
6.	BSC	S4CCA01	Biology for Engineers **(Board: BT)**	BT, CH, Phy, Che	42	0	0	48	3	50	50	100	3
7.	UHV	SHS02	Universal Human Values Course **(Board: IEM)**	Dept.	14	0	0	16	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
					14	0	0	16					
					If offered as Integrated Course				1½				
					0	0	28	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									
		SMC04	National Cadet Corps (NCC)	NCC CO									
			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	S4CCSA04	Unix and Shell Programming
S4CCSA02	MERN Stack	S4CCSA05	Data Analytics with Excel & R Laboratory

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.

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
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	42hrs	Practical Hours	-

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

UNIT I **(8 Hours)**

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 II **(8 Hours)**

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 III **(8 Hours)**

Definition of Random Variable, Discrete Probability distribution, expectation, Variance, Binomial distribution, Poisson distribution. Continuous Probability distribution- expectation, Variance, Normal distribution and Exponential distributions.

UNIT IV **(9 Hours)**

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 V **(9 Hours)**

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: On Successful completion of this course, students 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 & Year**
Textbooks				
1	Higher Engineering Mathematics	B.S.Grewal	Khanna Publications	43rd edition 2015
2	Higher Engineering Mathematics	Ramana .B.V	Tata-McGraw Hill	latest edition 2016
Reference Books				
1	Advanced Engineering Mathematics	Erwin Kreyszig	Wiley Publications	10th Edition, 2015
2	Advanced Engineering Mathematics	C. Ray Wylie and Louis C. Barrett	Tata-McGraw Hill	6th Edition, 2005
3	Applied Mathematics for Engineers and Physicists	Louis A. Pipes and Lawrence R. Harvill	McGraw Hill	3rd edition 2014

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

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

Program Articulation Matrix:

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
	2	**2**												

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER - III

OPERATING SYSTEMS (I)

Course Code	S3CCSI01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	3
Lecture Hours	42Hrs	Practical Hours	28hrs

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

UNIT I **(8L+4P)**

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, Modules [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 II **(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 III **(9L+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 IV	**(9L+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 V	**(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	

Lab Syllabus: Implementation of programs on the following Operating System concepts:
1. Basic Unix commands : a. mkdir, rm, cd, ls, mv, cat, chmode. b. head, tail, sort, grep, sed, pipe
2. Threads
3. Process Scheduling.
4. Process Synchronization.
5. Deadlock Avoidance.
6. Memory allocation techniques.
7. Page Replacement Algorithms

Course Outcomes: On Successful completion of this course, students will be able to	
1	**Identify** the services, functions and structure of different operating systems.
2	**Apply and analyze** appropriate scheduling algorithm for process selection and execution.
3	**Identify and analyze** the techniques used to solve process synchronization issues
4	**Apply and analyze** various deadlock prevention, avoidance, detection and recovery mechanisms to solve real world problems
5	**Analyze** the performance of various memory management techniques and page replacement algorithms

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

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
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	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2	2									2		

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER - III

DIGITAL CIRCUITS AND COMPUTER ORGANIZATIONS (I)

Course Code	S3CCSI02	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	3
Lecture Hours	42Hrs	Practical Hours	28hrs

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

UNIT I **(8L+4P)**

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 II **(8L+6P)**

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. **Chapter7**: 7.1.

UNIT III **(9L+6P)**

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 IV **(9L+6P)**

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 V	**(8L+6P)**
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) PISO ii) SIPO iii) SISO iv) PIPO v) Ring Counter vi) Johnson counter **9.** Design and implement an Asynchronous Counter using Decade Counter IC to Count up from 0 to 9. Display the count value on 7 Segment LED display using BCD to 7 segment code converter IC. **10.** Design and implement a 3 stage Asynchronous Counter using a J-K Flip Flops to count from 0 to n.	

Course Outcomes: On Successful completion of this course, students will be able to	
1	**Illustrate** the basic operational concepts of a computer system and **discuss** its performance parameters
2	**Interpret** various addressing modes and **apply** the same to **design** solution to a given problem
3	**Discuss** basic processing unit to generate control signals and to **design** the control sequence for execution of an instruction
4	**Explain** the various arithmetic algorithms and **apply** the same to solve a given problem
5	**Describe** memory organization and **design** the solution to the given problem

Sl. No	Title of the Book	Name of the Author/s	Name of the publisher	Edition & Year
Textbooks				
1	Computer Organization	Carl Hamacher, Zvonko	TMH	2005
Reference Books				
1	Computer Organization & Architecture	William Stallings.	PHI	2006
2	Computer SystemsDesign and Architecture	Vincent P. Heuring & Harry F. Jordan	PEARSON	2004

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	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:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2	2									2		

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
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	Exam Hours	3
Lecture Hours	42hrs	Practical Hours	-

Course objectives: This course will enable students to:	
1	Discuss the concepts of structures, union, files and dynamic memory allocation techniques.
2	Describe the properties of various data structures such as Stacks, Queues, Lists, and Trees
3	Implement the data structures such as Stacks, Queues, Lists, and Trees using C language.
4	Discuss the applications of various Data Structures

UNIT I **(8 Hours)**

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 II **(9 Hours)**

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 III **(8 Hours)**

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 IV	**(8 Hours)**
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 V	**(9 Hours)**
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 & 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)

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	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	

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2	2										2	

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER - III

DATA STRUCTURES AND APPLICATIONS LABORATORY

Course Code	S3CCSL01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	3
Lecture Hours	-	Practical Hours	28hrs

Course objectives: This course will enable students to:	
1	Design and implement different data structures.
2	Develop C programs for various applications of data structures.
3	Select appropriate data structure for a given problem.

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

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

Open Ended Problems

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

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

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

Course outcomes:

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

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

Conduct of Practical Examination:

- All laboratory experiments are to be included for practical examination.
- Breakup of marks and the instructions printed on the cover page of answer script to be strictly adhered 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	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	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	

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2	2										2	

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

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:

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

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

UNIT-1 **(6 Hours)**

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

UNIT-2 **(6 Hours)**

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

UNIT-3 **(6 Hours)**

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 **(6 Hours)**

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 **(6 Hours)**

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

Activities:

Plantation and adoption of a tree: Select suitable species in consultation with horticulture, forest or agriculture department. Interact with NGO/Industry and community to plant Tag the plant for continuous monitoring

Heritage walk and crafts corner: Survey in the form of questioner by connecting to the people and asking. Questions during survey can be asked in local language but report language is English.

Organic farming: Collect data on organic farming in the vicinity. Like types of crop, methodology etc.,

Water Conservation: Report on traditional water conservation practices (to minimize wastage)

Food Walk: Survey local food centres and identify its specialty, Identify and study the food ingredients, Report on the regional foods, Report on Medicinals values of the local food grains, and plants.

PEDAGOGY

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

COURSE TOPICS:

The course will introduce social context and various players in the social space, and present approaches to discovering and understanding social needs. Social immersion and inspiring conversional will culminate in developing an actual, idea for problem-based intervention, based on an in-depth understanding of a key social problem.

A total of 26 hrs engagement per semester for this course in 3rd semester of the B.E. program. The students will be divided into 1 group of 60 each. Each group will be handled by one ffaculty mentor.

