GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER I

Name of Department: - **Computer Science and Engineering**

1. Subject Code: **TCS 101**

   Course Title: **Fundamental of computer and Introduction to 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** **SEE** **50**

5. Credits: **3**

6. Semester: **I**

7. Category of Course: **DC**

8. Pre- requisite: Basic Knowledge of Mathematics

9. **Course Outcome**:**	CO1: Learn the concepts of IT and understand the fundamentals of basic building blocks of computer science. CO2: Understand basic data types and syntax of C programming. CO3: Propose solution to problem by using tools like algorithm and flowcharts. CO4: Analyze and select the best possible solution for decision-based problems using decision making skills and develop the aptitude to solve iterative problems using different types of looping statements. CO5: Implement complex problems as a collection of sub problems by applying modularization in applications using functions. CO6: Apply and implement the concept arrays for providing solution to homogenous collection of data types.

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

10. **Details of the Course:**

Sl. No.	**Contents**	**Contact Hours**
1	**UNIT- I** Generation of computers, Computer system memory hierarchy, Input/Output, RAM/ROM, Software & Hardware, Understand bit, byte, KB, MB, GB and their relations to each other, Operating System overview, Computer Networks Overview	**8**

	Algorithms and Flow Charts – Examples of Flow charts for loops and conditional statements	
2	**UNIT- 2** First C program - Hello world, How to open a command prompt on Windows or Linux. How to read and print on screen - printf(),scanf(),getchar(), putchar() Variables and Data types - Variables, Identifiers, data types and sizes, type conversions, difference between declaration and definition of a variable, Constants Life of a C program (Preprocessing, Compilation, Assembly, Linking, Loading, Execution), Compiling from the command line, Macros, Operators – equality and assignment, Compound assignment operators, Increment and decrement operators, Performance comparison between pre and post increment/decrement operators, bitwise operators, Logical Operators, comma operator, precedence and associativity.	**10**
3	**UNIT- III** Conditional statements: if statement, if-else statement, ternary statement or ternary operator, nested if-else statement, switch statement, Difference between performance of if else and switch, Advantages of if else and switch over each other Loops: ‘for’ loops, ‘while’ loops, ‘do while’ loops, entry control and exit control, break and continue, nested loops	**8**
4	**UNIT- IV** Functions: Function prototype, function return type, signature of a function, function arguments, call by value, Function call stack, Recursion v/s Iteration, passing arrays to functions, Storage classes: Automatic, Static, Register, External, Static and Dynamic linking implementation, C program memory (show different areas of C program memory and where different type of variables are stored), scope rules.	**7**
5	**UNIT- V** **Arrays:** Single-dimensional arrays, initializing arrays, computing address of an element in array, character arrays, segmentation fault, bound checking, Searching and Sorting.	**10**
	Total	**43**

11**. Suggested Books:**

SL. No.	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication/Reprint**
	Text Books		
1.	- Peter Prinz, Tony Crawford,”C in a Nutshell”, Oreilly Publishers,	1st	2011
2.	- Peter Norton, “Introduction to computers”, TMH,	6th	2009
	- E.Balagurusamy,”Programming in ANSI C”,McGraw Hill	6th	2015
	Reference Books		
1.	Steve Oualline, “Practical C programming”, Orielly Publishers, 2011.	3rd	2011
2.	Brian W Kernighan, Dennis M Ritcie, ”The C Programming Language”, Prentice Hall, 1988. R3. Herbert Schildt,” C: The Complete Reference”, 4thEdition.TMH, 2000.	2nd	2000
3.	Yashwant Kanetkar,”Let Us C”,BPB Publication	8th	2007

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam / Lab Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER II

Name of Department: - **Computer Science and Engineering**

1. Subject Code: **TCS 201**

   Course Title: **Programming for Problem Solving**

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: **II**

7. Category of Course : **DC**

8. Pre- requisite: Basic Knowledge of Mathematics and Computer Fundamentals, TCS 101

9. **Course Outcome**:**	CO1: Learn and apply concepts of strings and multi-dimensional array for providing solutions to homogenous collection of data types CO2: Propose solution to problem by using tools like algorithm and flowcharts. CO3: Apply the concept of pointers to optimize memory management by overcoming the limitations of arrays. CO4: Process and analyze problems based on heterogeneous collection of data using structures. CO5: Apply concepts of file handling to implement data storage and retrieval tasks. CO6: Implement the basic real life problems using python.

