Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

L	T	P	Credits
3	0	0	3

Course Name : **Data Structures with C**

Course Code : **TCS302**

Semester : **III**

Pre-requisite : **Fundamentals of Computer & Introduction to Programming (TCS101), Programming for problem solving (TCS201).**

Course Learning Objectives:

- To provide the students; an in-depth understanding of structure and implementation of the common data structures used in computer science.
- To provide the ability to solve problems by choosing and applying the appropriate data structures.
- To understand concepts about stacks, queues, lists, trees and graphs and its real-world applications.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Apply appropriate operations to solve computational problems based on the fundamental concepts of data structures, algorithm analysis, arrays, stacks, and recursion.	L2 – Understand	Understanding, Analytical Thinking
CO2	Analyze queues and various linked list structures and their operations for efficient memory management and data manipulation.	L3 – Apply	Programming, Problem Solving
CO3	Analyze tree-based data structures, searching techniques to optimize data storage and retrieval.	L5 – Create	Programming, Design, Implementation
CO4	Apply sorting algorithms and binary search tree operations for efficient searching and sorting applications.	L2 – Analyze	Algorithmic Analysis, Critical Thinking
CO5	Apply graph traversal algorithms, disjoint sets, and minimum spanning tree algorithms for complex data processing applications.	L4 – Evaluate	Data Structure Design, Problem Solving

---

Details of the Course:

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Unit 1: **Contact Hours/Teaching Hours: 10**

Introduction: Basic Terminology, Pointer, Structure, Algorithm Complexity and Time-Space trade-off, Abstract Data Type (ADT).
Arrays: Single and Multidimensional Arrays, address calculation, application of arrays, Sparse Matrices.
Stacks: Array Implementation of stack, Operations Associated with Stacks, Application of stack: Conversion of Infix to Prefix and Postfix Expressions, Evaluation of postfix expression using stack, Recursion and its types.

Self-Learning Hours = 8

Students shall practice writing algorithms, analyze time and space complexity, and implement array and stack, its associated operations in C.

Team Activity Hours = 2

Group discussion for selection of appropriate data structures for real-world application problems.

Unit 2: **Contact Hours/Teaching Hours: 10**

Queues: Array implementation of queues, Operations Associated with Queue, Circular queue, Dequeue, and Priority Queue.
Linked list: Singly Linked Lists, Two-way Header List, Linked Representation and Implementation of Stack and queue, doubly linked list, Circular Linked list, Linked List in Array, Polynomial representation and addition, Generalized linked list.

Self-Learning Hours = 8

Students shall implement queue and link list and associated operations with real world application problems in C.

Team Activity Hours = 2

Group-based activity on solving practical problems using link list & queue.

Unit 3:

Contact Hours/Teaching Hours: 9

Trees: Basic terminology, Binary Trees, Binary tree representation, algebraic Expressions, Complete Binary Tree, Binary Heaps. Extended Binary Trees, Array and Linked Representation of Binary trees, Threaded Binary trees. Traversing in Binary and Threaded Binary trees, Huffman algorithm & Huffman tree.
Binary Search Trees: Binary Search Tree (BST), Insertion and Deletion in BST, Complexity of Search Algorithm, Path Length, AVL Trees.

Self-Learning Hours = 6

Students shall implement binary tree, Hash table and perform associated operations on these data structures.

Team Activity Hours = 2

Group programming activity to develop applications using Huffman algorithm & Hash table.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Unit 4: Searching and Sorting

Contact Hours/Teaching Hours: 9

Sorting: Types and its Properties, Insertion Sort, Bubble Sorting, Quick Sort, Two-Way Merge Sort, Heap Sort

Searching and Hashing: Sequential search, binary search, comparison and analysis, Hash Table, Hash Functions, Collision Resolution Strategies, Hash Table Implementation.

Self-Learning Hours = 6

Students shall implement sorting techniques, and Binary search tree and analyze their time complexity.

Team Activity Hours = 2

Group activity on comparing the performance of different sorting algorithms using sample datasets.

Unit 5:

Contact Hours/Teaching Hours: 08

File Structures: File Organization, Organization of records into Blocks, Indexing, B and B+ Tree index Files.

Graph: Traversal (DFS, BFS), Disjoint Sets: Union-Find Data Structure, Path Compression, Union by Rank/Size. Minimum spanning tree (Kruskal's & Prim's), Shortest Path: Dijkstra, Bellman-ford and Floyd Warshall

Self-Learning Hours = 6

Students shall implement Graph and spanning tree and its associated operations.

Team Activity Hours = 2

Group activity & discussions to provide solution of the real-world problems based on graphs.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO01	3	2	1	1	2	-	-	-	1	1	1		2	2	1
CO02	2	3	3	2	2	-	-	1	2	1	1	-	2	2	2
CO03	3	3	3	2	2	-	-	1	2	1	1	-	3	3	2
CO04	3	3	3	3	2	-	-	1	2	1	2	1	3	3	2
CO05	3	3	3	3	3	-	1	1	2	2	2	1	3	3	3
Avg.	2.8	2.8	2.6	2.1	0.4	0	0.5	0.8	1.8	1.2	1.4	0.4	2.6	2.6	2

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

---

Text Books:

1. Horowitz and Sahani, (2008). *Fundamentals of data Structures,* 2nd Edition, Galgotia Publication Pvt. Ltd., New Delhi.
2. R. Kruse (2006), *Data Structures and Program Design in C*, 2nd Edition, Pearson Education Asia.

3. A. M. Tenenbaum (2014), *Data Structures using C & C++*, 2nd Edition, Prentice-Hall of India Pvt. Ltd., New Delhi.

Reference Books:

1. K Loudon, (2000), *Data Structures and Algorithms with Object Oriented Design Pattern in C++*, 1st Edition, Shroff Publisher & Distributors Pvt. Ltd.
2. Bruno R Preiss (1998), Data Structures and Algorithms in C++,1st Edition, Jhon Wiley & Sons, Inc.