Guideline for Assessment Process:

Continuous Internal Evaluation (CIE)

Student shall keep a separate dairy and prepare report in consultation with the mentor/s to indicate what he has observed and learned in the social connect period.

Report shall be handwritten or blog with paintings, sketches, poster, video and/or photograph with Geo tag.

The report should be signed by the mentor.

The report shall be evaluated on the basis of the following criteria (see Table below) and/or other relevant criteria pertaining to the activity completed.

Each UNIT is evaluated for 35 Marks and final presentation will be for 15 marks.

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

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
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	28hrs	Practical Hours	28hrs

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

UNIT I **(6 Hours)**

An Overview of Java: Object-Oriented Programming, A First Simple Program, A Second Short Program Classes, Objects and Methods; Inheritance Classes, Objects and Methods Introduction, Defining a Class, Fields Declaration, Methods Declaration, Creating Objects, Accessing Class Members, Constructors, Methods Overloading, Static Members. Inheritance: Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called. Method Overriding, Dynamic Method Dispatch. Using Abstract Classes, Using final with Inheritance.

UNIT II **(6 Hours)**

Packages and Interfaces Packages: Defining a Package, Finding Packages and CLASSPATH, A Short Package Example, Access protection: An Access Example. Importing Packages. Interfaces: Defining an Interface, Implementing Interfaces, Nested Interfaces. Applying Interfaces, Variables in Interfaces. Default Interface Methods: A More Practical Example. Multiple Inheritance Issues, and Use Static Methods in an Interface.

UNIT III **(6 Hours)**

Exception Handling and Generics: Exception Handling Fundamentals, Exception Types, Uncaught Exceptions. Using try and catch: Displaying a Description of an Exception, Multiple Catch Clauses. throw, throws, finally. **What Are Generics?** A Simple Generics Example, The General Form of a Generic Class, Creating a Generic Method. Generic Constructors, Some Generic Restrictions.

UNIT IV **(5 Hours)**

Programming with I/O, Applets: I/O Basics, Streams, Byte Streams and Character Streams, The Predefined Streams. Reading Console Read the values, Reading characters, Reading Strings, Writing Console Output. The PrintWriter Class, Reading and Writing Files, Automatically Closing a File. **Applet Fundamentals:** Applet Basics. The Applet Class, The HTML APPLET Tag.

UNIT V	**(5 Hours)**
Event Handling, Introducing the AWT: Basic Event Handling, Working with Windows, Graphics and Text. **AWT Classes:** Window Fundamentals, Component, Container, Panel, Window, Frame, Working with Graphics: Drawing Lines, Rectangles etc, Working with Color, working with Fonts. Basic Java GUI using JavaFX: JavaFX application structure.	

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 program.
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 along with JavaFX.

Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the publisher**	**Edition & Year**
Textbooks				
1	Java The Complete Reference	Herbert Schildt	Tata McGraw Hill	9th Edition, 2014
Reference Books				
1	Object-Oriented Programming With JAVA Essentials and Applications	RajKumar Buyya, S Thamarai Selvi	Tata McGraw Hill	2009

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
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**	
Overall CO	**3**	**3**	**3**										**3**	

Program Articulation Matrix:

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
	3	**3**	**3**										**2**	

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	Write a program to Input Student Full name, College Id, Branch, year of admission, that can create a VCF file contact to be imported to a phone
3	Write a JAVA program demonstrating Method overloading and Constructor overloading.
4	Design a super class called Staff with details as Staff ID, Name, Phone, and Salary. Extend this class by writing three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a JAVA program to read and display at least 3 staff objects of all three categories.
5	Demonstrate dynamic dispatch using abstract class in JAVA.
6	Create two packages P1 and P2. In package P1, create class A, class B inherited from A, class C. In package P2, create class D inherited from class A in package P1 and class E. Demonstrate working of access modifiers (private, public, protected, default) in all these classes using JAVA.
7	Write a java program to perform simple command line calculator with an exception handler.
8	Stock Market: Read data from .csv file and calculate min(low), max(high), average for the year, month, week, between 2 dates, day, last 5 days, 20 days
9	Write a Java program that reads a text file and displays the contents on the screen.
10	Write a Java program to check whether the given element is present in a given array or not using generic method.

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
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	28hrs	Practical Hours	28hrs

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

UNIT I **(6L+4P)**

Fundamentals of WWW: A Brief Introduction to the Internet, The World Wide Web, Web Browsers, Web Servers, Uniform Resource Locators, 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: Version History, HTML and XHTML DTDs: The Specifications Up Close, (X)HTML Document Structure, Browsers and (X)HTML, The Rules of (X)HTML, **TextBook1:Chapter1**

UNIT II **(6L+6P)**

HTML5:

Hello HTML5, Loose Syntax Returns, XHTML5, HTML5: Embracing the Reality of Web Markup, Presentational Markup Removed and Redefined , HTML5 Document Structure Changes, Adding Semantics ,HTML5's Open Media Effort, Client-Side Graphics with <canvas>,HTML5 Form Changes, Emerging Elements and Attributes to Support Web Applications

TextBook1:Chapter2

UNIT III **(5L+6P)**

Cascading Style Sheets(CSS):

Introduction, CSS Overview , CSS Rules, Example with Type Selectors and the Universal Selector, CSS Syntax and Style, Class Selectors, ID Selectors, span and div Elements, Cascading, style Attribute, style Container, External CSS Files, CSS Properties, Color Properties, RGB Values for Color, Opacity Values for Color, HSL and HSLA Values for Color, Font Properties, line-height Property, Text Properties, Border Properties, Element Box, padding Property, margin Property, Case Study: Description of a Small City's Core Area.

Bootstrap: Introduction to Bootstrap, Why use Bootstrap, Bootstrap Examples-Tables, forms, nav menu, Breakpoints, Grids, carousel,cards.

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

UNIT IV	**(5L+6P)**
Module-4:Tables and CSS, Links and Images 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 with Table Values, a Element, Relative URLs, Navigation within a Web Page, CSS for Links , Responsive Images, Positioning Images, Shortcut Icon, iframe Element **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 V	**(6L+6P)**
Module-5:Introduction to JavaScript: Functions, DOM, Forms, and Event Handlers History of JavaScript, Hello World Web Page, Buttons, Functions, Variables, Identifiers, Assignment Statements and Objects, Document Object Model, Forms and How They're Processed: Client-Side versus Server-Side, form Element, Controls, Text Control, Accessing a Form's Control Values, reset and focus Methods, handling errors in Javacript. **TextBook2:8.2 to8,13,8.15, 8.16** Form Element, Controls, Text Control, Email Address Generator web page, Event Handler Attributes, on change, onmouseover, onmouseevent, Using no script to accommodate disabled Javascript (Chapter 8.11 to 8.18, 8.20)	

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 & Year**
Textbooks				
1	TextBook-1:HTML&CSS:The CompleteReference	ThomasA.Powell	TataMcGraw Hill	FifthEdition
2	WEBPROGRAMMING with HTML5,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	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	2		2										2	
CO2			2										2	
CO3			2										2	
CO4			2										2	
CO5			2										2	
Overall CO	2		2										2	

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2		2										2	

LAB COMPONENT

1. Design a HTML5 webpage for SIT HALCYON summer fest. The document must contain the following:

(i) Fest Heading (center of the page with color font)

(ii) Two paragraphs which describes about HALCYON

(iii) Create unordered list of 5 Halcyon events and ordered list of 3 cash prizes

(iv) Add a marquee "SIT Summer Fest 2025

(v) Insert a suitable background image and display the following output to register for Dance event

(i)

TO JOIN DANCE EVENT

Solve the following equation: $ax^2+b_ix^3+cxy+d_i^3$

Limited seats available!