*** Describe the specific knowledge, skills, or competencies the students are expected to acquire or demonstrate.*

10. **Details of the Course:**

Sl. No.	**Contents**	**Contact Hours**
1	**UNIT- I** Multi-Dimensional Arrays- Initializing arrays , row major and column major form of an array, character strings and arrays, Strings – Declaration of strings, Initialization of strings using arrays and pointers, Standard library functions of string.	**6**
2	**UNIT- 2**	**10**

	Pointers –Basic of pointers and addresses, Pointers and arrays, Pointer arithmetic, passing pointers to functions, call by reference. Accessing string through pointers. Dynamic memory management in C - malloc(), calloc(), realloc(), free(), memory leak,Dangling, Void, Null and Wild pointers Structures - Structures, array of structures, structure within structure, union, typedef, self-referential structure, pointer to structure	
3	**UNIT- III** **File Handling** - Opening or creating a file, closing a file, File modes, Reading and writing a text file using getc(), putc(), fprintf() ,fscanf(),fgets(), fputs(), Reading and writing in a binary file, counting lines in a text file, Search in a text file, Random file accessing methods- feof(), fseek(), ftell() and rewind() functions.	**8**
4	**UNIT- IV** **Introduction to Python-** History of Python, Need of Python Programming, Python features, First Python Program, Running python Scripts, Variables, Reserved words, Lines and indentation, Quotations, Comments, Input output. Data Types, Operators and Expressions: Standard Data Types – Numbers, strings, Boolean, Operators – Arithmetic Operators, comparison Operators, assignment Operators, logical Operators, Bitwise Operators.	**10**
5	**UNIT- V** Control flow – if, if-elif-else, for, while, break, continue, pass, range(), nested loops. Functions – Handling functions in Python File Handling – Reading text file, writing text file, copying one file to another	**10**
	Total	**44**

11**. Suggested Books:**

SL. No.	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication /Reprint**
	Text Books		
1.	• Peter Prinz, Tony Crawford,”C in a Nutshell”, Oreilly Publishers,	1st	2011

2.	• Yashwant Kanetkar,"Let Us C",BPB Publication	8th	2007
	Reference Books		
1.	• Steve Oualline, "Practical C programming", Orielly Publishers, 2011.	3rd	2011
2.	• Brian W Kernighan, Dennis M Ritcie,"The C Programming Language",Prentice Hall, 1988. R3. Herbert Schildt," C: The Complete Reference", 4thEdition.TMH, 2000.	2nd	2000
3.	• E.Balagurusamy,"Programming in ANSI C", McGraw Hill	6th	2015

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam / Lab Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS-308** Course Title: **Logic Design & 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** **SEE** **50**

5. Credits: **3**

6. Semester: **3**

7. Category of Course: **DC**

8. Pre-requisite: Basic Electronics Engineering (TEC 101/201)

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: Analyze 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.

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

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, Demultiplexers, 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	8

	table, Flip-flop conversion, Analysis and design of synchronous sequential circuits.	
3	**Unit 3:** Registers, Shift register, Universal shift register, Counters (Ripple & Synchronous): Introduction & Design, Introduction to memory, types of memory, PLD: PAL, PLA, ROM Introduction to Computer Organization & Architecture, Von Neumann and Harvard Architecture, RISC and CISC machines, Evolution of Intel x86 and ARM architecture, Basic measures of computer performance, Amdahl's Law, Little's Law.	10
4	**Unit 4:** Computer Arithmetic (Integer and Floating Point): Representation, Addition, Subtraction, Multiplication and Division. Machine Instruction characteristics, Addressing Modes, Processor structure and operation, Instruction Cycle, Instruction Pipelining: Strategy, performance, Hazards. Control unit operation and microprogrammed control.	10
5	**Unit 5:** Memory hierarchy: Locality and performance, Cache memory: Principles and elements of design, Internal memory, External memory, I/O interface: External devices, I/O modules, Programmed I/O, Interrupt driven I/O, Direct Memory Access. Introduction to alternative architectures.	10
	Total	**48**

11**. Suggested Books:**

S.No	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication / Reprint**
	Textbooks		
1.	M. Morris Mano, Digital Logic and Computer Design, Pearson	1st	2016
2.	W. Stalling, Computer Organization and Architecture, Pearson	11th	2022
	Reference Books		
1.	Charles H. Roth Jr., Fundamentals of Logic Design, Wadsworth Publishing	5th	2005
2.	John P Hayes, Computer Architecture and Organization, McGraw Hill	3rd	2017

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE 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** **25** **SEE 50**
5. Credits: **3**
6. Semester: **III**
7. Category of Course: **DC**
8. Pre-requisite:

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.