Online Resources:

1. **GeeksforGeeks – Data Structures**
   https://www.geeksforgeeks.org/data-structures/
2. **NPTEL – Programming, Data Structures and Algorithms**
   https://nptel.ac.in/
3. **Programming Problems**
   https://leetcode.com/

Graphic Era HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Object Oriented Programming with C++**
Course Code	:	**TCS307**
Semester	:	**III**
Pre-requisite	:	**Fundamental of Computer & Introduction to Programming (TCS101), Programming for problem solving (TCS201).**

L	T	P	Credits
3	0	0	3

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Define the fundamental concepts of object-oriented programming and identify the basic components of C++ programming, including data types, operators, arrays, strings, pointers, functions, header files, and namespaces.	**L1 – Remember**	Recall and Identify
CO2	Explain the concepts of classes, objects, constructors, destructors, friend functions, dynamic objects, copy constructors, this pointer, and operator overloading in C++.	**L2 – Understand**	Explain and Interpret
CO3	Apply the concepts of inheritance to develop C++ programs using appropriate access specifiers, constructors, destructors, and virtual base classes.	**L3 – Apply**	Implement and Execute
CO4	Differentiate between compile-time and run-time polymorphism and examine the implementation of function overloading, function overriding, virtual functions, abstract classes, virtual destructors, and dynamic binding in C++ applications.	**L4 – Analyze**	Differentiate and Examine
CO5	Apply file handling, exception handling, templates, and Standard Template Library (STL) components to develop efficient C++ programs.	**L3 – Apply**	Implement and Solve Problems

---

Details of the Course:

Unit 1: Introduction to Object-Oriented Programming **Contact Hours/Teaching Hours: 10**

Need for Object-Oriented Programming, Procedure-Oriented vs Object-Oriented Programming, Overview of C++, Structure of C++ Program, Header Files and Namespaces, Data Types, Variables, Operators, Expressions and Statements, Arrays and Strings, Pointers, User-defined Functions, Argument Passing, Inline Functions, Recursive Functions.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Self-Learning Hours = 6

Students shall practice writing C++ programs using data types, operators, arrays, strings, pointers, user-defined functions, inline functions, and recursive functions.

Team Activity Hours = 3

Group discussion and implementation of simple C++ programs demonstrating procedural and object-oriented programming approaches.

Unit 2: Classes, Objects and Operator Overloading **Contact Hours/Teaching Hours: 09**

Classes and Objects, Class Specification, Scope Resolution Operator, Access Specifiers, Member Functions, Data Hiding, Constructors, Parameterized Constructors, Copy Constructors, Destructors, Static Data Members, Friend Functions, Passing and Returning Objects, Arrays of Objects, Dynamic Objects, Pointers to Objects, This Pointer.

Operator Overloading: Fundamentals, Unary and Binary Operator Overloading, Operator Overloading using Friend Functions, Overloading of +, -, Pre-increment (++), Post-increment (++), << and >> Operators.

Self-Learning Hours = 6

Students shall implement programs using constructors, destructors, friend functions, copy constructors, this pointer, dynamic objects, and operator overloading.

Team Activity Hours = 3

Group programming activity to develop class-based applications demonstrating operator overloading.

Unit 3: Inheritance **Contact Hours/Teaching Hours: 09**

Need for Inheritance, Base Class and Derived Class, Types of Inheritance (Single, Multiple, Multilevel, Hierarchical and Hybrid), Public, Private and Protected Inheritance, Constructors and Destructors in Inheritance, Passing Parameters to Base Class Constructors, Virtual Base Classes.

Self-Learning Hours = 6

Students shall implement different inheritance models and study constructor and destructor execution order.

Team Activity Hours = 3

Group activity to design reusable applications using different inheritance mechanisms.

Unit 4: Polymorphism and Virtual Functions **Contact Hours/Teaching Hours: 09**

Polymorphism, Compile-time and Run-time Polymorphism, Function Overloading, Function Overriding, Virtual Functions, Calling Virtual Functions through Base Class Pointers, Pure Virtual Functions, Abstract Classes, Virtual Destructors, Early Binding and Late Binding.

Self-Learning Hours = 6

Students shall develop programs demonstrating function overloading, overriding, virtual functions, and abstract classes.

Team Activity Hours = 3

Case study on the application of runtime polymorphism in software development.

Unit 5: File Handling, Exception Handling and STL **Contact Hours/Teaching Hours: 09**

C++ Stream Classes, I/O Manipulators, File Streams (fstream), File Handling Operations, Function Templates, Exception Handling Fundamentals, Throwing and Catching Exceptions, Re-throwing Exceptions, Standard Template Library (STL), Containers, Vectors, Lists, Maps, Iterators and Algorithms.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Self-Learning Hours = 6

Students shall implement programs using file handling, templates, exception handling, and STL containers.

Team Activity Hours = 3

Group activity to develop mini-projects using STL containers, exception handling, and file management.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO01	3	2	1	-	2	-	-	-	-	1	-	2	2	2	1
CO02	3	2	2	-	2	-	-	-	-	1	-	2	2	2	1
CO03	3	3	2	-	2	-	-	-	-	1	-	2	3	2	1
CO04	3	3	2	1	2	-	-	-	-	1	-	2	3	2	2
CO05	3	2	3	1	3	-	-	-	-	1	-	2	3	3	2
Avg.	3	2.4	2	1	2.2	-	-	-	-	1	-	2	2.6	2.2	1.4

Text Books:

1. Herbert Schildt (2017), The Complete Reference C++, 4th Edition, McGraw Hill.
2. Balagurusamy E (2020), Object oriented Programming with C++, 8th Edition, McGraw Hill.

Reference Books:

1. Paul Deitel and Harvey Deitel (2016), C++: How to Program, 10th Edition, Pearson.

Online Resources:

1. **NPTEL-Fundamentals of Object Oriented Programming**
   nptel.ac.in/courses/106107574
2. **GeeksforGeeks – Object Oriented Programming in C++**
   Object Oriented Programming in C++ - GeeksforGeeks

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Python Programming**
Course Code	:	**TCS346**
Semester	:	**III**
Pre-requisite	:	**Fundamental of Computer & Introduction to Programming (TCS101), Programming for Problem Solving (TCS201)**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To understand Python syntax, data types, operators, input/output operations, control statements, functions, modules, recursion, and exception handling.
- To apply Python data structures such as lists, tuples, dictionaries, sets, regular expressions, and file handling for basic data processing.
- To demonstrate the basic concepts of object-oriented programming including classes, objects, inheritance, polymorphism, encapsulation, and abstraction.
- To use Python libraries such as NumPy, Pandas, Matplotlib, and Seaborn for numerical computing, data handling, and visualization.
- To analyze datasets using data cleaning techniques, exploratory data analysis, summary statistics, correlation, visualization, and outlier detection.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Explain fundamental Python programming concepts including syntax, data types, control structures, functions, modules, and exception handling.	L2 – Understand	Employability Skill – Foundational Programming Literacy
CO2	Apply Python programming constructs and data structures to solve computational problems and process data effectively.	L3 – Apply	Employability Skill – Problem-Solving and Data Handling
CO3	Develop modular and reusable Python programs using functions, file handling, regular expressions, and object-oriented programming concepts.	L3 – Apply	Employability Skill – Software Development
CO4	Use Python libraries and modern programming tools for numerical computation, data manipulation, visualization, and exploratory analysis.	L3 – Apply	Employability Skill – Data Analytics and Visualization

CO5	Analyze structured datasets to identify patterns, relationships, outliers, and insights using statistical and visualization techniques..	L4 – Analyze	Employability Skill – Analytical Thinking and Data Interpretation

---

Details of the Course:

Unit 1: Python Basics and Functions **Contact Hours/Teaching Hours: 9**

Syntax and Semantic Basics: Data types: strings and string handling, integers, floats, Variable assignments and expressions, operators, Basic input/output operations, control statements

Defining and calling functions: Function parameters and return values- Using built-in modules, Creating and using custom modules, recursive functions, Exception handling basics.

