III

SEMESTER

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS308** Course Title: **Logic Design and Computer Organization**
2. Contact Hours: **L:** 3 **T**: 0 **P**: 0
3. Examination Duration (Hrs): **Theory** 3 **Practical** 0
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSC
8. Pre-requisite: Basic Electronics Engineering (TEC101/ TEC201)

9. **Course Outcome:**	After completion of the course the students will be able to**:** CO1: Understand the process of minimizing Boolean function and obtaining the combinational logic circuits from Boolean functions. CO2: Analyse the basic storage elements in digital circuits and develop sequential circuits by applying them. CO3: Evaluate the design of different types of register, counter, and programmable logic devices. CO4: Apply the concept of digital logic circuits in computer organization & architecture and evaluate the computer performance. CO5: Create the arithmetic logic used in computer and describe the machine instruction execution. CO6: Understand the memory hierarchy of computer and how different I/O devices interact with the processing unit.

10. **Details of the Course:**

Sl. No.	Contents	Contact Hours
1	**Unit 1:** Simplification of Boolean Function using K-map method (upto 5 variables) and Quine-Mc Clusky method. Nand and Nor Implementation. Combinational Logic: Introduction, Analysis & Design Procedure, Binary Adder & Subtractor, Decimal Adder, Binary Multiplier, Magnitude Comparator, Decoders, Encoders, Multiplexers, De-MJultiplexers, code conversion. Introduction to HDL description of combinational logic circuits	10

2	**Unit 2**: Sequential Logic: Introduction, Types of Sequential circuits, Basic storage elements (Latch and Flip-flops), Characteristic equations & tables, excitation table, Flip-flop conversion, Register, Universal Shift register.	8
3	**Unit 3:** Counters: Ripple & Synchronous binary counters, BCD counter, mod-n counter, ring & Johnson counter. Mealy and Moore model. Design and analysis of synchronous sequential circuit. State diagram and state table.	10
4	**Unit 4:** Introduction to Computer Organization & Architecture, Von Neumann and Harvard Architecture, RISC and CISC machines, Basic measures of computer performance, Amdahl's Law, Little's Law. Computer Arithmetic (Integer and Floating Point): Representation, Addition, Subtraction, Multiplication and Division. Processor structure and operation, Instruction Cycle, Basic concept of pipelining: Arithmetic, instruction, and processor level. Control unit operation and microprogrammed control.	10
5	**Unit 5:** Basic concept of Memory, Memory Addressability: Big-endian and Little-endian method, Memory hierarchy: Locality and performance, Cache memory: Principles and elements of design (Replacement algorithm, Effective Access Time, Hit Rate, and Miss penalty), I/O interface: External devices, I/O modules, Programmed I/O, Interrupt driven I/O, Direct Memory Access, Bus arbitration. Introduction to alternative architectures (Flynn Taxonomy).	10
	Total	**48**

Text Books:

Authors Name	**Title**	**Edition**	**Publisher, Country**	**Year**
M. Morris Mano	Digital Logic and Computer Design	1st Edition	Pearson	2016
W. Stalling	Computer Organization and Architecture,	11th Edition	Pearson	2022

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Charles H. Roth Jr., Larry L. Kinney	Fundamentals of Logic Design,	7th Edition	CL Engineering, India	2015
John P Hayes	Computer Architecture and Organization,	3rd Edition	McGraw Hill	2017

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS308.1	Understand the process of minimizing Boolean function and obtaining the combinational logic circuits from Boolean functions.	1	3	-	-	-	-	-	-	1	1	-	1	2	-	-
TCS308.2	Analyze the basic storage elements in digital circuits and develop sequential circuits by applying them.	1	2	3	-	-	-	1	1	2	1	-	1	2	2	1
TCS308.3	Evaluate the design of different types of register, counter, and programmable logic devices.	1	2	2	3	3	1	-	1	1	2	1	2	3	2	2
TCS308.4	Apply the concept of digital logic circuits in computer organization & architecture and evaluate the computer performance.	1	2	2	3	-	-	1	1	1	1	1	1	3	2	1
TCS308.5	Create the arithmetic logic used in computer and describe the machine instruction execution.	1	2	3	-	-	1	-	1	1	1	1	1	2	2	2
TCS308.6	Understand the memory hierarchy of computer and how different I/O devices interact with the processing unit.	1	2	-	-	-	-	1	1	1	1	-	1	2	2	-
TCS 308		1	2.2	2.66	3	-	1	-	1	1.16	1.2	1	1	2	2	1.33

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS 302** Course Title: **Data structures with C**
2. Contact Hours: **L:** 3 **T**: 0 **P**: 0
3. Examination Duration (Hrs): **Theory** 3 **Practical** 0
4. Relative Weight: **CIE** 25 **MSE** 25 **SEE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSC
8. Pre-requisite: Fundamentals of Computer & Introduction to Programming (TCS101), Programming for problem solving (TCS201)
9. Course Outcome:

	After completion of the course the students will be able to**:** CO1: Describe the concept of Data Structures and assess how the choice of data structures impacts the performance of programs CO2: Compare and contrast merits and demerits of various data structures in terms of time and memory complexity. CO3: Identify and propose appropriate data structure for providing the solution to the real world problems. CO4: Implement operations like searching, insertion, deletion, traversing mechanism etc. on various data structures CO5: Be familiar with advanced data structures such as balanced search trees, hash tables, AVL trees, priority queues, ADT etc. CO6: To augment merits of particular data structures on other data structure to develop innovation in subject of study.

10. **Details of the Course:**

SL. NO.	Contents	Contact Hours
1	**Unit 1:** **Introduction:** Basic Terminology, Pointer and dynamic memory allocation, Elementary Data Organization, Data Structure operations, Algorithm Complexity and Time-Space trade-off Arrays**:** Array Definition, Representation and Analysis, Single and Multidimensional Arrays, address calculation, application of arrays, Array as Parameters, Ordered List, Sparse	10

	Matrices. Stacks:Array. Representation and Implementation of stack, Operations on Stacks: Push & Pop, Array Representation of Stack, Linked Representation of Stack, Operations Associated with Stacks, Application of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack. Recursion: **Recursive** definition and processes, recursion in C, example of recursion, Tower of Hanoi Problem, tail recursion.	
2	**Unit 2:** **Queues:** Array and linked representation and implementation of queues, Operations on Queue: Create, Add, Delete, Full and Empty. Circular queue, Dequeue, and Priority Queue**.** **Linked list**: Representation and Implementation of Singly Linked Lists, Two-way Header List, Traversing and Searching of Linked List, Overflow and Underflow, Insertion and deletion to/from Linked Lists, Insertion and deletion Algorithms, Doubly linked list, Linked List in Array, Polynomial representation and addition, Generalized linked list.	10
3	**Unit 3:** **Trees**: Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree. Extended Binary Trees, Array and Linked Representation of Binary trees, Traversing Binary trees, Threaded Binary trees. Traversing Threaded Binary trees, Huffman algorithm & Huffman tree**.** **Searching and Hashing**: Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation	9
4	**Unit 4:** ***Sorting****:* Insertion Sort, Bubble Sorting, Quick Sort, Two Way Merge Sort, Heap Sort, Sorting on Different Keys, Practical consideration for Internal Sorting. **Binary Search Trees**: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees	9
5	**Unit 5:** **File Structures**: Physical Storage Media File Organization, Organization of records into Blocks, Sequential Files, Indexing and Hashing, Primary indices, Secondary indices, B+ Tree index Files, B Tree index Files, Indexing and Hashing Comparisons, Graph, Traversal(DFS,BFS) ,Minimum spanning tree	8
	Total	**46**

Text Books:

Authors Name	**Title**	**Edition**	**Publisher, Country**	**Year**
Horowitz and Sahani,	Fundamentals of data Structures	2nd Edition	Galgotia Publication Pvt. Ltd., New Delhi	2008

R. Kruse	Data Structures and Program Design in C	2nd Edition	Pearson Education Asia,	2006
A. M. Tenenbaum	Data Structures using C & C++",	2nd Edition	Prentice-Hall of India Pvt. Ltd., New Delhi.	2014
K Loudon,	Data Structures and Algorithms with Object Oriented Design Pattern in C++"	1st Edition	Shroff Publisher & Distributors Pvt. Ltd.	2000
Bruno R Preiss,	Data Structures and Algorithms in C++",	1st Edition	Jhon Wiley & Sons, Inc.	1998

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS302.1	Describe the concept of Data Structures and assess how the choice of data structures impacts the performance of programs	2	3	-	1	-	-	-	-	1	1	1	-	1	2	1
TCS302.2	Compare and contrast merits and demerits of various data structures in terms of time and memory complexity.	-	3	-	-	1	-	-	1	1	1	1	1	1	2	1
TCS302.3	Identify and propose appropriate data structure for providing the solution to the real world problems.	1	2	2	3	1	-	2	2	1	1	1	1	1	2	1
TCS302.4	Implement operations like searching, insertion, deletion, traversing mechanism etc. on various data structures	1	2	3	1	1	-	-	1	1	1	1	1	1	1	1
TCS302.5	Be familiar with advanced data structures such as balanced search trees, hash tables, AVL trees, priority queues, ADT etc.	-	1	2	3	1	-	1	1	1	1	1	1	2	2	1
TCS302.6	To augment merits of particular data structures on other data structure to develop innovation in subject of study.	1	1	2	3	1	-	1	1	2	1	2	2	2	2	2
TCS302			1.25	2	2.25	2.2	1	-	1.33	1.2	1.16	1	1.16	1.2	1.33	1.83

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS307** Course Title: **Object Oriented Programming with C++**
2. Contact Hours: **L:** 3 **T:** 0 **P:** 0
3. Examination Duration (Hrs): **Theory** 3 **Practical** 0
4. Relative Weight: **CIE** 25 25 **SEE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSC
8. Pre-requisite: Fundamental of Computer & Introduction to Programming (TCS101), Programming for problem solving (TCS201)

9. **Course Outcome:**	After completion of the course the students will be able to**:** CO1: Demonstrate the C++ Program uses data types, operators, expressions, array, strings and functions. CO2: Implement Constructors (Parameterized, Copy), this pointer, friend function, dynamic objects, arrays of objects. CO3: Illustrate the Operator Overloading of +, -, preincrement, postincrement, << and >>. CO4: Implement the single, multiple, multilevel and hybrid inheritance in C++. CO5: Illustrate function overloading, Overriding and virtual functions. CO6: Carry out exception handling techniques and provide solutions to storage related problems using STL.