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

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	**10**

	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 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**

11**. Suggested Books:**

SL. No.	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication / Reprint**

	Textbooks		
1.	Horowitz and Sahani, “Fundamentals of data Structures”, Galgotia Publication Pvt. Ltd., New Delhi.	2nd	**2008**
2	R. Kruse etal, “Data Structures and Program Design in C”, Pearson Education Asia,	2nd	**2006**
3	A. M. Tenenbaum, “Data Structures using C & C++”, Prentice-Hall of India Pvt. Ltd., New Delhi.	2nd	2014
4	K Loudon, “Mastering Algorithms with C”, Shroff Publisher & Distributors Pvt. Ltd.	1st	2000
5	Bruno R Preiss, “Data Structures and Algorithms with Object Oriented Design Pattern in C++”, Jhon Wiley & Sons, Inc.	1st	1998
6	Adam Drozdek, “Data Structures and Algorithms in C++”, Thomson Asia Pvt	4th	2013

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS 307** 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** **MSE** **25** **SEE** **50**

5. Credits: **3**

6. Semester: **III**

7. Category of Course: **DC**

8. Pre-requisite: **Subject Name with Code**

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.

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

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 passing, inline functions, recursive functions.	10

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**

11**. Suggested Books:**

SL. No.	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication / Reprint**
	Textbooks		
1.	Herbert Schildt,The Complete Reference C++, McGraw Hill	4th	2017
2	Balagurusamy E,Object oriented Programming with C++	8th	2020
	Reference Books		
1.	Paul Deitel and Harvey Deitel, C++: How to Program, Pearson	10th	2016

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TMA 316** 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: **4**

6. Semester: **III**

7. Category of Course: **DC**

8. Pre-requisite: TMA101 Engineering Mathematics-I

   TMA201 Engineering Mathematics-II

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.

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

10. **Details of the Course:**

SL. NO.	**Contents**	**Contact Hours**
1	**Unit 1:** **Relations and Functions:** Review of Sets,	**11**

	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	
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**

11**. Suggested Books:**

SL. No.	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication / Reprint**
	Textbooks		
1.	Kenneth H. Rosen:" Discrete Mathematics and its Applications", , McGraw Hill,.	6th Edition	2007
2	JayantGanguly: "A Treatise on Discrete Mathematical Structures", Sanguine-Pearson,.	2nd	2011
	Reference Books		
1.	D.S. Malik and M.K. Sen: "Discrete Mathematical Structures: Theory and Applications", Thomson,.	2nd	2004
2	Thomas Koshy:" Discrete Mathematics with Applications", Elsevier,.	1st	2005, Reprint 2008
3	Ralph P. Grimaldi:" Discrete and Combinatorial Mathematics" Pearson Education,.	5th	2004

4	S.B.Singh, Jaikishor and Ekata, “Discrete Mathematics”, Khanna Publication,.	3rd	2011

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **PCS-308** Course Title: **Logic Design & Computer Organization Lab**

2. Contact Hours: **L: 0** **T: 1** **P: 2**

3. Examination Duration (Hrs): **Theory 0** **Practical 3**

4. Relative Weight: **CIE 25** **MSE 25** **SEE 50**

5. Credits: **2**

6. Semester: **3**

7. Category of Course: **DC**

8. Pre-requisite: Basic Electronics Lab

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**

11**. Suggested Books:**

SL. No.	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication / Reprint**
	Textbooks		
1.	M. Morris Mano, Michael D. Ciletti, Digital Design: With an Introduction to the Verilog HDL, VHDL, and System Verilog, Pearson	6th	2018
	Reference Books		
1.	John P Hayes, Computer Architecture and Organization, McGraw Hill	3rd	2017

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **PCS 302** Course Title: **Data structures Lab**