Registration opens

~~First Deadline: 1st April 2025~~

Second Deadline: 30th April 2025

Event Date is :2nd May 2025

2. Create following tables using HTML5. Properly align cells, give suitable cell padding and cell spacing, and apply background color, bold and emphasis necessary

- Insert SIT college logo, followed by college name & department title
- Add two paragraphs about the college and department
- Design the time table of your class

Siddaganga Institute of Technology
Department of Computer Science & Engineering

The department was started in the year 1986, with an intake of 60 students offering a bachelors degree in Computer Science & Engineering, under Bangalore University. In these years of academic services, the department has grown up steadily, and at present the department hosts three autonomous programs, under the affiliation of Visveswaraya Technological University .The department has V.T.U recognized research centre, which was started in the year 2003.

The department has well experienced and highly qualified faculty members, who are specialized in some of the latest areas of Computer Science and Engineering, which includes, computer networks, wireless network security, cryptography, wireless sensor networks, cloud computing, bigdata,machine learning and so on. All faculties participate in various workshops and seminars to keep themselves updated with the latest technology. Faculty are also actively involved in some of the joint research projects with industries and research organizations and 13 patents are filed till date on the project works

3.Use HTML5 for performing following tasks: Insert suitable title, background colour, text colour for each div.

i. Draw a square using HTML5 SVG , fill the square with green color and make 6px brownstroke width
ii. Draw a rectangle using HTML5 SVG , fill the rectangle with blue color and make 8px redstroke width
iii. Draw a ellipse using HTML5 SVG , fill the ellipse with orange color and make 3px bluestroke width
iv. Draw a circle using HTML5 SVG , fill the circle with yellow color and make 4px greenstroke width
v. Draw a pentagon using HTML5 SVG , fill the pentagon with black color and make 4px greenstroke width
vi. Write the following mathematical expression by using HTML5

MathML $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

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

4. Design the registration form to validate the following fields of the Registration page. Apply suitable CSS properties.

- User Name (Name should contains alphabets and the length should not be less than 6 characters).
- Password (Password should not be less than 6 characters length).

- E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)
- Mobile Number (Phone number should contain 10 digits only).

url and Address (should not be Empty

5.Demonstrate the following HTML5 Semantic tags- <article>, <aside>, <details>, <figcaption>, <figure>, <footer>, <header>, <main>, <mark>, <section> for a webpage that gives information about your project details (projects completed, project undertaking etc.)

6. i)Create 3 paragraphs and apply the following CSS properties

Use Inline CSS

Para1: red color text using RGB format, underline the text , display the text in uppercase
Use Internal CSS
Para2: make the text bold using font-weight, font family as times new roman, line height:2.0
Use External CSS
Para3: make the font family as times new roman, line height:3.0, text color is blue

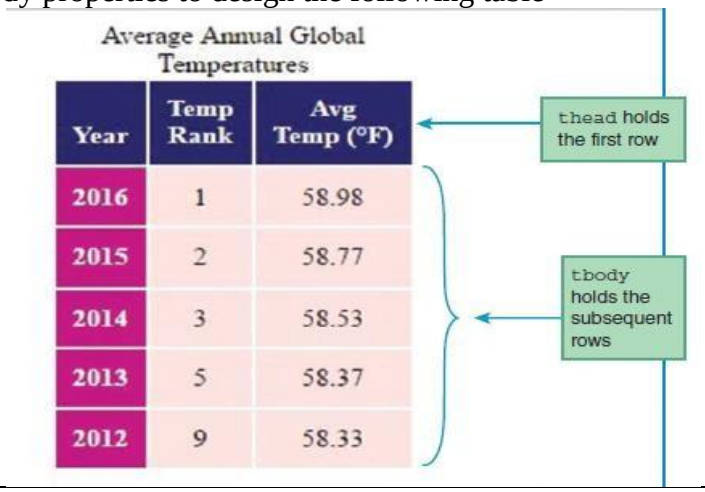
ii)Demonstrate the website design to interpret the CSS-Box model (use a division tag and include content, padding, border, and margin).

7. i)Use CSS Structural Pseudo-Class Selectors to design the following table

Noon Power Generation (positive) and Consumption (negative)

Component Description	Overall Size	Noon Power	Installed Cost
PV Solar Collectors	137 m^2 panel area	+18 kW	$45,000
Immediate Consumption	274 m^2 floor area	-5 kW	
Chilled Water Storage	2.3 m diameter x 2.1 m high	-2 kW	$1000
Battery Storage	1.3 m x 1.0 m x 1.1 m high, 1250kg	+18 kW	$6000

ii)use thead and tbody properties to design the following table



8.Develop and demonstrate JavaScript with POP-UP boxes and functions for the following problems:

Input: Click on Display Date button using onclick() function Output: Display date in the textbox

Input: A number n obtained using prompt Output: Factorial of n number using alert

Input: A number n obtained using prompt

Output: A multiplication table of numbers from 1 to 10 of n using alert

Input: A number n obtained using prompt and add another number using confirm

Output: Sum of the entire n numbers using alert

8. Create calculator interface with HTML and CSS

9. Create Calculator interface with HTML, CSS and javascript

10. Design a BMI calculator using HTML, CSS and Javascript. Inputs are Height and Weight.
11.Write an HTML page that contains a selection box with a list of 4 countries. When the user selects a country, its capital should be printed next in the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).
12. Create a webpage containing 3 overlapping images using HTML, CSS and JS. Further when the mouse is over any image, it should be on the top and fully displayed.

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
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	28hrs	Practical Hours	28hrs

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

UNIT I **(5L+5P)**

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 II **(6L+5P)**

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 III **(6L+6P)**

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

Reading and Writing Files: Files and File Paths, The os.path UNIT, The File Reading/Writing Process, Saving Variables with the shelve UNIT, Saving Variables with the

pprint.pformat() Function, Project: Generating Random Quiz Files,
Organizing Files: The shutil UNIT, Walking a Directory Tree, Compressing Files with the zipfile UNIT, Project: Renaming Files with American-Style Dates to European-Style Dates
Debugging: Raising Exceptions, Getting the Traceback as a String, Assertions, Logging, IDLE's Debugger.
Chapter7, Chapter9, Chapter10 & Chapter11 (Automate the Boring Stuff with Python by AlSweigart)