10. **Details of the Course:**

Sl. No.	Contents	Contact Hours
1	**Unit 1:** **Introduction:** Need of object-oriented programming, Overview of C++, Header Files and Namespaces, Sample C++ program, Different data types, operators, expressions, and statements, arrays and strings, pointers & user-defined types function components, argument	10

	passing, inline functions, recursive functions.	
2	**Unit 2:** **Classes & Objects:** Class Specification, Objects, Scope resolution operator, Access members, defining member functions, Data hiding, Constructors, Parameterized constructors, Destructors, Static data members, Friend functions, passing objects as arguments, Returning objects, Arrays of objects, Dynamic objects, Pointers to objects, Copy constructors, This Pointer. **Operator overloading:** Fundamentals of Operator Overloading, Overloading Binary Operators and unary operators, Operator overloading using friend functions such as +, -, pre-increment, post-increment, overloading of << and >>.	9
3	**Unit 3:** **Inheritance:** Necessity of inheritance, Types of inheritance with examples, Base Class and Derived class, Public, private and protected access modifiers, inheriting multiple base classes, working of Constructors and Destructors in Inheritance, Passing parameters to base class constructors, Virtual base classes	9
4	**Unit 4:** **Virtual functions and Polymorphism**: Polymorphism, function overloading, Overriding Methods, Virtual function, Calling a Virtual function through a base class reference, Pure virtual functions, Abstract classes, Virtual Destructors, Early and late binding	9
5	**Unit 5:** **I/O System Basics and STL:** C++ stream classes, I/O manipulators, fstream and the File classes, basic file operations, function templates Exception Handling: Exception handling fundamentals, Throwing an Exception, Catching an Exception, Re-throwing an Exception, An exception example. **STL:** An overview, containers, vectors, lists, maps, Algorithms	9
	Total	**46**

Text Books:

Authors Name	**Title**	**Edition**	**Publisher, Country**	**Year**
Herbert Schildt	The Complete Reference C++,	4th Edition	McGraw Hill	2017
Balagurusamy E	Object oriented Programming with C++	8th Edition	McGraw Hill	2020

Reference Books:

Authors Name	**Title**	**Edition**	**Publisher, Country**	**Year**
Paul Deitel and Harvey Deitel	C++: How to Program	10th Edition	Pearson	2016

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS307.1	Demonstrate the C++ Program uses data types, operators, expressions, array, strings and functions.	3	-	-	-	-	-	-	-	-	-	-	-	2	-	-
TCS307.2	Implement Constructors (Parameterized, Copy), this pointer, friend function, dynamic objects, arrays of objects.	2	2	3	-	-	-	-	-	-	-	-	2	2	3	-
TCS307.3	Illustrate the Operator Overloading of +, -, preincrement, postincrement, << and >>.	2	2	2	-	2	-	-	-	-	-	-	1	2	1	-
TCS307.4	Implement the single, multiple, multilevel and hybrid inheritance in C++.	1	2	3	-	2	-	-	-	-	-	-	2	2	2	-
TCS307.5	Illustrate function overloading, Overriding and virtual functions.	2	2	2	-	2	-	-	-	-	-	-	1	2	1	-
TCS307.6	Carry out exception handling techniques and provide solutions to storage related problems using STL.	-	-	3	1	2	-	-	-	1	1	-	1	1	2	1
TCS307			2	2	2.6	1	2		-	-	1	1	-	1.4	1.83	1.8

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TMA316** Course Title: **Discrete Structures and Combinatorics**
2. Contact Hours: **L:** 3 **T:** 1 **P:** 0
3. Examination Duration (Hrs): **Theory** 3 **Practical** 0
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSC
8. Pre-requisite: Engineering Mathematics-I (TMA101), Engineering Mathematics-II (TMA201)

9.**Course Outcome:**	After completion of the course the students will be able to**:** CO1: Be able to specify and manipulate basic mathematical objects such as sets, functions, and relations. Demonstrate an understanding of partial order relations and Lattices. CO2: Understand the basics of discrete probability and number theory, and be able to apply the methods from these subjects in problem solving. CO3: Produce convincing arguments, conceive and/or analyze basic mathematical proofs and discriminate between valid and unreliable arguments. CO4: Discriminate, identify and prove the properties of groups and subgroups CO5: Be able to apply basic counting techniques to solve combinatorial problems CO6: Demonstrate different traversal methods for trees and graphs. Model problems in Computer Science using graphs and trees.

10. **Details of the Course:**

SL. NO.	Contents	Contact Hours
1	**Unit 1:** **Relations and Functions:** Review of Sets, Relations - properties, equivalence relation, matrix and Graph representation, Closure operations Functions, Types of functions, Invertability, Composition of functions and Inverse functions,Partially ordered Sets and Lattices. Lattice Properties, Lattices as Boolean Algebra	11
2	**Unit 2:** **Probability Theory** Basics of Probability, Conditional Probability; Random Variables, probability mass and density function, commutative distribution function, expected values, mean, variance and standard deviation, Distributions: Binomial. Poisson, normal, uniform,, exponential,	9
3	**Unit 3:** **Fundamentals of Logic:** Basic Connectives and Truth Tables, Logical Equivalence – The Laws of Logic, Logical Implication – Rules of Inference. The Use of Quantifiers, **Methods of Proof:** Different methods of proof – Direct Proof, Indirect Proof, Counter examples, Principle of Induction.	9
4	**Unit 4:** **Groups:** Definitions, Examples, and Elementary Properties, Homomorphism, Isomorphism, permutation groups and cyclic Groups, subgroups, cosets, and Lagrange's Theorem **Counting:** Set cardinality and counting, Sum and Product Rules, Inclusion Exclusion Principles, Pigeonhole principle, permutations and combinations, Basics of recurrence relations and, generating Functions	10
5	**Unit 5:** **Graphs and Trees** Fundamentals of Graphs Graph types – undirected, directed, weighted; - Representing graphs and graph isomorphism -connectivity-Euler and Hamilton paths, Isomorphism Tree properties, traversal techniques;	9
	Total	**48**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Kenneth H. Rosen	Discrete Mathematics and its Applications"	6th Edition	McGraw Hill,.	2007
Jayant Ganguly	A Treatise on Discrete Mathematical Structures"	2nd Edition	Sanguine-Pearson,.	2011

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
D.S. Malik and M.K. Sen	Discrete Mathematical Structures: Theory and Applications	2nd Edition	Thomson,.	2004
Thomas Koshy	Discrete Mathematics with Applications	1st Edition	Elsevier,.	2005, Reprint 2008
Ralph P. Grimaldi	Discrete and Combinatorial Mathematics	5th Edition	Pearson Education,	2004
S.B.Singh, Jaikishor and Ekata,	Discrete Mathematics	3rd Edition	Khanna Publication,.	2011

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TMA 316.1	Be able to specify and manipulate basic mathematical objects such as sets, functions, and relations. Demonstrate an understanding of partial order relations and Lattices.	3	-	-	-	-	-	-	-	1	-	1	-	-	-	1
TMA316.2	Understand the basics of discrete probability and number theory, and be able to apply the methods from these subjects in problem solving.	2	2	3	1	-	-	1	1	1	1	1	1	1	2	1
TMA316.3	Produce convincing arguments, conceive and/or analyze basic mathematical proofs and discriminate between valid and unreliable arguments.	1	1	1	2	-	-	-	1	1	2	1	-	1	2	-
TMA316.4	Discriminate, identify and prove the properties of groups and subgroups	3	-	-	-	-	-	-	-	-	1	1	-	1	1	-
TMA316.5	Be able to apply basic counting techniques to solve combinatorial problems	1	2	3	-	-	-	-	-	1	1	1	-	-	2	-
TMA316.6	Demonstrate different traversal methods for trees and graphs. Model problems in Computer Science using graphs and trees.	-	2	-	3	1	1	-	-	1	1	1	-	1	1	-
TMA316			2	1.75	2.33	2	1	1	1	1	1	1.2	1	1	1	1.6

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **PCS308** Course Title: **Logic Design and Computer Organization Lab**
2. Contact Hours: **L:** 0 **T:** 0 **P:** 2
3. Examination Duration (Hrs): **Theory** 0 **Practical** 2
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 1
6. Semester: III
7. Category of Course: DSC
8. Pre-requisite: Electronics Engineering Lab (PEC151 / PEC251)

9. **Course Outcome:**	After completion of the course the students will be able to**:** CO1: Understand various logic gates and digital circuits. CO2: Identify various digital ICs and understand its operation. CO3: Design elementary digital circuits under real and simulated environment. CO4: Simulate various logic circuits using simulation tool.