2. Contact Hours: **L: 0** **T**: **1** **P**: **2**

3. Examination Duration (Hrs): **Theory** **0** **Practical** **3**

4. Relative Weight: **CIE** **25** **MSE** **25** **SEE** **50**

5. Credits: **2**

6. Semester: **3**

7. Category of Course: **DC**

8. Pre-requisite: **TCS101, TCS 201**

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

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

10. **Details of the Course:**

Sl. No.	**List of problems for which student should develop program and execute in the Laboratory**	**Contact Hours**
16.	**Array :** a) Write a C program to find out the sum all even elements from an array. b) Write a C program to find out union of two given arrays. c) Write a C program to find out intersection of two given arrays. d) Write a C program to store N elements into the array and the reverse the contents of that array.	
17.	a. Write a C program to find element with maximum value from an array.	

	b. Write a C program to create a dynamic array. c. Write a C program to Implementation Stack Using Array.	
18.	a. Write a C program to Implementation queue Using Array. b. Write a C program to convert infix expression into postfix expression. c. Write a C program to evaluate any postfix expression.	
19.	a. Link list: b. Create a Single Linked List with pointers left & right where new nodes are always added after the right. Then user will input a key that should be searched in the linked list & the element having the key value should be deleted & linked list should be updated. If elements is not found then a message “Unsuccessful Search" should be displayed.	
20.	a. Write a program to insert string in linked list in alphabetical order. b. Write a program to search a node from a linked list. c. Doubly Linked List: Write a C program to implement doubly linked list with following operations. **i)Insert ii) delete ii) display.**	
21.	a) Circular linked list: Write a C program to implement Circular. b) linked list with following operations. i)Insert ii) delete ii) display.	
22.	a. Write a C program to implement Stack Using inked List. b. Write a program to implement queue using double pointers. c. Write a C program to implement Queue Using Linked List.	
23.	**Tree and Graphs:** a) Write a C program to create Binary search tree and perform following operations on it. **i)Insert node ii) Delete node iii) Search node**. Execute simple graph traversals algorithms (DFS and BFS) .	
24.	**SORTING TECHNIQUES and SEARCHING TECHNIQUES** a) Write a C program to sort an array using Bubble Sort technique. b) Write a C program to sort an array using selection sort technique. c) Write a C program to sort an array using Insertion sort technique.	
25.	d) Write a C program to sort an array using Merge Sort technique. e) Write a C program to sort an array using Quick Sort technique. f) Write a C program to implement Linear Search. g) Write a C program to implement Binary Search. h) Write a C program to implement Hashing.	
	Total	

11. Suggested Books:

SL. No.	Name of Authors/Books/Publishers	Edition	Year of Publication / Reprint

	Textbooks		
1.	Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, Universities Press	2nd	2014
2.	Data Structures Through C, Yashavant Kanetkar	3rd	**2019**
	Reference Books		
1.	Seymour Lipschutz, Data Structures Schaum's Outlines, McGraw Hill	1st	2014
2.	Data Structures Using C A Practical Approach for Beginers		**2021**

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1.	Subject Code:	**PCS 307**		Course Title:	**OOP WITH C++ LAB**		
2.	Contact Hours:	**L:** 0	**T**: 1	**P**: 2			
3.	Examination Duration (Hrs):		**Theory**		**Practical**		
4.	Relative Weight:	**CIE** 50	**MSE** 25			**SEE** 50	
5.	Credits:	**2**					
6.	Semester:	**III**					
7.	Category of Course:	**DC**					
8.	Pre-requisite:	**PCS 101, PCS 102**					

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

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

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.	

<table>
<tr><td></td><td>Implement a C++ program to read the names of users and number of units consumed and display the charges with names</td><td></td></tr>
<tr><td>2.</td><td>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</td><td></td></tr>
<tr><td>3.</td><td>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</td><td></td></tr>
<tr><td>4.</td><td>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.

<b>Input</b>
First line contains an integer as input. Next line contains space separated integers denoting the elements of the array
<b>Output</b>
In the output you have to print an integer that denotes the maximum special sum

<table>
<tr><td colspan="2"><b>Input/Output Format</b></td></tr>
<tr><td><b>Typical Input</b></td><td><b>Expected Output</b></td></tr>
<tr><td>5
1 3 1 2 5</td><td>8</td></tr>
<tr><td>10
2 1 3 9 2 4 -10 -9 1 3</td><td>9</td></tr>
</table>
</td><td></td></tr>
<tr><td>5.</td><td>Implement a C++ program to demonstrate the concept of data abstraction using the concept of Class and Objects</td><td></td></tr>
<tr><td>6.</td><td>Define a class Hotel in C++ with the following specifications
<b>Private members</b>
• 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,</td><td></td></tr>
</table>