UNIT IV	**(6L+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 V	**(5L+6P)**
Web Scraping: Project: MAPIT.PY with the web browser UNIT, Downloading Files from the Web with the requests UNIT, Saving Downloaded Files to the Hard Drive, HTML, Parsing HTML with the BeautifulSoup UNIT **Working with Excel Spread sheets:** Excel Documents, Installing the open pyxl UNIT, Reading Excel Documents, Project: Reading Data from a Spread sheet, Writing Excel Documents **Working with CSV files and JSON data:** The csv UNIT, Project: Removing the Header from CSV Files, JSON and APIs, The json UNIT, Project: Fetching Current Weather Data Chapter12, Chapter13 & Chapter16 (Automate the Boring Stuff with Python by Al Sweigart)	

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

Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the publisher**	**Edition & 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:
1. Programs on basic concepts of python.
2. Programs on Strings
3. Programs on lists, tuples and dictionaries.
4. Programs on regular expressions.
5. Programs on exception handling.
6.Programs on files operations.
7.Programs on Classes and objects.
8.Programs on Web-Scrapping
9.Programs to work with CSV
10.Programs to work with JSON and other file formats

Conduct of Practical Examination:

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

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

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

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	3		3										2	

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE 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	28hrs	Practical Hours	28hrs

Course objectives: This course will enable students to:	
1	The course provides the basic principles of object-oriented programming using C++.
2	The course introduces the following topics such as classes, overloading, data abstraction, information hiding, encapsulation, inheritance, polymorphism, templates.
3	The course briefly covers C++ implementation and object-oriented considerations for software design and reuse

UNIT I **(6L+5P)**

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 II **(6L+6P)**

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 III **(5L+5P)**

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 IV	**(5L+6P)**
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 V	**(6L+6P)**
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 & Year**
Textbooks				
1	Object Oriented Programming with C++,	E Balagurusamy	Tata McGraw Hill	5th edition, ISBN: 9781259029936
Reference Books				
1	The Complete reference C++,	Herbert Schildt	Tata McGraw Hill	4th Edition
2	Object Oriented Programming with C++	Robert Lafore	SAMS Pearson Education	4th Edition
3	C++ Primer	Stanley B. Lippman,	Addison Wesley	4th edition, 2005
4	Object- Oriented Programming with C++	Sourav Sahay	Oxford University Press	1st edition, 2009

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	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	

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	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
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
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. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE 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	28hrs

Course objectives: This course will enable students to:	
1	To familiar with basic command of Git
2	To create and manage branches merge, branching.
3	To understand how to collaborate and work with Remote Repositories.
4	To familiar with version controlling commands

Sl.no	Experiments
1	**Setting Up and Basic Commands** Initialize a new Git repository in a directory. Create a new file and add it to the staging area and commit the changes with an appropriate commit message.
2	**Creating and Managing Branches** Create a new branch named “feature-branch.” Switch to the “master” branch. Merge the“feature-branch” into “master.”
3	**Creating and Managing Branches** Write the commands to stash your changes, switch branches, and then apply the stashed changes.
4	**Collaboration and Remote Repositories** Clone a remote Git repository to your local machine.
5	**Collaboration and Remote Repositories** Fetch the latest changes from a remote repository and rebase your local branch on to the updated remote branch.
6	**Collaboration and Remote Repositories** Write the command to merge “feature-branch” into “master” while providing a custom commit message for the merge.
7	**Git Tags and Releases** Write the command to create a lightweight Git tag named “v1.0” for a commit in your localrepository.
8	**Advanced Git Operations** Write the command to cherry-pick a range of commits from “source-branch” to the current branch.
9	**Analysing and Changing Git History** Given a commit ID, how would you use Git to view the details of that specific commit,including the author, date, and commit message?
10	**Analysing and Changing Git History** Write the command to list all commits made by the author “JohnDoe” between “2023-01-01”and “2023-12-31.”
11	**Analysing and Changing Git History** Write the command to display the last five commits in the repository’s history.
12	**Analysing and Changing Git History** Write the command to undo the changes introduced by the commit with the ID “abc123”.

Course outcomes:

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

1. Use the basics commands related to git repository
2. Create and manage the branches
3. Apply commands related to Collaboration and Remote Repositories
4. Use the commands related to Git Tags, Releases and advanced git operations
5. Analyse and change the git history

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

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

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2			2			2		2			2	

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE Outcome Based Education (OBE) and Choice Based Credit System (CBCS) **SEMESTER - III**			
PHP PROGRAMMING			
Course Code	S3CCSA03	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	28hrs

Course objectives: This course will enable students to:	
1	To introduce the PHP syntax, elements, and control structures
2	To make use of PHP Functions and File handling
3	To illustrate the concept of PHP arrays and OOPs

Sl. No.	**Experiments**
	AIM: Introduction to HTML/PHP environment, PHP Data Types, Variables, Literals, and operators
1	a. Develop a PHP program to calculate areas of Triangle and Rectangle. Develop a PHP program to calculate Compound Interest.
2	Demonstrating the various forms to concatenate multiple strings Develop program(s) to demonstrate concatenation of strings: (i) Strings represented with literals (single quote or double quote) (ii) Strings as variables (iii) Multiple strings represented with literals (single quote or double quote) and variables (iv) Strings and string variables containing single quotes as part string contents Strings containing HTML segments having elements with attributes
3	a. Develop a PHP Program(s) to check given number is: (i) Odd or even (ii) Divisible by a given number (N) (iii) Square of a another number Develop a PHP Program to compute the roots of a quadratic equation by accepting the coefficients. Print the appropriate messages.
4	a. Develop a PHP program to find the square root of a number by using the newton's algorithm. Develop a PHP program to generate Floyd's triangle.
5	a. Develop a PHP application that reads a list of numbers and calculates mean and standard deviation. 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 and 19, etc. The last "box" in the histogram should include scores between 90 and 100. Use a function to generate the histogram.
6	a. Develop PHP program to demonstrate the date() with different parameter options. Develop a PHP program to generate the Fibonacci series using a recursive function.
7	Develop a PHP program to accept the file and perform the following (i) Print the first N lines of a file Update/Add the content of a file
8	Develop a PHP program to read the content of the file and print the frequency of occurrence of the word accepted by the user in the file

9	Develop a PHP program to filter the elements of an array with key names.Sample Input Data: 1st array: ('c1' => 'Red', 'c2' => 'Green', 'c3' => 'White', c4 => 'Black') 2nd array: ('c2', 'c4') Output: Arra y([c1] => Red [c3] => White)
10	Develop a PHP program that illustrates the concept of classes and objects by reading and printing employee data, including Emp_Name, Emp_ID, Emp_Dept, Emp_Salary, and Emp_DOJ.
11	a. Develop a PHP program to count the occurrences of Aadhaar numbers present in a text. Develop a PHP program to find the occurrences of a given pattern and replace them with a text.
12	Develop a PHP program to read the contents of a HTML form and display the contents on a browser.
NOTE: Necessary HTML elements (and CSS) can be used for designing the experiments.	
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.	