10. **Details of the Course:**

Sl. No.	List of problems for which student should develop program and execute in the Laboratory	Contact Hours
1.	To realize two and three variable Boolean functions using basic gates and universal gates digital IC.	2
2.	To design and test a half/full adder circuit using digital IC gates.	2
3.	To design and test a half/full subtractor circuit using IC gates.	2
4.	To design, implement and test the function F(A,B,C,D) = m(1,3,5,7,9,15) + d(4,6,12,13) using a NOR-OR implementation.	2
5.	To design and test RS, JK, D and T flip flops using logic gates.	2
6.	To design and test shift registers using flip-flops.	2
7.	To design and test an asynchronous up/down counter.	2
8.	To design, implement and test Boolean functions using a multiplexer.	2
9.	To design and simulate the implementation of Binary to Gray code conversion and vice versa using OrCAD/PSPICE.	2

10.	To design and simulate the implementation of 4-bit binary adder-subtractor circuit using OrCAD/PSPICE.	2
11.	To design and simulate the implementation of 2-bit binary multiplier circuit using OrCAD/PSPICE.	2
12.	To design and simulate the implementation of Ring and Johnson counter using OrCAD/PSPICE.	2
13.	To design and simulate Booths Algorithm using Verilog HDL.	2
14.	To design and simulate 32-bit Floating-Point multiplier using Verilog HDL.	2
15.	To design and simulate 8-bit ALU using Verilog HDL.	2
	Total	**30**

Text Books:

Authors Name	**Title**	**Edition**	**Publisher, Country**	**Year**
M. Morris Mano, Michael D. Ciletti	Digital Design: With an Introduction to the Verilog HDL, VHDL, and System Verilog,	6th Edition	Pearson	2018

Reference Books:

Authors Name	**Title**	**Edition**	**Publisher, Country**	**Year**
John P Hayes	Computer Architecture and Organization,	3rd Edition	McGraw Hill	2017

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
PCS308.1	Understand various logic gates and digital circuits.	2	1							2	1		1	1		1
PCS308.2	Identify various digital ICs and understand its operation.	1	3	2		1	1		1	2	1		1	2	1	2
PCS308.3	Design elementary digital circuits under real and simulated environment.	1	2	2	2	3		1	1	2	1	1	2	2	2	1
PCS308.4	Simulate various logic circuits using simulation tool.	1	2	2	2	3		2	1	2	1		2	2	2	2
PCS 308			1.25	2	2	2	2.33	1	1.5	1	2	1	1	1.5	1.75	1.66

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **PCS302** Course Title: **Data structures Lab**
2. Contact Hours: **L:** 0 **T:** 0 **P:** 2
3. Examination Duration (Hrs): **Theory** 0 **Practical** 2
4. Relative Weight: **CIE** 25 **MSE** 25 **SEE** 50
5. Credits: 1
6. Semester: III
7. Category of Course: DSC
8. Pre-requisite: Fundamental of Computer & Introduction to Programming (TCS101), Programming for problem solving (TCS201)

9. **Course Outcome:**	After completion of the course the students will be able to**:** CO1: Implement Stack, Queues using array in C programming language. CO2: Create Linked lists (single, double, circular) and perform various operations on Linked lists and implement Stack, Queue using Linked list in C programming language. CO3: Create Binary Search tree and perform operations such as traversal, deletion and execute Linear, Binary search, hashing and simple graph structure. CO4: Implement the sorting algorithm (Bubble, insertion, selection, merge, quick) and compare the performance of these algorithms

.

9. **Details of the Course:**

Sl. No.	List of problems for which student should develop program and execute in the Laboratory	Contact Hours
1.	**ARRAYS** **Week 1:** • Write a C program to calculate the sum of all even elements in an array. • Write a C program to determine the union of two given arrays.	

	• Write a C program to determine the intersection of two given arrays. • Write a C program to store N elements in an array and print the contents of the array in reverse order.	
2.	**Week 2:** • Write a C program to find the element with the maximum value from an array. • Write a C program to create a dynamic array. • Write a C program to Implementation Stack Using Array.	
3.	**Week 3:** • Write a C program to Implementation queue Using Array. • Write a C program to convert infix expression into postfix expression. • Write a C program to evaluate any postfix expression.	
4.	**LINKED LIST** **Week 4:** • Write a C program to create and display a Singly Linked List. • Write a C program to insert a new node at the beginning, middle and end of a Singly Linked List. • Write a C program to insert a new node at any given position in a singly linked list.	
5.	**Week 5:** • Write a C program to search for an existing element in a singly linked list. • Write a C program that allows the user to input a key for searching in a singly linked list. The program should delete the node containing the key value and update the linked list accordingly. If the key is not found, display the message “Unsuccessful Search.” • Write a C program to merge two sorted singly linked lists into a single sorted linked list.	
6.	**Week 6:** • Write a C program to create and display a doubly linked list. • Write a C program to insert a node at the beginning, middle and end of a doubly linked list. • Write a C program to insert a new node at any given position in a doubly linked list. • Write a C program to search for an element in a doubly linked list and delete that element from the list.	
7.	**Week 7:** • Write a C program to create and display a Circular linked list. • Write a C program to insert a node at the beginning, middle and end of a Circular linked list.	

	• Write a C program to insert a new node at any given position in a Circular linked list. • Write a C program to search for an element in a Circular linked list and then delete that element from the list.	
8.	**Week 8:** • Write a C program to insert string in linked list in alphabetical order. • Write a C program to remove duplicates from a linked list. • Write a C program to create and reorder a linked list placing all even-numbered nodes ahead of all odd-numbered nodes.	
9.	**Week 9:** • Write a C program to implement Stack Using linked List. • Write a C program to implement queue using double pointers. • Write a C program to implement Queue Using Linked List.	
10.	**TREES AND GRAPHS** **Week 10:** • Write a C program to create Binary search tree and perform following operations on it. **i) Insert node ii) Delete node iii) Search node**. • Write a C program to perform Inorder, Preorder and Postorder traversal on a Binary Search Tree. • Write a C program to delete a node from a Binary Search Tree.	
11.	**Week 11:** • Write a C program to calculate the height of a Binary Search Tree. • Write a C program to find the minimum and maximum values and total number of nodes in a Binary Search Tree. • Write a C program to check if a Binary Search Tree is balanced.	
12.	**Week 12:** • Write a C program to implement the Breadth-First Search (BFS) algorithm on a graph. • Write a C program to implement the Depth-First Search (DFS) algorithm on a graph. • Write a C program to create a simple graph structure using adjacency lists.	
13.	**SORTING TECHNIQUES and SEARCHING TECHNIQUES** **Week 13:** • Write a C program to sort an array using Bubble Sort technique. • Write a C program to sort an array using Selection Sort technique. • Write a C program to sort an array using Insertion Sort technique.	
14.	**Week 14:** • Write a C program to sort an array using Merge Sort technique. • Write a C program to sort an array using Quick Sort technique.	

	• Write a C program to implement Linear Search and Binary Search.	
	Total	

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Ellis Horowitz and Sartaj Sahni,	Fundamentals of Data Structures in C,	2nd Edition	Universities Press	2014
Yashavant Kanetkar	Data Structures Through C	3rd Edition	BPB	2019

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Seymour Lipschutz,	Data Structures Schaum's Outlines	1st Edition	McGraw Hill Education	2017
Amol M. Jagtap, Ajit S. Mali	Data Structures Using C A Practical Approach for Beginners	1st Edition	CRC	2021

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
PCS302.1	Implement Stack, Queues using array in C programming language	1	2	3	3	2			2	2	2		2	2	2	1
PCS302.2	Create Linked lists (single, double, circular) and perform various operations on Linked lists and implement Stack, Queue using Linked list in C programming language.	1	2	3	3	2		1	2	2	2		2	2	2	1
PCS302.3	Create Binary Search tree and perform operations such as traversal, deletion and execute Linear, Binary search, hashing and simple graph structure.	1	2	2	2	2	1		2	2	2		2	2	2	2
PCS302.4	Implement the sorting algorithm (Bubble, insertion, selection, merge, quick) and compare the performance of these algorithms	1	2	3	3	2		1	2	2	2	1	2	2	2	2
PCS302			1	2	2.75	2.75	2	1	1	2	2	2	1	2	2	2

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **PCS307** Course Title: **OOP WITH C++ LAB**
2. Contact Hours: **L:** 0 **T:** 0 **P:** 2
3. Examination Duration (Hrs): **Theory** 0 **Practical** 2
4. Relative Weight: **CIE** 50 **MSE** 25 **SEE** 50
5. Credits: 1
6. Semester: III
7. Category of Course: DSC
8. Pre-requisite: Computer Lab-I (PCS151), Computer Lab - II (PCS251)

9. **Course Outcome**:**	After completion of the course the students will be able to**:** CO1: Evaluate the basic difference between object-oriented programming and procedural language and their data types. CO2: Implement the programs using C++ features such as object creation, compile time polymorphism, inheritance, abstraction, encapsulation etc. CO3: Design and solve programs that incorporates the use of object-oriented techniques such as abstract classes, pure virtual functions, and constructors. CO4: Create programs based on the concepts of virtual base classes, virtual functions and STL to solve real time problems

10. **Details of the Course:**

Sl. No.	List of problems for which student should develop program and execute in the Laboratory	Contact Hours
1.	An electricity board charges the following rates to domestic users to discourage large consumption of energy. For the first 100 units: - 60 P per unit For the next 200 units: -80 P per unit Beyond 300 units: -90 P per unit All users are charged a minimum of Rs 50 if the total amount is more than Rs 300 then an additional surcharge of 15% is added. Implement a C++ program to read the names of users and number of units consumed and display the charges with names	
2.	Construct a C++ program that removes a specific character from a given string and return the updated string. Typical Input: computer science is the future Typical Output: compuer science is he fuure	

3.	Implement a C++ program to find the non-repeating characters in string. Typical Input: graphic era university	
	Typical Output: c g h n p s t u v y	
4.	You are given an array of elements. Now you need to choose the best index of this array. An index of the array is called best if the special sum of this index is maximum across the special sum of all the other indices. To calculate the special sum for any index you pick the first element that is and add it to your sum. Now you pick next two elements i.e., and and add both of them to your sum. Now you will pick the next elements, and this continues till the index for which it is possible to pick the elements. Find the best index and in the output print its corresponding special sum. Note that there may be more than one best index, but you need to only print the maximum special sum. **Input** First line contains an integer as input. Next line contains space separated integers denoting the elements of the array **Output** In the output you have to print an integer that denotes the maximum special sum	
5.	Implement a C++ program to demonstrate the concept of data abstraction using the concept of Class and Objects	