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

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 = $(1/3)\pi r^2h$ Volume of hemisphere = $(2/3)\pi r^3$ Volume of cylinder = πr^2h	
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	

11**. Suggested Books:**

SL. No.	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication / Reprint**
	Textbooks		
1.	Herbert Schildt,"The Complete Reference C++", 4th Edition, Tata McGraw Hill, 2003.	4th	2003
2.	Balagurusamy." Object Oriented Programming with C++",8th Edition, Tata McGraw Hill,2020	8th	2020
	Reference Books		
1.	Paul Deitel and Harvey Deitel," C++: How to Program",10 Edition, Pearson	10th	2016

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS 351** Course Title: **Fundamental of Cloud Computing and Bigdata**

2. Contact Hours: **L: 3** **T**: **P**:

3. Examination Duration (Hrs): **Theory** **Practical**

4. Relative Weight: **CIE 25** **MSE 25** **SEE 50**

5. Credits: **3**

6. Semester: **III**

7. Category of Course: **DE**

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 computing applications. CO4: Analyze the cloud computing services management techniques, providers, anc standards. CO5: Distinguish the cloud computing services using Bigdata and big data analytics CO6: Design and deploy a cloud based web application.

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

10. **Details of the Course:**

Sl. No	**Contents**	**Contact Hours**
1	**Unit 1:** Introduction to Cloud Computing, Vision, History, Evolution, and Characteristics c Cloud Computing (NIST), Characteristic, Advantages and Disadvantages of Clou	9

	Computing, Cloud computing vs. Cluster computing vs. Grid computing, Importanc of Open Standards for digital age technologies.	
2	**Unit 2:** Working of Cloud Computing, Cloud Computing comparison with traditiona computing architecture (client/server), Impact of Networks, Web Development an User Interface (UI) on Cloud computing. Cloud Deployment Models: Public cloud, Private cloud, Hybrid cloud, Communit cloud.	9
3	**Unit 3:** Cloud Service Models: Infrastructure as a Service (IaaS), Platform as a Servic (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, Dat centre physical plant/building, Networking firewalls/security, Data storage in clou 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, busines 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 o 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, Machin learning, Regression analysis, Sentiment analysis, Social network analysis, Differenc between big data and big data analytics. Introduction to Big Data analytics, Data Analysis Techniques: A/B testing, Data fusio and data integration, Data mining, Machine learning, Natural language processin (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 Relationa Database Service (Amazon RDS), Amazon S3, Amazon CloudFront, Amazon Glacier and Amazon SNS. Service Management in Cloud Computing: Service Level Agreements (SLAs), Billin & Accounting. Economics of Cloud Computing: SWOT Analysis and Value Proposition, Genera 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.	9
	Total	**45**

11. **Suggested Books:**

SL. No.	Name of Authors/Books/Publishers	Edition	Year of Publication Reprint
	Textbooks		
1.	Rajkumar Buyya, Cloud Computing Principles and Paradigms Wiley,	1st	2013
2	Kannammal, Fundamentals of Cloud Computing, Cengag Learning,	1st	2015
3	Cloud Computing Bible, Barrie Sosinsky, Wiley-India,	1st	2011
	Reference Books		
1.	Jared Dean, Bigdata Data Mining and Machine Learning Wiley,	1st	2014
2	Vince Reynolds, Bigdata for Beginners, Create spac Independent Publishing Platform,	1st	2016

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS 392** Course Title: **Introduction to Cryptography**

2. Contact Hours: **L: 3** **T**: **P**:

3. Examination Duration (Hrs): **Theory** **Practical**

4. Relative Weight: **CIE 25** **MSE 25** **SEE 50**

5. Credits: **3**

6. Semester: **III**

7. Category of Course: **DE**

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.