Self-Learning Hours = 6

Students shall practice basic Python programs using variables, operators, input/output statements, conditional statements, loops, functions, recursion, modules, and exception handling.

Team Activity Hours = 9

Group activity on designing simple algorithms for real-life problems and implementing them using Python.

Unit 2: Data Structures, Regular Expressions and File Handling

Contact Hours/Teaching Hours: 08

Lists: creation, indexing, slicing, and methods. **Tuples:** usage and when to use. **Dictionaries:** creating, accessing, and manipulating. **Sets:** Creation and Operations. **Regular Expressions:** usage and applications, **Reading from and writing to files:** text and binary files

Self-Learning Hours = 6

Students shall practice small programs using lists, tuples, dictionaries, sets, regular expressions, and file handling.

Team Activity Hours = 3

Group-based problem-solving activity on data storage and manipulation using Python data structures.

Unit 3: Object-Oriented Programming using Python

Contact Hours/Teaching Hours: 10

Introduction to classes and objects, Attributes and methods, Inheritance: extending classes, Polymorphism: using a unified interface, Encapsulation: private and public members, Abstraction: why to use abstraction

Self-Learning Hours = 6

Students shall develop a simple class-based Python program such as a student record system, bank account system, library system, employee management system, or product inventory system.

Team Activity Hours = 3

Group discussion and demonstration on real-life examples of object-oriented programming concepts such as inheritance, polymorphism, encapsulation, and abstraction.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Unit 4: Python Libraries for Numerical Computing and Visualization

Contact Hours/Teaching Hours: 10

NumPy: arrays, array operations, statistical operations, Linear algebra operations, slicing, indexing, and reshaping
Pandas: DataFrame operations, indexing, merging, grouping
Matplotlib: Basic plotting, figures, and axes
Seaborn: Statistical data visualization

Self-Learning Hours = 6

Students shall explore NumPy and Pandas operations on a small dataset and create basic plots using Matplotlib and Seaborn.

Team Activity Hours = 3

Group activity on data visualization using bar charts, line charts, histograms, scatter plots, and box plots.

Unit 5: Data Cleaning and Exploratory Data Analysis

Contact Hours/Teaching Hours: 08

Data cleaning techniques, Exploratory data analysis (EDA): summary statistics, correlation, Visualization techniques: histograms, scatter plots, box plots, outliers

Self-Learning Hours = 6

Students shall perform basic data cleaning and exploratory data analysis on a small dataset and prepare a brief analytical report.

Team Activity Hours = 3

Case-study presentation on interpreting data using summary statistics, correlation, visualization, and outlier detection.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS346.1	2	1	-	-	2	-	-	-	-	1	-	2	1	2	1
TCS346.2	2	3	1	-	2	-	-	-	1	1	1	2	2	3	1
TCS346.3	2	2	3	-	2	-	-	-	1	1	1	2	3	2	1
TCS346.4	2	2	1	2	3	-	-	-	1	2	1	2	2	3	2
TCS346.5	2	3	1	3	3	-	-	-	1	2	1	2	2	3	2
Avg.	2	2.2	1.2	2.5	2.4	-	-	-	1	1.4	1	2	2	2.6	1.4

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

Text Books:

1. Let Us Python, Yahwant Kanetkar, BPB, India, 5th Edition, 2023.
2. Python Crash Course: A Hands-On, Project-Based Introduction to Programming, Eric Matthes, No Starch Press, USA, 3rd Edition, 2023.
3. Learning Python, Mark Lutz, O'Reilly Media, USA, 5th Edition, 2013.
4. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython, Wes McKinney, O'Reilly Media, USA, 2nd Edition, 2017

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Reference Books:

1. Automate the Boring Stuff with Python, Al Sweigart, No Starch Press, 2nd Edition, 2020.
2. Think Python, Allen B. Downey, O'Reilly Media, 2nd Edition, 2015.
3. Core Python Programming, Wesley J. Chun, Prentice Hall, 2nd Edition, 2006.
4. Programming in Python 3, Mark Summerfield, Addison-Wesley Professional, 2nd Edition, 2009.
5. Python for Data Analysis, Wes McKinney, O'Reilly Media, 3rd Edition, 2022.

Online Resources:

1. Official Python Documentation
   https://docs.python.org/
2. Official NumPy Documentation
   https://numpy.org/doc/
3. Official Pandas Documentation
   https://pandas.pydata.org/docs/
4. Official Matplotlib Documentation
   https://matplotlib.org/stable/index.html
5. Official Seaborn Documentation
   https://seaborn.pydata.org/

Graphic Era HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Responsible and Explainable AI**
Course Code	:	**TCS349**
Semester	:	**III**
Pre-requisite	:	**Fundamentals of AI and ML.**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

CLO01	Identify the fundamental ethical dilemmas in AI and classify data, algorithmic, and interactional biases in AI systems.
CLO02	Describe tools for achieving AI transparency and apply accountability frameworks to balance security and privacy in AI implementations.
CLO03	Examine the long-term impact of AI on employment, economy, and public policy through case-based discussion.
CLO04	Critically assess the strengths and limitations of feature importance, perturbation analysis, and rule-based explanation techniques.
CLO05	Construct heatmap and partial-dependence-plot visualizations and propose practical XAI applications for finance, healthcare, or autonomous systems.

.

Course Outcomes:

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

CO01	Explain the fundamental ethical principles, responsible AI practices, and societal impacts associated with the development and deployment of Artificial Intelligence systems.
CO02	Apply appropriate techniques to detect, measure, and mitigate bias in AI models for engineering applications while ensuring fairness and accountability.
CO03	Analyze transparency, accountability, privacy, and governance mechanisms to evaluate responsible AI systems in real-world scenarios
CO04	Evaluate Explainable Artificial Intelligence (XAI) methods, including local and global explanation techniques, for interpreting machine learning models across different application domains.
CO05	Design responsible and explainable AI solutions by integrating ethical principles, fairness, transparency, accountability, and explainability into the AI development lifecycle for engineering applications.

Details of the Course:

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Unit 1: **Contact Hours/Teaching Hours: 9**

UNIT:1 Introduction to AI Ethics, Bias in AI Systems, Fairness in AI

Fundamentals of ethics in AI, Importance of ethical AI development, Overview of common ethical dilemmas in AI, Types of biases in AI (data, algorithmic, interactional), Detecting and measuring bias, Mitigation strategies for reducing bias, Defining fairness in AI contexts, Fairness metrics and their trade-offs- Designing fair AI systems, Imminent risks from AI Models, Long-term risks from AI Models

Self-Learning Hours = 4

Students shall understand the concepts of ethics and biasness in AI.