6.	Define a class Hotel in C++ with the following specifications **Private members** • Rno Data member to store room number • Name Data member to store customer name • Tariff Data member to store per day charges • NOD Data member to store number of days of stay • CALC() Function to calculate and return amount as NOD*Tariff ,and if the value of days* Tariff >10000, then total amount is 1.05* days*Tariff. **Public members** • Checkin() Function to enter the content Rno, Name, Tariff and NOD • Checkout() Function to display Rno, Name, Tariff, NOD and Amount (amount to be displayed by calling function) CALC()	
7.	Implement a Program in C++ by defining a class to represent a bank account. Include the following: **Data Members** ● Name of the depositor ● Account number ● Type of account (Saving, Current etc.) ● Balance amount in the account **Member Functions** ● To assign initial values ● To deposit an amount ● To withdraw an amount after checking the balance ● To display name and balance	

Input/Output Format	
Typical Input	**Expected Output**
5 1 3 1 2 5 10 2 1 3 9 2 4 -10 -9 1 3	8 9

8.	Anna is a contender for valedictorian of her high school. She wants to know how many students (if any) have scored higher than her in the exams given during this semester. Create a class named Student with the following specifications: ➢ An instance variable named scores holds a student's 5 exam scores. ➢ A void input () function reads 5 integers and saves them to scores. ➢ An int calculateTotalScore() function that returns the sum of the student's scores. **Input Format** In the void Student::input() function, you must read 5 scores from standard input and save them to your scores instance variable. **Output Format** In the int Student::calculateTotalScore() function, you must return the student's total grade (the sum of the values in scores). The code in the editor will determine how many scores are larger than Anna's and print that number to the console. **Sample Input** The first line contains n, the number of students in Anna's class. The n subsequent lines contain each student's 5 exam grades for this semester. 3 30 40 45 10 10 40 40 40 10 10 50 20 30 10 10 **Sample Output** 1	

9.	Construct a Program in C++ to show the working of function overloading(compile time polymorphism) by using a function named calculate Area () to calculate area of square, rectangle and triangle using different signatures as required.	

10	Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance. **Data Members -** • partNumber (type String) • partDescription (type String) • quantity of the item being purchased (type int) • price_per_item (type double) Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoiceAmount() that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive,it should be set to0.0. Write a test application named invoiceTest that demonstrates class Invoice's capabilities.	
11	Imagine a tollbooth with a class called TollBooth. The two data items are of type unsigned int and double to hold the total number of cars and total amount of money collected. A constructor initializes both of these data members to 0. A member function called payingCar( )increments the car total and adds 0.5 to the cash total. Another function called nonPayCar( ) increments the car total but adds nothing to the cash total. Finally a member function called display( )shows the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car and another to count a non paying car. Pushing the ESC key should cause the program to print out the total number of cars and total cash and then exit.	
12	Create a class called Time that has separate int member data for hours, minutes and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. A member function should display it in 11:59:59 format. A member function named add() should add two objects of type time passed as arguments. A main ( ) program should create two initialized values together, leaving the result in the third time variable. Finally it should display the value of this third variable.	
13	Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method	

	calculateMonthlyInterest() to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12.This interest should be added tosavingsBalance. Provide a static method modifyInterestRate() that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs2000.00 and Rs3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers	
14	Create a class Complex having two int type variable named real & img denoting real and imaginary part respectively of a complex number. Overload +, - , == operator to add, to subtract and to compare two complex numbers being denoted by the two complex type objects	
15	Using the concept of operator overloading. Implement a program to overload the following: a. Unary – b. Unary ++ preincrement, postincrement c. Unary -- predecrement, postdecrement	
16	Using the concept of operator overloading. Implement a program to overload the following: With the help of friend function a. Unary – b. Unary ++ preincrement, postincrement c. Unary -- predecrement, postdecrement	
17	Create a Base class that consists of private, protected and public data members and member functions. Try using different access modifiers for inheriting Base class to the Derived class and create a table that summarizes the above three modes (when derived in public, protected and private modes) and shows the access specifier of the members of base class in the Derived class.	
18	You are given three classes A, B and C. All three classes implement their own version of func. In class A, func multiplies the value passed as a parameter by 2. In class B, func multiplies the value passed as a parameter by 3. In class C, func multiplies the value passed as a parameter by 5.You are given class D such that You need to modify the class D and implement the function update_val which sets D's val to new_val by manipulating the value by only calling the func defined in classes A, B and C.It is guaranteed that new_val has only 2, 3 and 5 as its prime factors. Implement class D's function update_val. This function should update D's val only by calling A, B and C's func. **Sample Input**	

	new_val = 30 **Sample Output** A's func called 1 times B's func called 1 times C's func called 1 times	
19	Create a class called Student that contains the data members like age, name, enroll_no, marks. Create another class called Faculty that contains data members like facultyName, facultyCode, salary,deptt, age, experience, gender. Create the function display() in both the classes to display the respective information. The derived Class Person demonstrates multiple inheritance. The program should be able to call both the base classes and displays their information. Remove the ambiguity (When we have exactly same variables or same methods in both the base classes, which one will becalled?) by proper mechanism	
20	Implement a real case scenario by a proper C++ code to provide the solution to Diamond Problem in C++	
21	Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from base shape. Add to the base class, a member function get_data() to initialize base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived class to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area. Remember the two values given as input will be treated as lengths of two sides in the case of rectangles and as base and height in the case of triangle and used as follows: Area of rectangle = x * y Area of triangle = ½ *x*y	
21.	Create a base class called CAL_AREA(Abstract). Use this class to store float type values that could be used to compute the volume of figures. Derive two specific classes called cone, hemisphere and cylinder from the base CAL_AREA. Add to the base class, a member function getdata ( ) to initialize base class data members and another member function display volume( ) to compute and display the volume of figures. Make display volume ( ) as a pure virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a cone, cylinder and hemisphere interactively and display the volumes. Remember values given as input will be and used as follows:	

	Volume of cone = $(1/3)\pi r^2 h$ Volume of hemisphere = $(2/3)\pi r^3$ Volume of cylinder = $\pi r^2 h$	
22.	The task is to debug the existing code to successfully execute all provided test files. You are required to extend the existing code so that it handles the std::invalid_argument exception properly. More specifically, you have to extend the implementation of the process_input function. It takes integer n as an argument and has to work as follows: 1. It calls function largest_proper_divisor(n). 2. If this call returns a value without raising an exception, it should print in a single line result=d where d is the returned value. 3. Otherwise, if the call raises an invalid_argument exception, it has to print in a single line the string representation of the raised exception, i.e., its message. 4. Finally, no matter if the exception is raised or not, it should print in a single line returning control flow to the caller after any other previously printed output. **Input Format** The input is read by the provided locked code template. In the only line of the input, there is a single integer n, which is going to be the argument passed to function process input. **Output Format** The output should be produced by the function process_input as described in the statement. **Sample Input** 0 **Sample Output** the largest proper divisor is not defined for n=0 returning control flow to the caller Explanation 1 In the first sample, n = 0, so the call largest_proper_divisor(0) raises an exception. In this case, the function process_input prints two lines. In the first of them, it prints the string representation of the raised exception, and in the second line, it prints returning control flow to the caller. **Sample Input** 9 **Sample Output** result=3	
23.	Templates are the foundation of generic programming, which involves writing code in a way that is independent of any particular type. Write a program that can create a list (create a class list) of given type (int, float, char etc.) and perform insertion and deletion on list object.	

24.	Construct a C++ program to demonstrate different methods of List, Vector and Map in STL (Standard Template Library)	
25.	You are provided with a vector of N integers. Then, you are given 2 queries. For the first query, you are provided with 1 integer, which denotes a position in the vector. The value at this position in the vector needs to be erased. The next query consists of 2 integers denoting a range of the positions in the vector. The elements which fall under that range should be removed. The second query is performed on the updated vector which we get after performing the first query. Input Format The first line of the input contains an integer N. The next line contains N space-separated integers (1-based index). The third line contains a single integer x, denoting the position of an element that should be removed from the vector. The fourth line contains two integers a and b denoting the range that should be erased from the vector inclusive of a and exclusive of b. Output Format Print the size of the vector in the first line and the elements of the vector after the two erase operations in the second line separated by space. Sample Input 6 1 4 6 2 8 9 2 2 4 Sample Output 3 1 8 9 Explanation The first query is to erase the 2nd element in the vector, which is 4. Then, a modified vector is {1 6 2 8 9}, we want to remove the range of 2~4, which means the 2nd and 3rd elements should be removed. Then 6 and 2 in the modified vector are removed and we finally get {1 8 9}	
	Total	

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Herbert Schildt	The Complete Reference C++,	4th Edition	McGraw Hill	2017
Balagurusamy E	Object oriented Programming with C++	8th Edition	McGraw Hill	2020

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Paul Deitel and Harvey Deitel	C++: How to Program	10th Edition	Pearson	2016

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
PCS307.1	Evaluate the basic difference between object-oriented programming and procedural language and their data types.	1	3	-	-	2	1	-	2	1	2	-	1	2	1	2
PCS307.2	Implement the programs using C++ features such as object creation, compile time polymorphism, inheritance, abstraction, encapsulation etc.	1	2	2	3	2	1	-	1	1	2	-	2	3	1	1
PCS307.3	Design and solve programs that incorporates the use of object-oriented techniques such as abstract classes, pure virtual functions, and constructors.	3	1	3	-	2	1	-	2	1	2	-	2	2	3	3
PCS307.4	Create programs based on the concepts of virtual base classes, virtual functions and STL to solve real time problems	2	2	3	1	3	2	1	1	2	2	1	2	2	2	3
PCS307		1.75	2	2.66	2	2.25	1.25	1	1.5	1.25	2	1	1.75	2.25	1.75	2.25

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS351** Course Title: **Fundamental of Cloud Computing and Bigdata**
2. Contact Hours: **L:** 3 **T:** 1 **P:** 0
3. Examination Duration (Hrs): **Theory** 3 **Practical** 0
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSE
8. Pre-requisite: NA

9. **Course Outcome:**	After completion of the course the students will be able to**:** CO1: Identify the importance of cloud computing services for the digital ag technologies. CO2: Differentiate the services and deployment models of cloud computing. CO3: Evaluate the case studies of the different types of cloud computin applications. CO4: Analyze the cloud computing services management techniques providers, and standards. CO5: Distinguish the cloud computing services using Bigdata and big dat analytics. CO6: Design and deploy a cloud based web application.