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

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

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2											2	

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE Outcome Based Education (OBE) and Choice Based Credit System (CBCS) **SEMESTER - III**			
PARALLEL PROGRAMMING			
Course Code	S3CCSA04	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	28hrs

Course objectives: This course will enable students to:	
1	To program using Message Passing Paradigm
2	To program using shared address space
3	To program for GPUs using CUDA

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

Course outcomes:

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

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

Conduct of Practical Examination:

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

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	2												2	
CO2		2											2	
CO3	2	2											2	
Overall CO	2	2											2	

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2											2	

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE Outcome Based Education (OBE) and Choice Based Credit System (CBCS) **SEMESTER - III**			
DEVOPS			
Course Code	S3CCSA05	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	28hrs

Course objectives: This course will enable students to:	
1	Understand the principles and practices of DevOps and its role in software development.
2	Apply version control systems (like Git) to manage and track code changes effectively.
3	Implement continuous integration and continuous delivery (CI/CD) pipelines.
4	Use containerization tools (Docker) and orchestration tools (Kubernetes) effectively
5	Employ automation tools (like Ansible, Jenkins) for deployment and monitoring.

Sl.No	Hands on Sessions
1	**Introduction to DevOps and Linux:** DevOps Introduction: Software Development Life Cycles (SDLC), Agile Methodology, What is DevOps? Why DevOps? DevOps Importance, DevOps Model, DevOps Life cycle, Market Trend and Career Scope for DevOps, DevOps Tools. Introduction of Linux Operation System, Installing Pre-requisite Software's (SSH Tools and FTP Tools) in Desktop/Laptop
2	**Version Control with Git:** Account Creation in GitHub/GitLab/BitBucket, What is git? What is the VCS? What is SCM? What is Branch? What is Tag? Git Administration. Git commands, Working with git as a Developer perspective, SSH Key generation, PAT creation, Cloning Repositories, Merging Branches, Pull Requests, Cloning Remote Repo, Forking Repo, Branching strategy, Best practices for Releases/Code commits in any VCS. **Collaboration activity of a Repository in Github through Git**
3	Integration activity of a Repository in Github through Git.
4	**Advanced Git Operations** : Write the commands to cherry-pic a range of commits from "source-branch" to the current branch.
5	**Continuous Integration and Continuous Deployment (CI/CD):** CI/CD Concepts: An in-depth look at CI/CD principles and their importance. Jenkins: Setting up and configuring Jenkins for automated builds and deployments. Pipeline as Code: Creating and managing CI/CD pipelines with code **Setting up Jenkins and Creating Freestyle project with Git**
6	**Containerization with Docker:** Introduction to Containers: Understanding containerization and its benefits. Docker Essentials: Learning Docker fundamentals, containers, and images. Docker Orchestration: Exploring container orchestration. **Installing Docker Desktop and Creating Dockerhub account**
7	Creating a Docker image for app
8	**Kubernetes:** Kubernetes Introduction, Architecture, Kubernetes Cluster (Self-Managed) Setup Using Kubeadm. Kubernetes Namespace, Kubernetes Objects, POD Replication Controller, Replica Set, Daemon Set, Deployment Set, Rolling Update, Recreate, Stateful Set, Service. **Creating Pods & replicaset for the app in AWS (Kubernetes)**
9	**Open ended project**

Course Outcomes: On Successful completion of this course, students will be able to	
1	Understand the principles and practices of DevOps and its role in software development.
2	Apply version control systems (like Git) to manage and track code changes effectively.
3	Implement continuous integration and continuous delivery (CI/CD) pipelines.
4	Use containerization tools (Docker) and orchestration tools (Kubernetes) effectively
5	Employ automation tools (like Ansible, Jenkins) for deployment and monitoring.

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
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	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
		2						**2**		**2**			**2**	

IV SEMESTER

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER - IV

DESIGN AND ANALYSIS OF ALGORITHM

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	42hrs	Practical Hours	-

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

UNIT I **(8 Hours)**

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 II **(8 Hours)**

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 III **(8 Hours)**

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 IV **(9 Hours)**

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 V **(9 Hours)**

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 & Year
Textbooks				
1	Introduction to The Design & Analysis of Algorithms.	AnanyLevitin	Pearson Education	Ed2 2007. ISBN: 81-7808-984-X
Reference Books				
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	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	**2**												**2**	
CO2			**2**										**2**	
CO3			**2**										**2**	
CO4		**2**											**2**	
Overall CO	**2**	**2**	**2**										**2**	

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	**2**	**2**										**2**	

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER - IV

ARTIFICIAL INTELLIGENCE (I)

Course Code	S4CCSI03	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	3
Lecture Hours	42hrs	Practical Hours	28hrs

Course objectives: This 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 I **(9L+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 II **(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 III **(9L+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 IV	**(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 V	**(8L+4P 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 3rd edition)	

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 & 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											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1		2										2		
CO2		2										2		
CO3			2									2		
CO4			2									2		
CO5		1										1		
Overall CO		2	2									2		

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
		2	2									2		

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER - IV

DATA SCIENCE WITH PYTHON(I)

Course Code	S4CCSI04	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:2)	SEE Marks	50
Credits	4	Exam Hours	3
Lecture Hours	42hrs	Practical Hours	28hrs

Course objectives: This 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.
1.	Describe the concepts of association rule mining and ensemble modeling

UNIT I **(8 Hours)**

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 II **(8 Hours)**

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 III **(8 Hours)**

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 IV	**(9 Hours)**
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 V	**(9 Hours)**
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: On Successful 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 & 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 implement Agglomerative clustering algorithm. Print the dendrogram..
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											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
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											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2	2											2

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

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

SEMESTER - IV

DESIGN AND ANALYSIS OF ALORITHMS LAB

Course Code	S4CCSL01	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	50
Credits	1	Exam Hours	3
Lecture Hours	-	Practical Hours	28hrs

Course objectives: This course will enable students to:	
1	An introduction to the design and analysis of algorithms. (Synthesis)
2	Expose students to prove the correctness and analyse the running time of the basic algorithms. (Analysis)
3	To compare the running time of sorting and searching algorithms. (Comprehension)
4	Create an awareness of applying the algorithms and design techniques to solve problems. (application)
Sl.no.	**Experiments**
	Note: C/C++ language must be used to develop the following programs:
1	Sort a given set of elements using the Merge sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
2	Sort a given set of elements using the Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
3	Print all the nodes reachable from a given starting node in a given digraph using Depth First Search method.
4	Print all the nodes reachable from a given starting node in a digraph using Breadth First Search method.
5	Obtain the Topological ordering of vertices in a given digraph using source removal method.
6	Sort a given set of elements using the Heap sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.
7	Implement Horspool algorithm for String Matching.
8	Implement 0/1 Knapsack problem using memory functions.
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 Prim's algorithm.
12	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm
13	Implement N Queens problem using Backtracking.

14	Implement Assignment 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 by the examiners.
- Students can pick one experiment from the questions prepared by the examiners.
- Change of experiment is allowed only once and 20% Marks is to be deducted.

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	2		2										2	
CO2	2	2	2										2	
CO3	2		2										2	
Overall CO	2	2	2										2	

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2	2										2	

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	42hrs	Practical Hours	-

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

Teaching-Learning Process (General Instructions)

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

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

UNIT I **(8 Hours)**

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 II **(8 Hours)**

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 III **(8 Hours)**

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 IV **(9 Hours)**

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 V **(9 Hours)**

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.