Unit 2: **Contact Hours/Teaching Hours: 9**

Transparency and Accountability, AI in Surveillance and Privacy

Principles of RAI, Importance of transparency in AI, Tools and methods for achieving transparency, Accountability frameworks and regulatory compliance, AI applications in surveillance technologies, Privacy concerns and data protection laws, Balancing security and privacy in AI implementations

Self-Learning Hours = 4

Students shall use the various tools used in industry for AI.

Unit 3:

Contact Hours/Teaching Hours: 7

Societal Impacts of AI

AI's impact on employment and economy, AI in governance and public policy, Long-term implications of AI on society.

Unit 4: Searching and Sorting

Contact Hours/Teaching Hours: 9

Explainable Artificial Intelligence (XAI) Fundamentals

Introduction to Explainable AI, Need and importance of explainability in AI, Overview of XAI methods, Local Explanation Techniques, LIME, SHAP, Global Explanation Techniques, Feature importance, Decision trees as explanatory models- Rule-based explanations.

Self-Learning Hours = 6

Student shall implement the LIME, SHAP techniques.

Unit 5:

Contact Hours/Teaching Hours: 08

Visualization and Applications of Explainable AI

Visualization of feature effects and model decisions, Use of heatmaps and partial dependence plots- Network and layer activation visualizations in deep models

XAI in finance (credit scoring, fraud detection), XAI in healthcare (diagnostic systems, treatment recommendations), XAI in autonomous systems (vehicle control systems, robotics)

Self-Learning Hours = 6

Students shall implement various visualization methods using python.

Team Activity Hours = 2

Group activity & discussions to provide solution of the real-world problems like fraud detection, desiase detection etc .

Course Articulation Matrix

COs / POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	–	–	–	3	-	2	–	1	–	2	1	2	3
CO2	2	3	2	2	3	2	1	3	–	–	–	2	2	3	2
CO3	-	3	2	3	2	3	2	3	–	-	–	2	2	2	3
CO4	2	3	-	3	3	2	–	2	–	-	–	-	2	3	3
CO5	2	-	3	2	3	2	-	3	-	-	-	3	3	3	3

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

---

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year
Stuart Russell, Peter Norvig	Artificial Intelligence: A Modern Approach	3rd Edition	Pearson, USA	2009
Michael Munn, David Pitman	Explainable AI for Practitioners: Designing and Implementing Explainable ML Solutions	1st Edition	O'Reilly, India	2022
John Danaher	Ethics of Artificial Intelligence	2nd Edition	Oxford University Press, USA	2022

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
Denis Rothman	Hands-On Explainable AI (XAI) with Python	1st Edition	Packt Publishing, USA	2020
James Manyika, Michael Chui, and Michael Osborne	The Ethics of Artificial Intelligence	1st Edition	McKinsey Global Institute, USA	2018

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Fundamental of Cloud Computing and Big Data**
Course Code	:	**TCS 351**
Semester	:	**III**
Pre-requisite	:	**NA**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To introduce the fundamental concepts, characteristics, architecture, deployment models, and service models of cloud computing, and compare cloud computing with traditional, cluster, and grid computing environments.
- To develop an understanding of cloud computing technologies, virtualization, networking, and Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) through leading cloud platforms and case studies.
- To familiarize students with Big Data concepts, architectures, analytics techniques, machine learning approaches, and their applications in solving real-world problems.
- To provide practical knowledge of Amazon Web Services (AWS), including core cloud services, cloud security, disaster recovery, service management, and cloud-based application deployment.
- To enable students to analyze cloud computing economics, security, compliance, risks, and best practices while designing and deploying scalable cloud-based applications.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Identify the importance of cloud computing services for the digital Technologies.	**L2 – Understand**	Understanding of cloud computing concepts, knowledge of cloud architectures, problem-solving, and critical thinking.
CO2	Differentiate the services and deployment models of cloud computing.	**L3 – Apply**	Virtualization skills, cloud platform management, networking fundamentals, technical analysis, cloud service

			selection, and system administration.
CO3	Evaluate the case studies of the different types of cloud computing applications.	**L3 – Apply**	Data analysis, Big Data processing, machine learning fundamentals, analytical reasoning, data-driven decision-making, and problem-solving skills.
CO4	Analyse the cloud computing services management techniques, providers, and standards.	**L4 – Analyze**	AWS service management, cloud deployment, security implementation, disaster recovery planning, DevOps awareness, and hands-on cloud operations.
CO5	Design and deploy a cloud-based web application.	**L3 – Apply**	security management, compliance evaluation, cloud solution design, project implementation, strategic thinking, and professional decision-making.

Details of the Course:

Unit 1: **Contact Hours/Teaching Hours: 09**
Introduction to Cloud Computing: Vision, History, Evolution, and Characteristics (NIST), Advantages and Disadvantages of Cloud Computing, Cloud computing vs. Cluster computing vs. Grid computing, Importance of Open Source and Open Standards.

Unit 2: **Contact Hours/Teaching Hours: 09**
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.

Unit 3: **Contact Hours/Teaching Hours: 09**
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 management, Storage, Networking, Virtual Machine (resource) provisioning and manageability, Networking firewalls/security, Data storage in cloud computing (storage as a service), Amazon Elastic Compute Cloud (EC2), Open Stack, Case Study of IaaS. Platform as a

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Service (PaaS): PaaS definition, Service Oriented Architecture (SOA), Development tools, database management, Google App Engine, Microsoft Azure, Case Study of PaaS. Software as a Service (SaaS): SaaS definition, Web services, Web 2.0 & Web 3.0, Case Study of SaaS.

Unit 4: **Contact Hours/Teaching Hours: 09**
Introduction to Big Data, Characteristics, Architectures, Technologies, Applications, Advantages and Disadvantages of Big Data, Tools and Techniques applied in Big Data: Difference between big data and big data analytics. Introduction to Big Data analytics, Data fusion and data integration, Data mining, Machine learning, Natural language processing (NLP), Case study of Big Data.