10. **Details of the Course:**

Sl. No	Contents	Contact Hours
1	**Unit 1:** Introduction to Cloud Computing, Vision, History, Evolution, and Characteristics of Cloud Computing (NIST), Characteristic, Advantages and Disadvantages of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing, Importance of Open Standards for digital age technologies.	9
2	**Unit 2:** Working of Cloud Computing, Cloud Computing comparison with traditional computing architecture (client/server), Impact of Networks, Web Development and User Interface (UI) on Cloud computing. Cloud Deployment Models: Public cloud, Private cloud, Hybrid cloud, Community cloud.	9
3	**Unit 3:** Cloud Service Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). Infrastructure as a Service (IaaS): IaaS definition, Virtualization, Hypervisors, Machine Image, Virtual Machine (VM), Resource Virtualization, Server, Storage, Networking, Virtual Machine (resource) provisioning and manageability, Data centre physical plant/building, Networking firewalls/security, Data storage in cloud computing (storage as a service), Amazon Elastic Compute Cloud (EC2), Eucalyptus, Open Stack, Case Study of IaaS. Platform as a Service (PaaS): PaaS definition, Service Oriented Architecture (SOA), Cloud Platform and Management, Development tools, database management, business analytics, Operating systems, Google App Engine, Microsoft Azure, and Salesforce, Case Study of PaaS. Software as a Service (SaaS): SaaS definition, Web services, Web 2.0, Case Study of SaaS.	9
4	**Unit 4:** Introduction to Big Data, Characteristics, Architectures, Technologies, Applications, Advantages and Disadvantages of Big Data, Tools and Techniques applied in Big Data: Association rule learning, Classification tree analysis, Genetic algorithms, Machine learning, Regression analysis, Sentiment analysis, Social network analysis, Difference between big data and big data analytics. Introduction to Big Data analytics, Data Analysis Techniques: A/B testing, Data fusion and data integration, Data mining, Machine learning, Natural language processing (NLP), Statistics. Case study of Big Data.	9
5	**Unit 5:** Foundations Services of AWS: Savings, Security, Compliance and DRaaS Development Operations. AWS Services: Amazon Lambda, Amazon Relational Database Service (Amazon RDS), Amazon S3, Amazon CloudFront, Amazon Glacier, and Amazon SNS.	9

	Service Management in Cloud Computing: Service Level Agreements (SLAs), Billing & Accounting. Economics of Cloud Computing: SWOT Analysis and Value Proposition, General Cloud Computing Risks, (Performance, Network Dependence, Reliability, Outages, Safety Critical Processing Compliance and Information Security. Design and Deploy an Online Video Subscription Application on the Cloud.	
	Total	**45**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Rajkumar Buyya	Cloud Computing Principles and Paradigms,	1st Edition	Wiley,	**2013**
Kannammal,	Fundamentals of Cloud Computing	1st Edition	Cengage Learning,	**2015**
Barrie Sosinsky,	Cloud Computing Bible	1st Edition	Wiley-India	**2011**

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Jared Dean,	Bigdata Data Mining and Machine Learning	1st Edition	Wiley,	2014
Vince Reynolds,	Bigdata for Beginners, Create space	1st Edition	Independent Publishing Platform,	2016

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS351.1	Identify the importance of cloud computing services for the digital age technologies.	1	-	-	-	-	-	-	-	-	2	1	-	-	-	1
TCS351.2	Differentiate the services and deployment models of cloud computing.	1	2	3	-	2	-	2	2	2	1	2	1	1	1	2
TCS351.3	Evaluate the case studies of the different types of cloud computing applications.	1	3	-	-	1	1	-	1	1	1	1	1	-	-	2
TCS351.4	Analyze the cloud computing services management techniques, providers, and standards.	1	2	-	3	1	2	-	1	1	1	1	-	-	-	1
TCS351.5	Distinguish the cloud computing services using Bigdata and big data analytics.	1	2	3	1	2	1	2	1	1	1	2	1	-	1	2
TCS351.6	Design and deploy a cloud based web application.	2	2	3	-	3	-	3	2	1	1	2	1	2	3	1
TCS351		1.16	2.2	3	2	1.8	1.33	2.33	1.4	1.2	1.166	1.5	1	1.5	1.66	1.5

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS392** Course Title: **Introduction to Cryptography**
2. Contact Hours: **L:** 3 **T**: 1 **P**: 0
3. Examination Duration (Hrs): **Theory** 3 **Practical**
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSE
8. Pre-requisite: NA

9. **Course Outcome:**	After completion of the course the students will be able to**:** CO1: Classify security vulnerabilities involved in data communication over Internet and makeuse of classical algorithms to address the vulnerabilities. CO2: Apply symmetric block ciphers to secure data transmission and storage CO3: Analyze the various public key cryptographic systems and usage of hashing CO4: Appreciate the design of Public Key algorithms, mathematical background and make useof the same for data communication and message authentication CO5: Categorize types of viruses, worms, intrusion and decide measures to counter thethreats. CO6: Understand the legal aspects related to Cybercrime, Intellectual Property, Privacy,Ethical Issues.

.

10. **Details of the Course:**

Sl. No.	Contents	Contact Hours
1	**Unit – 1:** Introduction: Computer Security Concepts: The OSI Security Architecture, Security Attacks, Security Services, Security	8

	Mechanisms, a Model for Network Security, Standards Cryptography fundamentals and terminology; Cryptanalysis andBrute-Force Attack, Fundamental techniques of cryptography Substitution and Transposition; Classical Ciphers; Basics of Steganography.	
2	**Unit – 2:** Modern Cryptography: Symmetric Encryption and Message Confidentiality: Symmetric Encryption Principles, Fiestal structure. Symmetric Block Encryption Algorithms, Simple DES, double DES, Stream Ciphers and RC4, Random and Pseudorandom Numbers.	9
3	**Unit – 3:** Symmetric key distribution using symmetric encryption: A Key Distribution Scenario, Session Key Lifetime, A Transparent Key Control Scheme, Decentralized Key Control, Controlling Key Usage Mathematical Background for cryptography: prime number, Euclidean algorithm for GCD, Extended Euclidean algorithm for multiplicative inverse, Euler's totient function, their programming implementation.	10
4	**Unit 4:** Public-Key Cryptography: Public-Key Encryption Structure, Applications for Public-Key Cryptosystems, Requirements for Public-Key Cryptography, The RSA Public-Key Encryption Algorithm. Message Authentication: Approaches to Message Authentication, Authentication Using Conventional Encryption, Message Authentication without Message Encryption, MD5 Hash Algorithm.	9
5	**Unit 5:** System Security: Intruders, Intrusion Detection, Password Management, Types of Malicious Software, Viruses, Virus Countermeasures, Worms and Principles of Firewalls Legal and Ethical Aspects: Cybercrime and Computer Crime, Intellectual Property, Privacy, Ethical Issues.	9
	Total	**45**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
William Stallings	Network Security Essentials Applications and Standards,	6^{TH} Edition	Pearson	2018
William Stallings ,	Cryptography and Network Security	7^{TH} Edition	Pearson	2017

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Behrouz Forouzan	Cryptography and Network Security	3rd Edition	McGraw Hill	2015
Atul Kahate,	Cryptography and Network Security	3rd Edition	McGraw Hill	2017

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS392.1	Classify security vulnerabilities involved in data communication over Internet and make use of classical algorithms to address the vulnerabilities.	2	-	-	1	2	-	1	1	1	1	-	2	2	2	1
TCS392.2	Apply symmetric block ciphers to secure data transmission and storage	1	2	-	-	1	-	1	1	1	1	-	1	2	3	2
TCS392.3	Analyze the various public key cryptographic systems and usage of hashing	1	3	-	-	1	-	-	-	2	1	-	1	3	2	1
TCS392.4	Appreciate the design of Public Key algorithms, mathematical background and make use of the same for data communication and message authentication	1	1	3	-	-	-	-	-	2	1	-	1	3	3	2
TCS392.5	Categorize types of viruses, worms, intrusion and decide measures to counter thethreats.	-	1	2	1	-	-	-	1	1	1	-	1	2	2	3
TCS392.6	Understand the legal aspects related to Cybercrime, Intellectual Property, Privacy,Ethical Issues.	1	-	-	-	-	1	1	3	1	1	1	3	1	2	2
TCS392		1.2	1.75	2.5	1	1.33	1	1	1.5	1.33	1	1	1.5	2.166	2.33	1.833

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS331** Course Title: **Fundamental of IoT**
2. Contact Hours: **L:** 3 **T**: 0 **P**: 0
3. Examination Duration (Hrs): **Theory** 3 **Practical** 0
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSE
8. Pre-requisite: NA

9.**Course Outcome:**	After completion of the course the students will be able to**:** CO1: Explain the terms used in IoT. CO2: Describe key technologies in Internet of Things. CO3: Identify components needed to provide a solution for certain applications. CO4: Analyze security requirements in an IoT system. CO5: Design wireless sensor network architecture and its framework along with WSN applications. CO6: Understand business models for the Internet of Things.

.