** *Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

10. **Details of the Course:**

Sl. No.	**Contents**	**Contact Hours**
1	**Unit – 1:** Introduction: Computer Security Concepts: The OSI SecurityArchitecture, Security Attacks, Security Services, Security Mechanisms, a Model for Network Security, Standards Cryptography fundamentals and terminology; Cryptanalysis and Brute-Force Attack, Fundamental techniques of cryptography Substitution and Transposition; Classical Ciphers; Basics of Steganography.	8
2	**Unit – 2:** Modern Cryptography: Symmetric Encryption and MessageConfidentiality:	9

	Symmetric Encryption Principles, Fiestal structure. Symmetric Block Encryption Algorithms, Simple DES, double DES, Stream Ciphers and RC4, Random and Pseudorandom Numbers.	
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 UsageMathematical 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.	8
	Total	**44**

11**. Suggested Books:**

SL. No.	Name of Authors/Books/Publishers	Edition	Year of Publication / Reprint
	Textbooks		
1.	William Stallings, Network Security Essentials Applications and Standards, ,Pearson Education,	6th	2018
2	William Stallings , Cryptography and Network Security, Pearson Education,	7th	2017
	Reference Books		
1.	Behrouz Forouzan , Cryptography and Network Security, McGraw Hill,	3rd	2015
2	Atul Kahate, "Cryptography and Network Security", McGraw Hill Education,,	3rd	2017

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS 331** 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** **SEE** **50**
5. Credits: **3**
6. Semester: **III**
7. Category of Course: **DE**
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.

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

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**

11**. Suggested Books:**

SL. No.	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication / Reprint**
	Textbooks		
1.	Daniel Minoli, “Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications”, ISBN: 978-1-118-47347-4, Willy Publications	1st	**2013**
2	Bernd Scholz-Reiter, Florian Michahelles, “Architecting the Internet of Things”, ISBN 978-3- 642-19156-5 e-ISBN 978-3-642-19157-2, Springer	1st	**2011**
3	Parikshit N. Mahalle&Poonam N. Railkar, “Identity Management for Internet of Things”, River Publishers, ISBN: 978-87-93102-90-3 (Hard Copy), 978-87-93102-91-0 (ebook).	1st	2015

	Reference Books		
1.	HakimaChaouchi, "The Internet of Things Connecting Objects to the Web" ISBN : 978-1- 84821-140-7, Willy Publications	1st	2010
2	Olivier Hersent, David Boswarthick, Omar Elloumi, The Internet of Things: Key Applications and Protocols, ISBN: 978-1-119-99435-0, 2 nd Edition, Willy Publications	1st	**2015**
3	Daniel Kellmereit, Daniel Obodovski, "The Silent Intelligence: The Internet of Things",. Publisher: Lightning Source Inc; ISBN-10: 0989973700, ISBN-13: 978-0989973700.	1st	**2014**
4	Fang Zhaho, Leonidas Guibas, "Wireless Sensor Network: An information processing approach", Elsevier, ISBN: 978-81-8147-642-5.	1st	**2055**

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS- 341** Course Title: **Python Programming for Computing**

2. Contact Hours: **L:** **T**: **0** **P**: **0**

3. Examination Duration (Hrs): **Theory** **3** **Practical** **0**

4. Relative Weight: **CIE** **25** **MSE** **25** **SEE** **50**

5. Credits: **4**

6. Semester: **3rd**

7. Category of Course: **DE**

8. Pre-requisite: TCS 101, TCS 201

9. **Course Outcome**:**	After completion of the course the students will be able to**:** **CO1:** Describe the principles of structured programming and be able to describe, design, implement, and test structured programs using currently accepted methodology. **CO2:** Explain what an algorithm is and its importance in computer programming. **CO3:** Recognize and construct common programming idioms: variables, loop, branch, subroutine, and input/output. **CO4:** Define and demonstrate the use of the built-in data structures 'list' and 'dictionary'. **CO5:** Apply idioms to common problems such as text manipulation, web page building, and working with large sets of numbers. **CO6:** Design and implement a program to solve a real-world problem using the language idioms, data structures, and standard library.

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

10. **Details of the Course:**

Sl. No.	**Contents**	**Contact Hours**
1	**Unit 1:** **Introduction:** Introduction to Python: Importance of Python, Installing and working with Python in Windows, Linux and Mac, Using Python as calculator, Comments, how to define main function in Python. The concept of data types - Variables, Arithmetic Operators and Expressions **String manipulations** - Subscript Operator, Indexing, slicing a string, Converting strings to numbers and vice versa, split function. Control flow - if statements, for and while loops, nested loops, Short-	9