Unit 5: **Contact Hours/Teaching Hours: 09**
AWS Services: Amazon Lambda, Amazon Relational Database Service (Amazon RDS), Amazon S3, Amazon CloudFront, Amazon Glacier, Amazon SNS, Route 53 Service Management in Cloud Computing: Service Level Agreements (SLAs), Billing & Accounting, Compliance Economics of Cloud Computing: SWOT Analysis and Value Proposition, General Cloud Computing Risks. Hands-on project.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO01	1	-	-	-	-	-	-	-	-	2	1	-	-	-	1
CO02	1	2	3	-	2	-	2	2	2	1	2	1	1	1	2
CO03	1	3	-	-	1	1	-	1	1	1	1	1	-	-	2
CO04	1	2	-	3	1	2	-	1	1	1	1	-	-	-	1
CO05	1	2	3	1	2	1	2	1	1	1	2	1	-	1	2
Avg.	**1**	**2.25**	**3**	**2**	**1.5**	**1.3**	**2**	**1.25**	**1.25**	**1.5**	**1.4**	**1**	**1**	**1**	**2**

Text Books:
1. Rajkumar Buyya "Cloud Computing Principles and Paradigms 1st Edition Wiley 2013
2. **Rajkumar Buyya et al.** *Mastering Cloud Computing: Foundations and Applications Programming* 1st Edition *McGraw-Hill Education* **2021.**
3. **Arshdeep Bahga & Vijay Madisetti** *Cloud Computing: A Hands-On Approach* 2nd Edition *Universities Press* **2022**

Reference Books:
1. Bill Wilder *Cloud Architecture Patterns: Best Practices for Cloud-Based Software* 1st Edition O'Reilly Media 2020.
2. Mark Wilkins *Learning Amazon Web Services (AWS): A Hands-On Guide* 2nd Edition Pearson 2022.

Online Resources:
1. **GeeksforGeeks – Data Structures**
   https://www.geeksforgeeks.org/fundamental of cloud computing/
2. NPTEL: nptel.ac.in/courses/106105167
3. w3schools: AWS Cloud Computing

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Advanced Probability for AI**
Course Code	:	**TCS368**
Semester	:	**III**
Pre-requisite	:	**Mathematics for AI-I, Mathematics for AI-II**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To develop an understanding of Bayesian modelling for reasoning under uncertainty and probabilistic decision-making.
- To employ information-theoretic principles to measure uncertainty and improve the learning process of probabilistic models.
- To construct probabilistic graphical and sequential models to represent complex dependencies in data.
- Utilize inference and sampling algorithms for efficient parameter estimation and probabilistic model optimization.
- Examine the characteristics, assumptions, and mathematical formulations of major probabilistic generative model families.
- To develop and assess deep generative models for solving real-world artificial intelligence problems.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Understand and apply Bayesian principles to develop generative models and make data-driven decisions under uncertainty.	**L3– Apply**	Understanding, Analytical Thinking
CO2	Analyse and apply information-theoretic metrics to quantify uncertainty and optimize generative model training.	**L4-Analyze**	Analytical Thinking, Critical Thinking, Quantitative Reasoning
CO3	Design and analyze sequence models using probabilistic graphical models to capture structured dependencies.	**L4-Analyze**	Modelling Skills, Analytical Thinking, Logical Reasoning
CO4	Apply advanced inference and sampling techniques to approximate posterior distributions, estimate expectations, and optimize probabilistic generative models.	**L3- Apply**	Problem Solving, Computational Thinking, Decision Making

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

CO5	Compare and analyze the characteristics, mathematical foundations, and applications of major probabilistic generative model families.	**L2- Understand**	Critical Thinking, Comparative Analysis, Conceptual Understanding
CO6	Design, implement, and evaluate deep generative models using probabilistic reasoning for artificial intelligence applications.	**L6- Create**	Design Thinking, Innovation, Implementation Skills, Evaluation

Details of the Course:

Unit 1: Bayesian Thinking and Generative Decision-Making

Contact Hours/Teaching Hours: 10

Bayes' Theorem: Theory and computation, Priors and posteriors in generative modelling, Bayesian decision theory in AI tuning and inference, Predictive uncertainty, Personalization, and adaptive priors.

Self-Learning Hours = 6

Students shall study Bayesian inference through numerical examples, implement Bayes' theorem for simple probabilistic problems, and explore uncertainty estimation in generative AI models.

Team Activity Hours = 3

Students shall collaboratively solve Bayesian inference problems, compare prior and posterior distributions for different scenarios, and present applications of Bayesian reasoning in AI.

Unit 2: Information Theory for Generative Modelling

Contact Hours/Teaching Hours: 08

Entropy: Quantifying uncertainty and information content, Cross-entropy and log-likelihood in model training, Kullback-Leibler (KL) divergence and its use in model alignment, Mutual information in token relationships and sequence modelling, Compression, coding theory, and their connection to model generalization.

Self-Learning Hours = 6

Students shall compute entropy, cross-entropy, and KL divergence for different probability distributions and analyze their significance in optimizing generative models.

Team Activity Hours = 3

Students shall compare various information-theoretic measures, discuss their role in generative AI, and present case studies on model optimization.

Unit 3: Probabilistic Graphical Models in Sequence Modelling

Contact Hours/Teaching Hours: 10

Bayesian Networks and Markov Models, Graph structures in modelling token dependencies, Mathematical formulation of Hidden Markov Models (HMMs), Inference methods: Belief propagation and Viterbi decoding, Role of structured probabilistic models in autoregressive frameworks.

Self-Learning Hours = 6

Students shall construct Bayesian Networks and Hidden Markov Models for simple sequence data and analyze inference techniques used in probabilistic sequence modelling.

Team Activity Hours = 3

Students shall design graphical models for sequential data, compare Bayesian Networks and Hidden Markov Models, and demonstrate inference techniques through group presentations.

Unit 4: Sampling, Inference, and Approximation

Contact Hours/Teaching Hours: 10

Probabilistic sampling techniques in decoding algorithms, Monte Carlo integration for expectations in models, MCMC algorithms: Theory and convergence, Variational inference: ELBO and reparameterization trick, Approximate Bayesian methods in large-scale generative tasks.

Self-Learning Hours = 6

Students shall implement Monte Carlo sampling methods, compare MCMC and variational inference approaches, and investigate approximate inference techniques for probabilistic models.

Team Activity Hours = 3

Students shall evaluate different inference and sampling algorithms, compare their computational efficiency and convergence properties, and present their findings using practical examples.

Unit 5: Generative Models through a Probabilistic Lens

Contact Hours/Teaching Hours: 10

Probabilistic formulation of Variational Autoencoders (VAEs): Latent variables and inference, Generative Adversarial Networks (GANs): Adversarial training and divergence minimization, Diffusion models: Stochastic processes and reverse sampling, Mathematical structure of autoregressive models (e.g., GPT), Score-based models and stochastic differential equations (SDEs).

Self-Learning Hours = 6

Students shall study the probabilistic foundations of VAEs, GANs, diffusion models, and autoregressive models, and evaluate their applications in modern generative AI.