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**

CO1	**2**	**2**												**3**
CO2	**2**	**2**	**3**											**3**
CO3	**2**	**2**	**3**											**3**
CO4	**2**	**2**												**3**
CO5	**2**	**2**				**2**	**2**							**3**
Overall CO	**2**	**2**	**3**			**2**	**2**							**3**

Program Articulation Matrix:

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
	2	**2**	**3**			**2**	**2**							**3**

SEMESTER - IV			
DISCRETE MATHEMATICAL STRUCTURES			
Course Code	S4CCS02	CIE Marks	50
Teaching Hours/Week (L:T:P)	(3:0:0)	SEE Marks	50
Credits	3	Exam Hours	03
Lecture Hours	42hrs	Practical Hours	-

Course objectives: This 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 **(8 Hours)**
Relations, Properties of Relations, Computer Recognition- Zero-One Matrices and Digraphs, Partial order relation -Poset and Hasse-Diagrams, Equivalence Relation and Partitions, Extremal elements of a Poset, Lattice.

UNIT II **(8 Hours)**
Binary Operations and Properties, Definition of a Group, Examples and Elementary properties, Abelian Group, Homomorphism, Isomorphism and Cyclic Groups, Cosets and Lagrange's Thoerem, Normal subgroups.

UNIT III **(8 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 **(9 Hours)**
Definitions and Examples, Subgraphs, Complements, and Graph Isomorphism, Vertex Degree, Euler Trails and Circuits, Planar Graphs, Hamilton Paths and Cycles.

UNIT V **(9 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: On Successful completion of this course, students will be able to	
1	Compute zero-one matrix, composition of relations and draw Hasse diagram. (L3).
2	Apply the concept of groups and subgroup to verify Lagrange's theorem.. (L2).
3	Apply the theory of Boolean algebra to minimize switching functions. (L3).
4	Recognize types of graphs, outline properties of graphs, understand isomorphism and apply Graph theory tools in solving real world problems. (L2/ L3).
5	Color 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 & Year**

Textbooks				
1	Discrete & 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

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

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

Program Articulation Matrix:

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
	3	1												

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	42hrs	Practical Hours	-

UNIT I	**(8 Hours)**
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 II	**(8 Hours)**
ntroduction 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,	

UNIT III	**(9 Hours)**
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 IV	**(9 Hours)**
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 V	**(9 Hours)**
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.	**Title of the Book**	**Name of the**	**Name of the**	**Edition &**

No		Author/s	publisher	Year
Textbooks				
1	Discrete and Combinatorial Mathematics	Ralph P. Grimaldi	Pearson Education,	5th edition 2012
2	Applied Combinatorics	Alan Tucker	Wiley-India	5th edition 2011
Reference Books				
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 Articulation Matrix (CO-PO and CO_PSO MAPPING)

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

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	3	**1**												

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
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	42hrs	Practical Hours	-

UNIT I **(8 Hours)**

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 II **(8 Hours)**

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 III **(8 Hours)**

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

UNIT IV **(9 Hours)**

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

UNIT V **(9 Hours)**

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 & Year
Textbooks				
1	Linear algebra and its applications	David C. Lay	Pearson Education,	5th edition 2014
2				
Reference Books				
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 Articulation Matrix (CO-PO and CO_PSO MAPPING)

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

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	3	1												

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
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	42hrs	Practical Hours	-

Course objectives: This course will enable students to:	
1	To develop an ability to use algorithms for approximation problems w.r.t Differentiation and Integration
2	To develop an ability to use algorithms for approximation problems w.r.t Differential equations
3	To develop an ability to use algorithms for approximation problems w.r.t partial differential equations
4	To develop an ability to use algorithms for approximation problems w.r.t Linear Algebraic equations
5	To develop an ability to use algorithms for approximation problems w.r.t Finite element methods.

UNIT I **(8 Hours)**

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 II **(8 Hours)**

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 III **(9 Hours)**

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 IV **(8 Hours)**

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

UNIT V **(9 Hours)**

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 & Year**
Textbooks				
1	Numerical Methods	Balagurusamy,E.,	Tata McGraw –Hill	978-0074633113 Standard Edition, July 2017
Reference Books				
1	Numerical Analysis and Algorithms	Niyogi, Pradip	Tata McGraw –Hill	978-0070494930 2003

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1	**2**				**2**									
CO2	**2**				**2**									
CO3	**2**				**2**									
CO4	**2**				**2**									
CO5	**2**				**2**									
CO6	**2**				**2**									
Overall CO	**2**				**2**									

Program Articulation Matrix:

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
	2				**2**									

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
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	14hrs	Practical Hours	-

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

UNIT I **(3 Hours)**

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 II **(3 Hours)**

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 III **(3 Hours)**

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 IV **(2 Hours)**

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 V **(3 Hours)**

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 this course, 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

Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the publisher**	**Edition & Year**
Textbooks				
1	'Foundation Course in Human Values & Professional Ethics; Presenting a universal approach to value education through self -exploration'.	Gaur, R.R. & Sangal R	Excel Books	2016, ISBN: 978-8-174-46781-2
Reference Books				
1	'Human Values'	Tripathi A.N.	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**
CO1						1		3			
CO2						1		3			
CO3						1		3			
Overall level						**1**		**3**			

Program Articulation Matrix:

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
	2	**2**	**3**			**2**	**2**							**3**

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER - IV

MERN STACK

Course Code	S4CCSA02	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	28hrs

Course objectives: This course will enable students to:	
1.	To program using Message Passing Paradigm
2.	To program using shared address space
3.	To program for GPUs using CUDA

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

Course Outcomes: On Successful completion of this course, students will be able to	
1	To implement and debug program using Message Passing Interface (MPI)
2	To implement and debug program using OpenMP to use shared address space
3	To implement and debug programs on GPU

Conduct of Practical Examination:

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

Change of experiment is allowed only once and 20% Marks is to be deducted.

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1	2												2	
CO2	2	2											2	
CO3	2												2	
Overall CO	2	2											2	

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
	2	2											2	

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE Outcome Based Education (OBE) and Choice Based Credit System (CBCS) SEMESTER – IV			
UNIX & SHELL PROGRAMMING			
Course Code	S4CCSA04	CIE Marks	50
Teaching Hours/Week (L:T:P)	(0:0:2)	SEE Marks	100
Credits	1	Exam Hours	1.5
Lecture Hour	-	Practical Hours	28 Hrs

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

Experiments:

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

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

	- Checks every argument specified is a file or a directory and report accordingly. Whenever the argument is a file, the number of lines on it is also reported. -
9.	Develop a shell scripts using functions to perform the following : - To check the given string is palindrome or not. Display the input string, reversed string and the result with suitable messages. - To find the substring in a given string. Input to the shell script must be accepted from the user and display the resultant string(s) along with input string(s) with suitable messages.
10.	Develop a shell script to check the permission of a file, print file line contents along with line numbers and copy the contents of files to another file.
11.	Develop a make file to build executable. The build should be created using multiple .h and multiple .c source files

Course Outcomes:	
Upon completion of this course the student will be able to:	
CO1	To execute basic unix commands using bash shell
CO2	To program using shell scripting
CO3	To program in C to utilize unix services using filters, process and files

Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
CO1			**2**										**2**	
CO2			**2**										**2**	
CO3			**2**										**2**	
Overall CO			**2**										**2**	