	circuit (lazy evaluation), range () function, break and continue statements, pass statements	
2	**Unit 2:** **Data Structures:** Lists - Basic list operations, Replacing, inserting, removing an element; Searching and sorting a list, Methods of list objects, using lists as Stacks and Queues, List, and nested list Comprehensions. Tuple, Sets, Difference between list and tuple Dictionary - adding and removing keys, accessing, and replacing values, traversing dictionaries	9
3	**Unit 3** **File and Exception Handling in Python** Reading config files in python, Writing log files in python, Understanding read functions, read (), readline() and readlines(), Understanding write functions, write() and writelines(), Manipulating file pointer using seek, Programming using file operations Exception Handling - Exceptions, why use exceptions, raising an exception, try and except, try, except and else clause; try and finally	11
4	**Unit 4:** Regular Expressions and Python Packages Regular Expressions - re module, searching a string (match and search), Finding a string (findall), Break string into substrings (split), Replace part of a string (sub) Python packages: Simple programs using the built-in functions of packages matplotlib, NumPy, Pandas	9
5	**Unit 5:** **Python Functions and OOP Concepts** Python functions and modules - OS and SYS modules, defining python functions, calling a function, function arguments, Lambda, and map function, Importing python module. Classes and OOP - Class definition syntax, objects, class, and instance variables, Inheritance and multiple inheritance, Polymorphism, Overloading, Overriding, Data Hiding	10
	Total	**48**

11. Suggested Books:

SL. No.	Name of Authors/Books/Publishers	Edition	Year of Publication / Reprint
	Textbooks		
1.	Kenneth A. Lambert, "The Fundamentals of Python: First Programs", Cengage Learning.,	1st	**2011**
2.	Think Python: How to think like a Computer Scientist	2nd	**2015**
3.	Python Programming using Problem Solving Approach	1st	**2017**

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE UNIVERSITY), DEHRADUN

SEMESTER III

Name of Department: - Computer Science and Engineering

1. Subject Code: **TCS 332** Course Title: Fundamental of Information Security and Blockchain

2. Contact Hours: **L:** **3** **T**: **P**:

3. Examination Duration (Hrs): **Theory** **Practical**

4. Relative Weight: **CIE** **25** **MSE** **25** **SEE** **50**

5. Credits: **4**

6. Semester: **III**

7. Category of Course: **DE**

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

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate.*

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 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	9
2	**Unit – 2:** **Linux Basics and Scripting for Information Security**	8

	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	
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**

11**. Suggested Books:**

SL. No.	Name of Authors/Books/Publishers	Edition	Year of Publication / Reprint
	Textbooks		
1.	Georgia Weidman,“Penetration Testing: A Hands-on Introduction to Hacking”, No Starch Press,	1st	2020
2	George Icahn, “Blockchain: the complete guide to understanding blockchain technology”,	1st	2020
3	Antony lewis, “The basics of bitcoins and blockchains: an introduction to cryptocurrencies and the technology that powers them”	3rd	2020
	Reference Books		
1.	Andreas M. Antonopoulos, “Mastering Bitcoin: unlocking digital cryptocurrencies”, O'Reilly Media,	2nd	2017
2	Roger Wattenhofer, “Distributed Ledger Tehnology, The science of the Blockchain”, Inverted Forest Publishing,	2nd	2017
	Antonopoulos, Andreas M. and Wood, Gavin. Mastering Ethereum. O’Reilly Media,.	2nd	2018

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam

GRAPHIC ERA (DEEMED TO BE 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** **1** **0**

3. Examination Duration (Hrs): **Theory** **3** **tical** **0**

4. Relative Weight: **CIE** **25** **MSE** **25** **50**

5. Credits: **4**

6. Semester: **III**

7. Category of Course: **DC**

8. Pre-requisite: **TMA 101 Engineering Mathemaics I**, **TMA 201 Engineering Mathemaics 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.

*** Describe the specific knowledge, skills or competencies the students are expected to acquire or demonstrate*

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**

11**. Suggested Books:**

SL. No.	**Name of Authors/Books/Publishers**	**Edition**	**Year of Publication / Reprint**
	Textbooks		
1.	Marc Peter Deisenroth , A. Aldo Faisal, Cheng Soon Ong, MATHEMATICS FOR MACHINE LEARNING, Cambridge University Press	1st	2020
2.	Jay Dawani, Hands-On Mathematics for Deep Learning: Build a solid mathematical foundation for training efficient deep neural networks, Packt Publishing Limited	1st	2020
	Reference Books		
1.	Tamoghna Ghosh , Shravan Kumar Belagal Math, Practical Mathematics for AI and Deep Learning, BPB Publications	1st	2022

12.	**Mode of Evaluation**	Test / Quiz / Assignment / Mid Term Exam / End Term Exam