Team Activity Hours = 3

Students shall compare deep generative model architectures, analyze their strengths and limitations for different AI applications, and present a comparative review based on recent research literature.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
TCS368.1	Understand and apply Bayesian principles to develop generative models and make data-driven decisions under uncertainty.	3	3	2	2	2	3	1	1	2	1	2	3	2	2
TCS368.2	Analyze and apply information-theoretic metrics to quantify uncertainty and optimize generative model training.	3	3	2	3	2	2	1	1	1	1	2	3	2	2
TCS368.3	Design and analyze sequence models using probabilistic graphical techniques and structured dependencies.	3	2	3	3	2	2	1	1	2	1	2	3	2	2
TCS368.4	Apply advanced inference and sampling techniques to approximate expectations and perform model tuning in large-scale generative tasks.	3	3	3	3	2	2	1	1	2	1	3	3	3	2
TCS368.5	Explore and differentiate various probabilistic generative model families and their mathematical underpinnings.	3	2	3	2	2	2	1	1	1	1	2	3	2	2

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

TCS368.6	Evaluate, design, and implement deep generative models for AI applications using probabilistic reasoning.	3	3	3	3	3	3	2	1	2	2	3	3	3	3
TCS 368		1	2.66	2.66	2.66	2.16	2.33	1.16	1.66	1.66	1.16	2.66	1	2.33	2.16

Text Books:

1. Probabilistic Machine Learning: An Introduction, Murphy, K. P. (2022). *Probabilistic Machine Learning: An Introduction*. MIT Press.
2. Pattern Recognition and Machine Learning, Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*.

Reference Books:

1. Machine Learning: A Probabilistic Perspective, Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.
2. Information Theory, Inference, and Learning Algorithms, MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.
3. Bayesian Reasoning and Machine Learning
Barber, D. (2012). *Bayesian Reasoning and Machine Learning*. Cambridge University Press.

Online Resources:

1. **NPTEL – Artificial Intelligence**
2. **SWAYAM – Online Courses in Artificial Intelligence and Machine Learning**
https://swayam.gov.in
3. **AICTE ELIS Portal (Engineering Learning Resources)**
https://www.elis.aicte-india.org

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Introduction to Statistical Data Science**
Course Code	:	**TCS342**
Semester	:	**III**
Pre-requisite	:	**Mathematics for AI-I, Mathematics for AI-II**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- Develop a strong foundation in statistical data science by understanding the principles of data analysis, probability theory, probability distributions and their applications in solving real-world problems.
- Develop an understanding of descriptive statistics, discrete and continuous random variables, covariance, outlier detection and statistical techniques used for data exploration and preprocessing.
- Enable students to analyze relationships among variables using correlation, linear regression and multiple regression techniques for predictive modelling and statistical analysis.
- Introduce statistical inference concepts including Bayes' theorem, Bayesian classification, Central Limit Theorem, confidence intervals, hypothesis testing and Z-score for evidence-based decision making.
- Develop the ability to perform parametric and non-parametric statistical tests such as t-test, Z-test, ANOVA and Chi-square test for analyzing datasets and drawing meaningful conclusions in data science applications.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Explain the fundamental concepts of data science, probability, and probability distributions, and apply statistical methodologies to analyze real-world problems.	**L2 – Understand**	Understanding, Analytical Thinking
CO2	Investigate and evaluate fundamental concepts of statistics and data science, and determine appropriate statistical methods for different practical scenarios.	**L3 – Apply**	Problem Solving, Programming
CO3	Apply statistical models including correlation, regression, probability distributions and descriptive statistics using suitable statistical tools for data analysis.	**L3 – Apply**	Programming, Design, Implementation

CO4	Apply Bayesian classification, Central Limit Theorem, confidence intervals, hypothesis testing, and Z-score techniques for statistical inference and data-driven decision making.	**L4 – Analyze**	Algorithmic Analysis, Programming, Critical Thinking
CO5	Apply parametric and non-parametric statistical testing methods and utilize statistical analysis techniques for extracting meaningful insights from datasets and supporting data science applications.	**L3 – Apply**	Data Structure Design, Programming, Problem Solving

Details of the Course:

Unit 1: Introduction Data Analysis **Contact Hours/Teaching Hours: 9**

Introduction to Modern Data Analysis, Components of Data Science, Applications of Data Science, Fundamental Statistics, Introduction to Probability, Probability Distribution, Continuous and Discrete Probability Distribution

Normal Distribution, Bernoulli Distribution, Binomial Distribution, Negative Binomial Distribution, Geometric Distribution, Poisson Distribution

Self-Learning Hours = 6

Students will work in groups to identify real-world applications of Data Science and present them. Each team will compare different probability distributions using practical examples and discuss their relevance.

Team Activity Hours = 3

Explore real-life datasets and study additional probability distributions through online resources and research articles.

Unit 2: **Random Variables and Descriptive Statistics**

Contact Hours/Teaching Hours: 9

Discrete random variables, Continuous random variable, Markov-chain Monte Carlo, Descriptive Statistics (Mean, Median, Mode, Standard Deviation), Sample Covariance, Correlation, Covariance Matrix, Outlier Detection.

Self-Learning Hours = 6

Practice analyzing simple datasets using Excel/Python and explore introductory videos on random variables, descriptive statistics, and the basic idea of MCMC.

Team Activity Hours = 3

Students will work in groups to analyze a small dataset, identify outliers, and calculate mean, variance, and covariance using Excel or Python. Each team will explain whether the variables are discrete or continuous with real-life examples.

Unit 3: Concepts of correlation

Contact Hours/Teaching Hours: 10

Concepts of Correlation, Regression, Linear square estimation, Simple Linear Regression, Multiple Regression.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Self-Learning Hours = 6

Practice correlation and regression analysis on sample datasets using Excel or Python and interpret the results.

Team Activity Hours = 3

Students will work in groups to analyze a small dataset and identify the relationship between variables using correlation and simple linear regression. Each team will present their findings with scatter plots and discuss the prediction results.

Unit 4: Introduction to Statistical Testing

Contact Hours/Teaching Hours: 10

Naïve Bayes' Theorem, Bayesian classification, Central Limit theorem, Data Exploration & preparation, Confidence Interval, The hypothesis-testing, Z-Score.

Self-Learning Hours = 6

Explore sample datasets to practice data preparation, confidence intervals, hypothesis testing, and Naïve Bayes using online tutorials or Python.

Team Activity Hours = 3

Students will work in groups to classify simple datasets using Naïve Bayes and perform basic hypothesis testing with Z-scores using Excel or Python. Each team will interpret the results and present their conclusions.

Unit 5: Introduction to Statistical Testing

Contact Hours/Teaching Hours: 08

Parametric Testing: Z-Test and t-Test, Non-Parametric Testing: Chi-Square Test and One-Way ANOVA, Interpretation of Test Results and p-value.

Self-Learning Hours = 6

Explore sample datasets to practice data preparation, confidence intervals, hypothesis testing, and Naïve Bayes using online tutorials or Python.