10. **Details of the Course:**

SL. NO.	Contents	Contact Hours
1	**Unit 1:** **INTRODUCTION** Introduction to Internet of Things: History of IoT, About IoT, Overview and Motivations, Examples of Applications, Internet of Things Definitions and Frameworks: IoT Definitions, IoT Architecture, General Observations, ITU-T Views, Working Definition, IoT Frameworks, Basic Nodal Capabilities.	8

2	**Unit 2:** **FUNDAMENTAL IoT MECHANISMS AND KEY TECHNOLOGIES** Identification of IoT Objects and Services, Structural Aspects of the IoT, Environment Characteristics, Traffic Characteristics, Scalability, Interoperability, Security and Privacy, Open Architecture, Key IoT Technologies, Device Intelligence, Communication Capabilities, Mobility Support, Device Power, Sensor Technology, RFID Technology, Satellite Technology.	10
3	**Unit 3:** **RADIO FREQUENCY IDENTIFICATION TECHNOLOGY** RFID: Introduction, Principle of RFID, Components of an RFID system, Issues EPCGlobal Architecture Framework: EPCIS & ONS, Design issues, Technological challenges, Security challenges, IP for IoT, Web of Things. Wireless Sensor Networks: History and context, WSN Architecture, the node, Connecting nodes, Networking Nodes, Securing Communication WSN specific IoT applications, challenges: Security, QoS, Configuration, Various integration approaches, Data link layer protocols, routing protocols and infrastructure establishment.	10
4	**Unit 4:** **RESOURCE MANAGEMENT IN THE INTERNET OF THINGS** Clustering, Software Agents, Clustering Principles in an Internet of Things Architecture, Design Guidelines, and Software Agents for Object Representation, Data Synchronization. Identity portrayal, Identity management, various identity management models: Local, Network, Federated and global web identity, user-centric identity management, device centric identity management and hybrid-identity management, Identity and trust.	10
5	**Unit 5:** **INTERNET OF THINGS PRIVACY, SECURITY AND GOVERNANCE** Vulnerabilities of IoT, Security requirements, Threat analysis, Use cases and misuse cases, IoT security tomography and layered attacker model, Identity establishment, Access control, Message integrity, Non-repudiation and availability, Security model for IoT. Internet of Things Application: Smart Metering Advanced Metering Infrastructure, e-Health Body Area Networks, City Automation, Automotive Applications, Home Automation, Smart Cards.	10
	Total	**48**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Daniel Minoli	Building the Internet of Things with IPv6 and MIPv6: The	1st Edition	Willy Publications	**2013**

	Evolving World of M2M Communications",			
Bernd Scholz-Reiter, Florian Michahelles	Architecting the Internet of Things",	**1st** Edition	Springer	**2011**
Parikshit N. Mahalle & Poonam N. Railkar	Identity Management for Internet of Things",	**1st** Edition	River Publishers,	**2015**

Reference Books:

Authors Name	**Title**	**Edition**	**Publisher, Country**	**Year**
HakimaChaouchi,	"The Internet of Things Connecting Objects to the Web"	1st Edition	Willy Publications	2010
Olivier Hersent, David Boswarthick, Omar Elloumi	The Internet of Things: Key Applications and Protocols,	1st Edition	Willy Publications	2015
Daniel Kellmereit, Daniel Obodovski	The Silent Intelligence: The Internet of Things"	1st Edition	Lightning Source Inc;	2014
Fang Zhaho, Leonidas Guibas	"Wireless Sensor Network: An information processing approach",	1st Edition	Elsevier,	2005

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS331.1	Explain the terms used in IoT	1	-	-	-	-	-	-	1	1	2	1	-	1	-	1
TCS331.2	Describe key technologies in Internet of Things.	1	-	-	-	-	-	-	-	1	2	1	-	1	-	1
TCS331.3	Identify components needed to provide a solution for certain applications.	1	2	2	-	1	-	1	1	1	1	1	-	-	1	1
TCS331.4	Analyze security requirements in an IoT system.	2	3	-	1	1	-	-	2	1	-	2	-	1	-	1
TCS331.5	Design wireless sensor network architecture and its framework along with WSN applications.	1	2	-	2	2	1	-	-	1	-	1	2	2	1	2
TCS331.6	Understand business models for the Internet of Things.	1	3	-	-	2	1	-	1	1	-	2	-	1	2	1
TCS331		1.166	2.5	2	1.5	1.5	1	1	1.25	1	1.66	1.33	2	1.2	1.33	1.16

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS332** Course Title: **Fundamental of Information Security and Block chain**
2. Contact Hours: **L:** 3 **T:** 1 **P:** 0
3. Examination Duration (Hrs): **Theory** 3 **Practical**
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSE
8. Pre-requisite: NA

9. **Course Outcome:**	After completion of the course the students will be able to**:** CO1: Explain information security and blockchain CO2: Know the working of information security techniques CO3: Analyze the different information security protocols CO4 Use Blockchain to implement information security protocols CO5 Apply information security techniques in different applications CO6: Develop blockchain enabled information security protocols

.

10. **Details of the Course:**

Sl. No.	Contents	Contact Hours
1	**Unit – 1:** **Introduction to information security** What is information security, why we need information security, the zero trust model, overview of ethical hacking Protection against- unauthorised modification, unauthorised deletion and unauthorised access, different types of user authentication techniques, access control techniques	9

	Pillars of information security - confidentiality, availability and integrity Steps to fix a cyber crime - Identify cyber threats, analyse and evaluate threat, treatment Type of hackers - white hat, grey hat, black hat Penetration testing and its phases - reconnaissance, scanning, gaining access, maintaining access, covering tracks	
2	**Unit – 2:** **Linux Basics and Scripting for Information Security** Bash, linux commands, man page, adding and deleting, users and adding them to sudo group, switching users, creating, copying, moving and removing file, writing and appending text to a file, file permissions, working with editors, grep, cut command, starting and stopping services Introduction to bash scripting- basics of bash or shell scripting, conditional statements, loops, manipulating files Introduction to python - Basics of python, conditional statements, loops, list, tuple, dictionary, functions	8
3	**Unit – 3:** **Basics of Network and Web Security** TCP 3-way handshake, netcat - The Swiss Army Knife of TCP/IP Connections, use netcat to Listen on a port, pushing a command shell back to listener, transfer files, ICMP and Ping command Cross site scripting (XSS) attack, Phishing, Spear Phishing, Sql Injection Attack, Wireshark - A Packet Sniffing Tool	10
4	**Unit 4:** Overview of blockchain- Overview of blockchain, structure of a block, block header, block identifiers: block header hash and block height, genesis block, linking of blocks, merkle trees, and use of merkle root in payment verification	10
5	**Unit 5:** Blockchain mechanisms: Details of distributed ledger, smart contracts, bitcoins networks, mining process, consensus algorithms, proof of work, proof of stake, proof of weight, proof of capacity.	8
	Total	**45**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Georgia Weidman	Penetration Testing: A Hands-on Introduction to Hacking", No Starch Press,	1st Edition	No Starch Press,	**2020**
George Icahn	Blockchain: the complete guide to understanding	1st Edition	Createspace Independent Pub	**2017**

	blockchain technology",			
Antony lewis	The basics of bitcoins and blockchains: an introduction to cryptocurrencies and the technology that powers them"	3rd Edition	Mango Media	**2018**

Reference Books:

Authors Name	**Title**	**Edition**	**Publisher, Country**	**Year**
Andreas M. Antonopoulos,	Mastering Bitcoin: unlocking digital cryptocurrencies",	2nd Edition	O'Reilly Media,	**2017**
Roger Wattenhofer,	Distributed Ledger Tehnology, The science of the Blockchain",	2nd Edition	Inverted Forest Publishing,	**2017**
Antonopoulos, Andreas M. and Wood, Gavin	Mastering Ethereum.	2nd Edition	O'Reilly Media,.	**2018**

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS332.1	Explain information security and blockchain	-	1	-	-	-	1	-	-	1	2	-	¬1	2	2	3
TCS332.2	Know the working of information security techniques	1	2	-	-	-	-	-	-	1	1	-	1	2	1	3
TCS332.3	Analyze the different information security protocols	1	3	-	1	-	1	-	-	1	1	-	2	3	2	1
TCS332.4	Use Blockchain to implement information security protocols	1	2	-	3	2	-	-	-	1	1	-	2	2	2	3
TCS332.5	Apply information security techniques in different applications	3	-	-	-	1	1	-	-	1	-	2	2	2	3	2
TCS332.6	Develop blockchain enabled information security protocols	1	1	3	1	1	-	-	-	2	1	2	2	2	2	3
TCS332		1.4	1.8	3	1.66	1.33	1	-	-	1.16	1.2	2	1.8	2.16	2	2.5

High _ (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS346** Course Title: **Python Programming**
2. Contact Hours: **L:** **3** **T**: **0** **P**: **0**
3. Examination Duration (Hrs): **Theory** **3** **Practical** **0**
4. Relative Weight: **CIE** **25** **MSE** **25** **ESE** **50**
5. Credits: **3**
6. Semester: **3**
7. Category of Course: **DSC**
8. Pre-requisite: Fundamental of Computer & Introduction to Programming (TCS101), Programming for problem solving (TCS201)

9. **Course Outcome:**	After completion of the course, the students will be able to**:** CO1: Identify and describe the fundamental data types and basic operators used in Python programming. CO2: Explain the purpose and usage of functions in Python, including parameters, return values, and modules. CO3: Apply data structures like lists, tuples, and dictionaries to organize and manipulate data for engineering problems. CO4: Analyze the differences between object-oriented programming concepts like inheritance, polymorphism, and encapsulation, and their impact on program design. CO5: Evaluate the suitability of various libraries like NumPy, Pandas, Matplotlib, and Seaborn for specific tasks related to numerical computing, data analysis, and visualization in engineering applications. CO6: Design and implement Python programs using fundamental concepts, data structures, and libraries to solve basic engineering problems involving data cleaning, analysis, and visualization.