Program Articulation Matrix:

Course Outcomes	**Program Outcomes**											**PSOs**		
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
			2										**2**	

B.E. ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
Outcome Based Education (OBE) and Choice Based Credit System (CBCS)
SEMESTER - IV

DATA ANALYTICS WITH EXCEL AND R LABORATORY

Course Code	S4CCSA05	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	28Hrs

Course objectives: This course will enable students to:	
1	**Build** Spreadsheet application by creating structured data
2	**Automate** the problem solutions using Excel Formulas and advanced features
3	**Explore** Data Analytics features and Apply analytics using R

1. Create a spreadsheet for 100 students records which consist of following attributes Student USN, Student Name, course taken , Marks Obtained, Grade points

   a. **Create a suitable examination database and find the sum of the marks(total) of each Student and Average**

   b. Obtain **Letter grades, grade points and corresponding marks range as per below table:**

Level	Out-standing	Excellent	Very Good	Good	Above Average	Average	Pass	Fail
Letter Grades	O	A+	A	B+	B	C	P	F
Grade Points	10	9	8	7	6	5	4	0
Absolute Marks Range (%)	≥90	<90 - ≥80	<80 - ≥70	<70 - ≥60	<60 - ≥55	<55 - ≥50	<50 - ≥40	<40
	(90-100)	(80-89)	(70-79)	(60-69)	(55-59)	(50-54)	(40-49)	(0-39)

NE: If a student fails to satisfy attendance (85%) and/or CIE (40%) requirements for the course/s then such course/s shall be marked as NE i.e., Not Eligible to appear for SEE in that course/s.

F' Grade is awarded under the following conditions:

- Failing to secure 35% of marks in SEE (min. 35/100)

c. **Calculate CGPA and SGPA** as per below rule

The academic performance of a student is indicated by two different indices, Semester Grade Point Average (SGPA) and Cumulative Grade Point Average (CGPA).

- SGPA is an indication of the performance of the student in the current semester.

SGPA is calculated as indicated below.

$$\text{SGPA} = \frac{\Sigma\ [(\text{Course credits}) \times (\text{grade points})] \text{ for all courses with letter grades (in that semester)}}{\Sigma\ [(\text{Course credits})] \text{ for all courses registered in that semester including F grades and excluding W and DP courses.}}$$

- CGPA is an indication of the cumulative performance of the student from the first semester up to the current semester.

CGPA is calculated as indicated below.

$$\text{CGPA} = \frac{\Sigma\ [(\text{Course credits}) \times (\text{grade points})] \text{ for all the courses whose letter grades are E and above from the I semester till the current semester.}}{\Sigma\ [(\text{Course credits})] \text{ whose letter grades are E and above from the I semester till the current semester.}}$$

Calculate SGPA and CGPA for 1st semester, IInd Semester, Supplementary Semester, and for the entire course.

d. Calculate Percentage equivalence of CGPA. Apply the formula Percentage of Marks = (CGPA - 0.75)x10

e. Calculate CGPA range and corresponding class as per below table

CGPA	Class
≥5.00 & < 6.75	Second
≥6.75 & < 7.75	First
≥7.75	Distinction

Visualization:

a. Freeze rows and columns
b. Sort the data according to USN.
c. Remove duplicates.
d. Filter: Distinction, First and Second-class data
e. Format the data using different table styles, conditional formatting including colours, labels, or data bars etc.
f. Visualize the data using Charts and Perform result analysis using Grade against Each Subject. Find cases where majority of total value attributed to failure, second class, First class and distinction.
g. Assuming that F grade students are passed in the exam in subsequent semester attempt and modify the data accordingly. Visualize the same over time series of data and do outlier.
h. Apply the following formula suitably and analyse the outcome.
    - IS FUNCTIONS
    - CONDITIONAL FUNCTIONS
    - MATHEMATICAL FUNCTIONS

Additional – Identify and Learn the short cut key for using above Excel features

2. Create Sales Database (big data – min 100 records) to predict the sales of product against quantity sold, Net Value, Gross Value etc. Sample is shown below (100 rows, 32 columns).

TXN Type	TXN No	TXN Date	Year	Month	Month Name	Day	Day Name	Qtr	FY Qtr	Location Name	Salesman	Salesman Name	Customer	Customer Short Name	Cust Type Name	Cust City	Cust City Nam
INV	173914	02-01-2023	2023	1	January	2	Wednesday	Qtr 1	FY Qtr 3	Warehouse	TNS92	Ram	HA2161	Raj	Modern Trade	TMK	Tumakuru

Item Name	Item Short Name	Item Brand Name	Item Type Name	Item Make	Item Make Name	Item Identification	Grade 1	Quar
MQ 100 - Electro Appliances HAND BLENDER - 22111020	MQ 100 - Electro Appliances HAND BLENDER -	Electro Appliances	Handheld Appliances	GER	Germany	Electro Appliances	DSF19	

Discount	Gross Value	Net Value
0	13600	13600

i. Apply the following formula suitably and analyse the outcome.
    - HLOOKUP, VLOOKUP
    - FIND, SEARCH
    - LEFT, RIGHT, MID, TRIM, CONCANTENATION
    - COUNT, COUNTA, COUNTBLANK
    - DATE and TIME

ii. Apply Conditional Formatting including labels, colors, data bars etc.

a. Calculate How much Sales for Item # (# can be any specific item in datasheet)
b. Calculate Total Sales in any specific location
c. Calculate Total Sales during week and weekends
d. Apply conditional formatting for quarter wise sales data per salesperson. Note: Sample is shown below

Salesman Name	Qtr 1	Qtr 2	Qtr 3	Qtr 4
Ram	2,37,389	2,97,355	1,91,538	1,46,848
Jyothi	1,96,959	4,33,803	1,52,971	3,64,452
Sree Rav	85,892	1,83,200	3,51,871	2,48,707
Jaya	58,688	1,11,398	17,209	1,01,830

e. Apply conditional formatting for analysing change in sale. Note: Sample is shown below

Salesman Name	Qtr 1	Qtr 2	1st Half Sales	Qtr 3	Qtr 4	2nd Half Sales	Change
Ram	2,37,389	2,97,355	5,34,744	1,91,538	1,46,848	3,38,386	-37%
Jyothi	1,96,959	4,33,803	6,30,762	1,52,971	3,64,452	5,17,422	-18%
Sree Rav	85,892	1,83,200	2,69,092	3,51,871	2,48,707	6,00,578	123%
Jaya	58,688	1,11,398	1,70,087	17,209	1,01,830	1,19,038	-30%

f. Represent the Sales gain (e.g., Red bar for -ve, Blue bar for +ve) through data bar. Note: Sample is given below

Invoice Date	Sales	Bar
12-02-2019	(12,456)	(12,456)
08-04-2019	47,335	47,335
25-09-2019	84,133	84,133

iii. Analyze the data by visualization (apply charts) for Average Price per Month. (Note: Obtain Average price by computing the ratio of net value over quantity)

Analytics and Visualization :

a) Format existing data table into pivot table

b) Summarize City and Sales. Sample is shown below.

a.