Team Activity Hours = 3

Students will work in groups to classify simple datasets using Naïve Bayes and perform basic hypothesis testing with Z-scores using Excel or Python. Each team will interpret the results and present their conclusions.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO01	3	2	-	-	2	-	-	-	-	-	-	1	2	-	-
CO02	3	3	2	1	3	-	-	-	-	-	-	2	3	2	1
CO03	3	3	3	2	3	-	-	-	-	-	-	2	3	3	2
CO04	3	3	3	3	3	-	-	-	-	-	-	2	3	3	2
CO05	3	3	3	3	3	-	-	-	1	1	-	2	3	3	3
Avg.	3	2.8	2.75	2.25	2.8	-	-	-	1.0	1.0	-	1.8	2.8	2.75	2

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Text Books:

1. Douglas C. Montgomery & George C. Runger, **Applied Statistics and Probability for Engineers,** Sixth Edition, Wiley, USA, 2016.
2. M. Ross, **Introduction to Probability and Statistics for Engineers and Scientists**, Fourth Edition, Academic Press, USA, 2009.
3. James D. Miller, **Statistics for Data Science**, First Edition, Packt Publishing, UK, 2017.

Reference Books:

1. D. C. Agarwal & Dr. Pradeep K. Joshi, Probability & Statistics for Data Science, First Edition, Shree Sai Prakashan, India, 2022.
2. Dr. Mark Gardener, Beginning R – The Statistical Programming Language, First Edition, Wiley, USA, 2012

Online Resources:

1. **NPTEL – Introduction to Machine Learning**
2. https://nptel.ac.in/
3. **NPTEL – Data Analytics with Python**
   https://nptel.ac.in/
4. **Kaggle Learn**
   https://www.kaggle.com/learn
5. **W3Schools Python**
   https://www.w3schools.com/python

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Introduction to Cryptography**
Course Code	:	**TCS 392**
Semester	:	**III**
Pre-requisite	:	

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- Explain the fundamental concepts of computer and network security, including security attacks, security services, security mechanisms, cryptographic techniques, and steganography.
- Apply symmetric key cryptographic algorithms, key distribution techniques, and mathematical foundations of cryptography to secure data communication and storage.
- Analyze public-key cryptographic algorithms, message authentication techniques, and hash functions for ensuring confidentiality, integrity, and authentication of information.
- Evaluate the effectiveness of cryptographic algorithms, authentication mechanisms, and key management techniques in securing communication systems.
- Analyze system security mechanisms, intrusion detection techniques, malware countermeasures, firewall technologies, and the legal and ethical aspects of cybersecurity.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Explain the fundamental concepts of computer and network security, security attacks, security services, classical cryptographic techniques, and steganography to identify security vulnerabilities in communication systems.	**L2 – Understand**	Basic security knowledge and analytical skills.
CO2	Apply symmetric key cryptographic techniques, including DES, Double DES, stream ciphers, RC4, and key distribution mechanisms, to ensure secure data communication and storage.	**L3 – Apply**	Cryptography basics and problem-solving skills.
CO3	Analyze the mathematical foundations of cryptography, public-key cryptographic algorithms (RSA), and message authentication techniques using hash functions for secure communication.	**L4 - Analyze**	Mathematical and analytical skills.
CO4	Evaluate the effectiveness of authentication mechanisms, cryptographic algorithms, and key	**L5 – Evaluate**	Evaluation and decision-making skills.

	management techniques for securing data and network communications.		
CO5	Analyze system security threats, intrusion detection mechanisms, malicious software, firewall technologies, and legal and ethical issues related to cybersecurity for protecting information systems.	**L4 – Analyze**	Security analysis and problem-solving skills.

Details of the Course:

Unit 1: Introduction **Contact Hours/Teaching Hours: 08**

Introduction: Computer Security Concepts: The OSI Security Architecture, 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.

Self-Learning Hours = 6

Study security concepts and practice classical cryptography techniques.

Team Activity Hours = 3

Discuss cyberattack case studies and demonstrate classical ciphers in groups

Unit 2: Modern Cryptography

Contact Hours/Teaching Hours: 09

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.

Self-Learning Hours = 6

Study symmetric encryption algorithms and solve DES/RC4-based numerical problems.

Team Activity Hours = 3

Compare symmetric encryption techniques and demonstrate DES encryption in groups.

Unit 3: Symmetric Key Encryption

Contact Hours/Teaching Hours: 10

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.

Self-Learning Hours = 6

Study key distribution techniques and practice GCD, Euclidean algorithm, Euler's totient function, and related programming problems.

Team Activity Hours = 3

Solve key distribution case studies and implement cryptographic mathematical algorithms in groups.

Unit 4: Public Key Cryptography

Contact Hours/Teaching Hours: 09

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.

Self-Learning Hours = 6

Study RSA encryption, message authentication methods, and solve RSA numerical problems.

Team Activity Hours = 3

Demonstrate RSA encryption/decryption and compare message authentication techniques in groups.

Unit 5: System Security

Contact Hours/Teaching Hours: 09

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.

Self-Learning Hours = 6

Study system security threats, malware, firewalls, and cyber laws through case studies.

Team Activity Hours = 3

Analyze cybercrime case studies and present security measures and ethical practices in groups.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO01	3	2			1			2				1	3		
CO02	3	2	2		3							1	3	2	
CO03	3	3	2	2	3							1	3	3	
CO04	3	3	3	2	3							1	3	3	
CO05	2	2			2	3	2	3	1	2		2	2	2	1
Avg.	3	2	2	2	2	3	2	3	1	2		1	3	2	1

Text Books:

1. William Stallings (2018). Network Security Essentials Applications and Standards, 6^{th} Edition, Pearson Education.

2. William Stallings (2017). Cryptography and Network Security, 7th Edition, Pearson Education.

Reference Books:

1. Behrouz Forouzan,(2015).Cryptography and Network Security, 3rd Edition, McGraw Hill.
2. Atul Kahate,(2017).Cryptography and Network Security, 3rd Edition, McGraw Hill.