10. **Details of the Course:**

Sl. No.	Contents	Contact Hours
1	**UNIT 1: Python Basics and Functions and Modules** Syntax and Semantic Basics: Data types: strings, integers, floats, Variable assignments and expressions, Basic input/output operations Defining and calling functions: Function parameters and return values- Using built-in modules, Creating and using custom modules, Exception handling basics	10
2	**UNIT 2: Data Handling** Lists: creation, indexing, slicing, and methods Tuples: usage and when to use Dictionaries: creating, accessing, and manipulating Reading from and writing to files: text and binary files	8
3	**UNIT 3: Object-Oriented Programming** Introduction to classes and objects, Attributes and methods, Inheritance: extending classes Polymorphism: using a unified interface Encapsulation: private and public members	10
4	**UNIT 4: Libraries for AI/ML** NumPy: arrays, array operations, indexing, reshaping Pandas: DataFrame operations, indexing, merging, grouping Matplotlib: basic plotting, figures, and axes Seaborn: statistical data visualization	10
5	**UNIT 5: Intro to Data Science** Data cleaning techniques, Exploratory data analysis (EDA): summary statistics, correlation, Visualization techniques: histograms, scatter plots, box plots	8
	Total	**46**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Eric Matthes	Python Crash Course: A Hands-On, Project-Based Introduction to Programming	3rd Edition	No Starch Press, USA	2023
Mark Lutz	Learning Python	5th Edition	O'Reilly Media, USA	2013
Wes McKinney	Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython	2nd Edition	O'Reilly Media, USA	2017

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Joel Grus	Data Science from Scratch: First Principles with Python	2nd Edition	O'Reilly Media, USA	2019
Al Sweigart	Automate the Boring Stuff with Python: Practical Programming for Total Beginners	2nd Edition	No Starch Press, USA	2019

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS346.1	Identify and describe the fundamental data types and basic operators used in Python programming	1	2	-	1	-	-	-	-	-	-	-	2	3	2	1
TCS346.2	Explain the purpose and usage of functions in Python, including parameters, return values, and modules.	1	-	2	1	-	-	-	-	-	-	-	-	2	1	1
TCS346.3	Apply data structures like lists, tuples, and dictionaries to organize and manipulate data for engineering problems.	-	-	3	-	-	-	-	-	-	-	-	-	3	1	1
TCS346.4	Analyze the differences between object-oriented programming concepts like inheritance, polymorphism, and encapsulation, and their impact on program design.	-	3	-	-	-	-	-	-	-	-	-	2	3	2	1
TCS346.5	Evaluate the suitability of various libraries like NumPy, Pandas, Matplotlib, and Seaborn for specific tasks related to numerical computing, data analysis, and visualization in engineering applications.	3	-	-	-	2	-	-	-	-	-	-	-	2	2	1
TCS346.6	Design and implement Python programs using fundamental concepts, data structures, and libraries to solve basic engineering problems involving data cleaning, analysis, and visualization.	3	-	2	3	3	-	-	-	-	-	-	3	3	2	2
TCS346		2	2.5	2.33	1.66	2.5	-	-	-	-	-	-	2.33	2.66	1.66	1.16

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS364** Course Title: **Fundamentals of Artificial Intelligence and Machine Learning**
2. Contact Hours: **L:** 3 **T**: 0 **P**: 0
3. Examination Duration (Hrs): **Theory** 3 **Practical** 0
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSC
8. Pre-requisite: Fundamental of Computer & Introduction to Programming (TCS101)

9. **Course Outcome:**	After completion of the course, the students will be able to**:** CO1: Define Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning, and differentiate between them. CO2: Explain problem-solving frameworks in AI and describe search strategies like breadth-first, depth-first, and A*. CO3: Choose appropriate classification techniques like Logistic Regression, KNN, or SVM based on specific data characteristics. CO4: Compare and contrast different performance metrics like accuracy, precision, recall, and F1-score for evaluating ML models. CO5: Evaluate the strengths and limitations of specific unsupervised learning techniques like K-means and hierarchical clustering for a given task. CO6: Design a simple machine learning pipeline involving data pre-processing, model selection, and evaluation for a classification task.

10. **Details of the Course:**

Sl. No.	Contents	Contact Hours
1	**UNIT 1: Overview of AI and Intelligent Agents** Definition and history of AI, Differences between AI, ML, and deep learning, Applications and impact of AI in various sectors, Agents and environments, Types of agents (simple reflex, model-based, goal-based, utility-based), Agent architecture and environments **Problem Solving:** Problem-solving frameworks, Search strategies: breadth-first, depth-first, A* Heuristics: designing and applying heuristics	9
2	**UNIT 2: Knowledge and Reasoning** Knowledge-based AI, Logic and Reasoning: propositional and predicate logic, Inference in first-order logic, Building knowledge bases. **Uncertainty Handling:** Probabilities and Bayesian networks, Decision making: Expected utility- Markov decision processes	9
3	**UNIT 3: Introduction to ML and Data Preprocessing** What is Machine Learning? Types of Machine Learning: Supervised, Unsupervised, Reinforcement- ML in practice: Applications and case studies, Importance of data preprocessing, Data cleaning, normalization, and transformation, Feature selection, and dimensionality reduction. **Regression Analysis** Linear regression, Polynomial regression, Regularization methods: Ridge, Lasso	10
4	**UNIT 4: Classification Techniques** Logistic regression, K-nearest neighbors (KNN), Support vector machines (SVM) **Decision Trees and Random Forests** Building decision trees, Overfitting and pruning, Ensemble methods: Random Forests and boosting	10
5	**UNIT 5: Evaluation of ML Models and Clustering and Association** Training and testing data splits, Performance metrics: accuracy, precision, recall, F1-score, Confusion matrix and ROC curves K-means clustering, Hierarchical clustering, Apriori algorithm for association rule learning	10
	Total	**48**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Stuart Russell and Peter Norvig	Artificial Intelligence: A Modern Approach	4th Edition	Pearson Education Limited, USA	2024
Melanie Mitchell	Introduction to Artificial Intelligence	2nd Edition	McGraw-Hill Education, USA	2024
Tom M. Mitchell,	Machine Learning	1st Edition	Mc Graw Hill Publisher	2017
Manaranjan Pradhan, U Dinesh Kumar	Machine Learning using Python	1st Edition	Wiley, India	2017

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Richard E. Neapolitan and Kevin Goda	Artificial Intelligence: Foundations and Applications	4th Edition	Elsevier Science & Technology, Netherlands	**2024**
Aurélien Géron	Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow	2nd Edition	O'Reilly Media, Inc., USA	2024
E. Alpaydin,	Introduction to Machine Learning	3rd Edition	PHI Publisher	2015

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS364.1	Define Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning, and differentiate between them	1	2	-	-	-	-	-	-	-	-	-	2	2	1	1
TCS364.2	Explain problem-solving frameworks in AI and describe search strategies like breadth-first, depth-first, and A*	-	2	-	-	-	-	-	-	-	-	-	1	2	2	1
TCS364.3	Choose appropriate classification techniques like Logistic Regression, KNN, or SVM based on specific data characteristics	2	-	-	2	-	-	-	-	-	-	-	2	2	2	1
TCS364.4	Compare and contrast different performance metrics like accuracy, precision, recall, and F1-score for evaluating ML models	-	-	-	2	-	-	-	-	-	-	-	1	3	1	1
TCS364.5	Evaluate the strengths and limitations of specific unsupervised learning techniques like K-means and hierarchical clustering for a given task	1	2	-	-	-	-	-	-	-	-	-	1	3	1	1
TCS364.6	Design a simple machine learning pipeline involving data pre-processing, model selection, and evaluation for a classification task	-	-	3	-	-	-	-	-	-	-	-	2	3	2	1
TCS364		**1.33**	**2**	**3**	**2**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**1.5**	**2.5**	**1.5**	**1**

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS-343** Course Title: Mathematical Foundations for Artificial Intelligence
2. Contact Hours: **L:** 3 **T**: 1 **P**: 0
3. Examination Duration (Hrs): **Theory** 3 **Practical** 0
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: 3
7. Category of Course: DSE
8. Pre-requisite: (**TMA 101) Engineering Mathematics I, (TMA 201) Engineering Mathematics II**

9. **Course Outcome:**	After completion of the course, the students will be able to**:** CO1: Understand the basic concepts of Linear Algebra such as System of Linear Equation, Matrices, Vector Space, Rank, etc. CO2: Understand the basic principles of probability, Bayes theorem, understand the definitions of discrete, continuous, and joint random variables, compute the mean, variance and covariance of random variables. CO3: Solve problems on matrix decompositions such as Choleskey Decomposition, Eigen Decomposition and Diagonalization, Singular Value Decomposition CO4: Describe the vector calculus concepts such as differentiation of Univariate Function, Partial Differentiation and Gradients. CO5: Analyze various mathematical concepts, that are required to build AI & ML models. CO6: Create an AI & ML models by applying the concepts of mathematics such as Linear Algebra, Analytical Geometry, Matrix, Calculus, Probability, etc.