Cust City Name	Sum of Quantity	Sum of Net Value
Bangalore	15,348	23,27,133

c) Calculate the City and Sales Contribution. Sample is shown below.

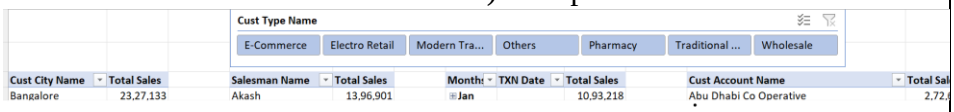
a.

Cust City Name	Total Qty	Total Sales	WOB
Bangalore	15,348	23,27,133	17.29%

d) Apply the slicers

e) Create multiple pivot table with reference to Customer Type, Salesman, Month wise sales transactions and Customer Account Name(Insert extra column by name Customer Account Name to the dataset). Sample is shown below.

a.



f) Outline form - Filter, Sort, Group and Apply advanced conditional formatting to the pivot table

g) Create Pivot Report - Analyse and Visualize sales

h) Create Pivot table with external data – show in tabular form

Use Slicers, auto filters etc

3. Create worksheet with following fields: Empno, Ename, Basic Pay(BP), Travelling Allowance(TA), Dearness Allowance(DA), House Rent Allowance(HRA), Income Tax(IT), Provident Fund(PF), Net Pay(NP).

Apply appropriate Excel formulas to,

a) Calculate DA , where DA = 50% of basic

Analyze and Visualize, total DA obtained in FY. Apply the conditional formatting for highest DA paid for an employee.

b) Calculate Total HRA received in a given FY

Analyze,

How much HRA can be claimed?

The HRA exemption is calculated as the minimum of the following three amounts:

i) Actual HRA received from the employer

ii) Rent Paid – 10% of Basic Salary

Visualize HRA claims of employee. Apply conditional formatting for minimum and maximum claims.

c) Calculate Income Tax

The tax rate for a salaried individual (under 60 years) as per the old regime is as follows:-

The tax rates for FY 2023-24, as per the new tax regime is, as follows:-

Tax Slab	Tax Rate
Upto Rs 3,00,000	Nil
Rs. 3,00,001-6,00,000	5%
Rs. 6,00,001-9,00,000	10%
Rs. 9,00,001-12,00,000	15%
Rs. 12,00,01-15,00,000	20%
Above Rs. 15,00,000	30%

Analyze how to save Income tax for a salaried individual?

To reduce taxes, it is highly important to understand the salary structure:-

- Taxable Salary Income = Salary (-) Exemptions (Exempt allowances + Standard deduction +Professional Tax)
- Net taxable income= Taxable Salary Income (-) Deductions under chapter VIA

Visualize the deductions and apply the conditional formatting appropriately

d) Calculate Provident Fund

Assume, Employees' portion of the EPF = 12% * (Basic Salary + DA)

Analyze and Visualize,

i) Employee contribution and Employer Contribution for Basic + DA <= Rs. 12000

ii) Employee contribution and Employer Contribution for Basic + DA >= Rs. 12000

Sample is shown below:

Employee Provident Fund Calculator (EPF)

Employee Name:			
Rate of increase of Basic Pay:	5.00%	Interest Rate	8.65%
Your Contribution:	12.00%	CTC	30000
Company's Contribution:	3.67%	Basic Pay + DA:	12000

Year	Opening Balance	Basic Pay	Employee Yearly Contribution	Company Yearly Contribution	Total Contribution	Rate of Interest	Closing Balance
1	0.00	12000.00	17280.00	5283.76	22563.76	8.65%	24515.53
2	24515.53	12600.00	18144.00	5548.00	23692.00	8.65%	52377.48

e) Calculate Net pay.

Net Salary: Gross Salary – Deductions

Apply conditional formatting for highest net salary and lowest net salary and visualize the data.

4. i) Animate the charts using VBA macro

Consider Sales details for the year 2024 and 2025.

Develop animated Bar chart with data range of Month on X – Axis and Sales on Y – Axis correspondingly. The chart should contain title, label, and legends.

ii) Automated Gantt Chart and Cost Estimation Generator using VBA in Excel.

The objective is to automate the creation of a Gantt chart and total project budget estimation from a task list using a VBA macro. The macro should dynamically create a timeline, highlight task durations in a Gantt-style format, and calculate the overall project cost. Analyze the timeline and cost obtained

Open Ended Project – Students are informed to download xlsx or csv files with large dataset (both structured and unstructured data) from Kaggle or GitHub or any online repositories. Import the files and analyze the data using Excel formulas and VBA macros. Use Slicers, Auto Filter, Conditional Formatting and Excel Data Analytics.

Data Analytics using R

5 Assess the Financial Statement of an Organization being supplied with 2 vectors of

	data: Monthly Revenue and Monthly Expenses for the Financial Year. You can create your own sample data vector for this experiment) Calculate the following financial metrics: a. Profit for each month. b. Profit after tax for each month (Tax Rate is 30%). c. Profit margin for each month equals to profit after tax divided by revenue. d. Good Months – where the profit after tax was greater than the mean for the year. e. Bad Months – where the profit after tax was less than the mean for the year. f. The best month – where the profit after tax was max for the year. g. The worst month – where the profit after tax was min for the year. Note: a. All Results need to be presented as vectors b. Results for Dollar values need to be calculated with $0.01 precision, but need to be presented in Units of $1000 (i.e 1k) with no decimal points c. Results for the profit margin ratio need to be presented in units of % with no decimal point. d. It is okay for tax to be negative for any given month (deferred tax asset) e. Generate CSV file for the data
6	**Employee Productivity** Create a data frame with employee ID, name, department, age, and productivity score. a) Subset employees from the Sales department over 40 with a productivity score above 80. b) Merge with a second data frame containing employee ID and weekly working hours. c) Find correlation between age and productivity (Pearson/Spearman). Plot and log-transform the data.
7	**Student Performance in Analytics** Create a data frame with student ID, name, course marks, grade, and project score. a) Filter students with project score > 85 and grade 'A'. b) Merge with attendance data using student ID. c) Identify top predictors of course marks and handle multi collinearity.
8	**Online Course Platform** Create a data frame with course ID, title, enrollment, rating, and instructor. a) Filter courses with rating > 4.5 and enrollments > 1000. b) Merge with course ID and completion rate. c) Visualize relationships, remove correlated variables like rating and completion rate.
9	**Smart City Traffic Monitoring** Create a data frame with junction ID, vehicle count, average speed, and accident rate. a) Subset records with high traffic and low speed. b) Merge with road type using junction ID. c) Find important features for accident prediction and handle collinearity.

Course Outcomes: Upon completion of this course the student will be able to:	
1	Use advanced functions and productivity tools to assist in developing worksheets
2	Manipulate data lists using Outline and PivotTables.
3	Use Consolidation to summarize and report results using R programming language.
4	Identify and Solve big data related problems using Excel Analytics and R

Course Articulation Matrix (CO-PO and CO_PSO MAPPING)

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
CO1		2												2
CO2		2												2
CO3		2												2
CO4		2												2
Overall CO		**2**												**2**

Program Articulation Matrix:

Course Outcomes	Program Outcomes											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
		2												**2**