10. **Details of the Course:**

Sl. No.	Contents	Contact Hours
1	**Unit 1:** **Linear Algebra:** System of Linear Equation**,** Matrices, Solving system of Linear Equation, Vector Spaces, Linear Independences, Basis and Rank, Linear Mappings, Affine Space.	10
2	**Unit 2:** **Analytic Geometry:** Norms, Inner Products, Lengths and Distances, Angles and Orthogonality, Orthonormal basis, Orthogonal Compliment, Inner Product of Function, Orthogonal Projections, Rotations.	10
3	**Unit 3:** **Matrix Decomposition** Determinant and Trace, Eigen Values and Eigen Vectors, Choleskey Decomposition, Eigen Decomposition and Diagonalization, Singular Value Decomposition, Matrix Approximation, Matrix Pylogency	10
4	**Unit 4:** **Vector Calculus** Differentiation of Univariate Function, Partial Differentiation and Gradients, Gradients of Vector-Valued Functions, Gradients of Matrices, Linearization and Multivariate Taylor Series	10
5	**Unit 5:** Probability and Distribution Discrete and Continuous Probability, Sum Rule, Product Rule, Bayes' Theorem, Gaussian Distribution, Change of Variables/Inverse Transform	10
	Total	**50**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Marc Peter Deisenroth , A. Aldo Faisal, Cheng Soon Ong,	MATHEMATICS FOR MACHINE LEARNING	1st	Cambridge University ,Press	2020
Jay Dawani,	Hands-On Mathematics for Deep Learning: Build a solid mathematical foundation for training efficient deep neural networks	1st	Packt Publishing Limited, UK	2020

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Tamoghna Ghosh, Shravan Kumar Belagal	Shravan Kumar Belagal Math, Practical Mathematics for AI and Deep Learning	1st	BPB Publications ,India	2022

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS343.1	Understand the basic concepts of Linear Algebra such as System of Linear Equation, Matrices, Vector Space, Rank, etc.	2	3	-	1	-	-	-	-	1	1	1	-	1	2	1
TCS343.2	Understand the basic principles of probability, Bayes theorem, understand the definitions of discrete, continuous, and joint random variables, compute the mean, variance and covariance of random variables.	-	3	-	-	1	-	-	1	1	1	1	1	1	2	1
TCS343.3	Solve problems on matrix decompositions such as Choleskey Decomposition, Eigen Decomposition and Diagonalization, Singular Value Decomposition	1	2	2	3	1	-	2	2	1	1	1	1	1	2	1
TCS343.4	Describe the vector calculus concepts such as differentiation of Univariate Function, Partial Differentiation and Gradients.	1	2	3	1	1	-	-	1	1	1	1	1	1	1	1
TCS343.5	Analyze various mathematical concepts, that are required to build AI & ML models.	-	1	2	3	1	-	1	1	1	1	1	1	2	2	1
TCS343.6	Create an AI & ML models by applying the concepts of mathematics such as Linear Algebra, Analytical Geometry, Matrix, Calculus, Probability, etc.	1	1	2	3	1	-	1	1	2	1	2	2	2	2	2
TCS 343		1.25	2.00	2.25	2.20	1.00	-	1.33	1.20	1.17	1.00	1.17	1.20	1.33	1.83	1.17

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS 342** Course Title: **Introduction to Statistical Data Science**
2. Contact Hours: **L:** 3 **T**: 1 **P**: 0
3. Examination Duration (Hrs): **Theory** 3 **Practical** 0
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: III
7. Category of Course: DSE
8. Pre-requisite: Fundamental of Computer & Introduction to Programming (TCS101

9. **Course Outcome:**	After completion of the course, the students will be able to**:** CO1: Critically analyze statistical methodologies in order to assess best practice guidance when applied to real-world problems in specific contexts CO2: Investigate and evaluate key concepts of statistics and data science techniques and assess when to apply such techniques in practical situations CO3: Contextualize, implement statistical models using different statistical tools CO4: Develop the ability to build and assess data-based models. CO5: Understand fundamental principles of statistics and data science applications and technologies in order to provide strategies to address processing of datasets with a variety of characteristics. CO6: Apply knowledge about algorithms for statistical analysis, machine learning or data extraction in new areas within data science.

10. **Details of the Course:**

Sl. No.	Contents	Contact Hours
1	**Unit 1:** Introduction to data science: Introduction to modern data analysis, Data science components, Data Science Applications. Fundamental Statistics- Probability Distribution: Introduction to Probability, Probability Distribution (Continuous and discrete- Normal, Bernoulli, Binomial, Negative Binomial, Geometric and Poisson distribution)	**10**
2	**Unit 2:** Discrete random variables, Continuous random variable, Markov-chain Monte Carlo, Descriptive Statistics- Sample covariance, Sample covariance matrix, Outlier.	**10**
3	**Unit 3:** Concepts of Correlation, Regression, Linear square estimation, Simple Linear Regression, Multiple Regression	**9**
4	**Unit 4:** Naïve Bayes' Theorem, Bayesian classification, Central Limit theorem, Data Exploration & preparation, Confidence Interval, The hypothesis-testing, Z-Score.	**8**
5	**Unit 5:** Parametric Testing: t-Test and Z-Test, Non-parametric Testing: ANOVA and chi-Square	**10**
	Total	**47**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Douglas C. Montgomery , George C. Runger	Applied Statistics and Probability for Engineers	6th	Wiley, United States	2016
M. Ross	Introduction to Probability and Statistics for Engineers and Scientists	4th	Academic Press, United States	2009
James D. Miller	Statistics for Data Science	1st	Packt Publishing Limited ,UK	2017

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Dr. D.C. Agarwal & Dr. Pradeep K. Joshi	Probability & Statistics for Data Science	1st	Shree Sai Prakashan,India	2022
Dr. Mark Gardener, "Beginning R	The Statistical Programming Language	1st	John willey& Sons, USA	2012

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS342.1	Critically analyze statistical methodologies in order to assess best practice guidance when applied to real-world problems in specific contexts	3	-	-	-	-	-	-	-	-	-	-	-	2	-	-
TCS342.2	Investigate and evaluate key concepts of statistics and data science techniques and assess when to apply such techniques in practical situations	2	2	3	-	-	-	-	-	-	-	-	2	2	3	-
TCS342.3	Contextualize, implement statistical models using different statistical tools	2	2	2	-	2	-	-	-	-	-	-	1	2	1	-
TCS342.4	Develop the ability to build and assess data-based models.	1	2	3	-	2	-	-	-	-	-	-	2	2	2	-
TCS342.5	Understand fundamental principles of statistics and data science applications and technologies in order to provide strategies to address processing of datasets with a variety of characteristics.	2	2	2	-	2	-	-	-	-	-	-	1	2	1	-
TCS342.6	Apply knowledge about algorithms for statistical analysis, machine learning or data extraction in new areas within data science.	-	-	3	1	2	-	-	-	1	1	-	1	1	2	1
TCS342		2.00	2.00	2.60	1.00	2.00	-	-	-	1.00	1.00	#DIV/0!	1.40	1.83	1.80	1.00

High _ (3); Medium correlation (2); Low correlation (1), No _ (-)

GRAPHIC ERA HILL UNIVERSITY, DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS-324** Course Title: **Information Security Foundations**
2. Contact Hours: **L:** 3 **T**: 1 **P**: 0
3. Examination Duration (Hrs): **Theory** 4 **Practical** 0
4. Relative Weight: **CIE** 25 **MSE** 25 **ESE** 50
5. Credits: 3
6. Semester: 3
7. Category of Course: **DSE**
8. Pre-requisite: **Fundamental of Computers (TCS 101), Programming for problem solving (TCS 201)**

9. **Course Outcome:**	After completion of the course the students will be able to**:** CO1: Explain symmetric and asymmetric key cryptosystems. CO2: Know the working of cryptography techniques. CO3: Analyze the different types of cryptosystems. CO4: Use cryptographic techniques to implement information security protocols. CO5: Apply cryptographic techniques in different applications. CO6: Develop symmetric and asymmetric key cryptosystems.

10. **Details of the Course:**

Sl. No.	Contents	Contact Hours
1	**Unit 1:** **Introduction to information security** What is information security, why we need information security, the zero trust model, overview of ethical hacking Protection against- unauthorized modification, unauthorized deletion and unauthorized access, different types of user authentication techniques, access control techniques Pillars of information security - confidentiality, availability and integrity Steps to fix a cyberc rime - Identify cyber threats, analyze and evaluatethreat, treatment Type of hackers - white hat, grey hat, black hat Penetration testing and its phases - reconnaissance, scanning, gaining access, maintaining access, covering tracks. SSL and Transport layer security.	10
2	**Unit 2:** **Basics of cryptography** What is cryptography, what is confidentiality, data integrity, authentication, and nonrepudiation, applications of cryptography - chip based payment cards, digital currencies, computer passwords, digital communications, plaintext, cipher-text, cipher - characteristics of a good cipher, encryption, decryption, Key - significance of key length, symmetric and asymmetric key cryptography, cryptanalysis, OSI security architecture- security attacks, security services, security mechanisms	10
3	**Unit 3:** **Mathematics applied in information security** Concept of divisibility, prime numbers, importance of prime numbers in cryptography, euclid theorem for GCD, extended euclidean algorithm, modular arithmetic, random number generators, deterministic and nondeterministic random number generators, XOR, bit shifts, euler's totient theorem, chinese remainder theorem.	8

4	**Unit 4:** **Symmetric key cryptosystem** Secret Key (symmetric) cryptography - stream and block ciphers, additive and multiplicative ciphers, rail fence technique, playfair cipher, hill cipher, vernam cipher, Vigenère Cipher, RC4 algorithm, DES, 2DES, 2-3DES, 3DES, AES, block cipher modes of operations.	10
5	**Unit 5:** **Asymmetric key cryptosystem, digital signature, and message integrity** RSA, Diffie Hellman key exchange protocol, Elliptic curve cryptography (ECC), ElGamal encryption system. DSS algorithm, RSADS algorithm, ECDSA algorithm, Message integrity, hash functions, MAC functions, HMAC	8
	Total	**46**

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
William Stallings	Cryptography and Network Security: Principles and Practice	8th	Pearson Publication, India	2020

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Charles P. Pfleeger, Shari Lawrence Pfleeger, Jonathan Margulies	Security in Computing", 5th Edition	**5th**	Prentice Hall, India	**2015**
William Stallings	Network Security Essentials: Applications and Standards	**6th**	Prentice Hall, India	**2016**

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS324.1	Explain symmetric and asymmetric key cryptosystems.	2											1	1	2	1
TCS324.2	Know the working of cryptography techniques.		2	1		1							1	2	1	1
TCS324.3	Analyze the different types of cryptosystems.		2	1		1							1	2	1	1
TCS324.4	Use cryptographic techniques to implement information security protocols.	2			1	2							1	1	2	1
TCS324.5	Apply cryptographic techniques in different applications.		2	1									1	2	1	1
TCS324.6	Develop symmetric and asymmetric key cryptosystems.	2	1									1	2	1	1	1
TCS 324		2.00	1.75	1.00	1.00	1.33	-	-	-	-	-	1.00	1.17	1.50	1.33	1.00

High correlation (3); Medium correlation (2); Low correlation (1), No correlation (-)