Online Resources:

1. **GeeksforGeeks – Cryptography Tutorial**
   https://www.geeksforgeeks.org/computer-networks/cryptography-tutorial/
2. **Coursera – Cryptography L**
   https://www.coursera.org/learn/crypto
3. **NPTEL – Cryptography and Network Security**
   https://nptel.ac.in/courses/106105162

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name : **Information Security Foundations**
Course Code : **TCS324**
Semester : **III**
Pre-requisite : **Fundamental of Computers (TCS101), Programming for Problem Solving (TCS201)**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To understand the fundamental concepts of information security, security threats, authentication, access control, ethical hacking, penetration testing, SSL and TLS.
- To understand the basic concepts of cryptography, security goals, cryptographic applications, keys, cryptanalysis, and OSI security architecture.
- To apply mathematical concepts such as prime numbers, Euclidean algorithms, modular arithmetic, Euler's totient theorem and the Chinese remainder theorem in information security.
- To understand and apply symmetric-key cryptosystems, classical ciphers, RC4, DES, 3DES, AES and block cipher modes of operation.
- To understand and apply asymmetric cryptography, key exchange, digital signatures, elliptic curve cryptography, hash functions, MAC and HMAC for secure communication and message integrity.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Explain fundamental information security concepts including threats, authentication, access control, security pillars, hackers, penetration testing, SSL and TLS.	L2 – Understand	Employability Skill – Cybersecurity Foundations
CO2	Explain the fundamental concepts of cryptography, security goals, cryptographic applications, keys, cryptanalysis, and OSI security architecture.	L2 – Understand	Employability Skill – Security and Cryptographic Literacy
CO3	Apply mathematical concepts such as GCD, extended Euclidean algorithm, modular arithmetic, Euler's totient theorem and Chinese remainder theorem in information security.	L3 – Apply	Employability Skill – Mathematical Problem-Solving
CO4	Apply symmetric-key cryptographic techniques including classical ciphers, RC4, DES, 3DES, AES and block cipher modes of operation.	L3 – Apply	Employability Skill – Secure System Implementation

CO5	Analyze and apply asymmetric cryptosystems, key exchange, digital signatures, ECC, ElGamal, hash functions, MAC and HMAC for message integrity and secure communication.	L4 – Analyze	Employability Skill – Cybersecurity and Secure Communication

---

Details of the Course:

Unit 1: Introduction to Information Security **Contact Hours: 10**

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 cybercrime – identify cyber threats, analyze and evaluate threat, treatment. Types 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.

Self-Learning Hours = 6

Students shall practice identifying common security threats, authentication methods, access-control techniques and penetration-testing phases.

Team Activity Hours = 3

Group activity on analysing a simple real-life cyber incident and mapping threats, vulnerabilities, security controls and appropriate security services.

Unit 2: Basics of Cryptography **Contact Hours: 10**

What is cryptography, confidentiality, data integrity, authentication and nonrepudiation. Applications of cryptography – chip-based payment cards, digital currencies, computer passwords and digital communications. Plaintext, ciphertext and cipher; characteristics of a good cipher; encryption and decryption; key and significance of key length; symmetric and asymmetric key cryptography; cryptanalysis. OSI security architecture – security attacks, security services and security mechanisms.

Self-Learning Hours = 6

Students shall practice identifying security goals, classifying cryptographic systems and distinguishing plaintext, ciphertext, encryption, decryption and keys.

Team Activity Hours = 3

Group-based comparison of cryptographic applications and OSI security attacks, services and mechanisms.

Unit 3: Mathematics Applied in Information Security **Contact Hours: 8**

Concept of divisibility, prime numbers and 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.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Self-Learning Hours = 6

Students shall solve numerical problems involving GCD, modular arithmetic, Euler's totient theorem and the Chinese remainder theorem.

Team Activity Hours = 3

Team problem-solving activity on mathematical computations used in cryptographic algorithms.

Unit 4: Symmetric Key Cryptosystem **Contact Hours: 10**

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 operation.

Self-Learning Hours = 6

Students shall practice encryption and decryption using selected classical ciphers and study the working of RC4, DES, 3DES and AES.

Team Activity Hours = 3

Group activity comparing classical and modern symmetric ciphers and demonstrating block cipher modes of operation.

Unit 5: Asymmetric Key Cryptosystem, Digital Signature and Message Integrity
Contact Hours: 8

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.

Self-Learning Hours = 6

Students shall solve basic RSA and Diffie-Hellman problems and practice identifying suitable digital-signature and message-integrity mechanisms.

Team Activity Hours = 3

Case-study presentation on secure communication, digital signatures, hashing, MAC/HMAC and applications of asymmetric cryptography.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Articulation Matrix

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO 1	PSO 2	PSO 3
TCS324.1	2	1	-	2	1	-	-	-	-	1	-	2	1	2	1
TCS324.2	2	2	1	-	2	-	-	-	1	1	1	2	2	3	1
TCS324.3	2	3	1	-	2	-	-	-	1	1	1	2	2	3	1
TCS324.4	2	2	2	1	3	-	-	-	1	2	1	2	2	3	2
TCS324.5	2	3	2	2	3	-	-	-	1	2	1	2	2	3	2

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

Text Books

1. William Stallings, Cryptography and Network Security: Principles and Practice, 8th Edition, Pearson Publication, India, 2020.
2. Charles P. Pfleeger, Shari Lawrence Pfleeger and Jonathan Margulies, Security in Computing, 5th Edition, Prentice Hall, India, 2015.

Reference Books

1. William Stallings, Network Security Essentials: Applications and Standards, 6th Edition, Prentice Hall, India, 2016.
2. Behrouz A. Forouzan, Cryptography and Network Security, McGraw-Hill.

Online Resources

1. NIST Cybersecurity Resources – https://www.nist.gov/cybersecurity
2. NIST Computer Security Resource Centre – https://csrc.nist.gov/
3. OWASP Foundation – https://owasp.org/
4. IETF RFCs – https://www.rfc-editor.org/

Graphic Era HILL UNIVERSITY
Established by an Act of the State Legislature of Uttarakhand (Adhiniyam Sankhya 12 of 2011)
University under section 2(f) of UGC Act, 1956

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name : **Logic Design & Computer Organization**
Course Code : **TCS308**
Semester : **III**
Pre-requisite : **Basic Electronics Engineering**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- Apply Boolean algebra, Karnaugh maps, Quine–McCluskey method, and logic gate implementations to simplify and design combinational digital circuits including adders, subtractors, multiplexers, decoders, encoders, comparators, and code converters.
- Analyse and design sequential logic circuits using latches, flip-flops, registers, and shift registers, and perform flip-flop conversions based on characteristic and excitation tables.
- Design and analyse synchronous and asynchronous counters, finite state machines (Mealy and Moore models), and develop state diagrams and state tables for sequential circuit applications.
- Explain computer organization and architecture concepts, processor operation, instruction cycle, pipelining, arithmetic operations, and evaluate computer performance using appropriate performance measures.
- Analyse memory organization, cache memory, I/O techniques, bus arbitration mechanisms, and compare alternative computer architectures based on Flynn's taxonomy.

Course Outcomes:

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

Course Outcome (CO)	Statement	Bloom's Level	Skill
CO1	Apply Boolean algebra and combinational logic design techniques for digital circuit implementation.	L2-Understand L3- Apply , L6- Create	Conceptual Understanding, Problem Solving, Circuit Design & HDL Programming
CO2	Analyze and design sequential logic circuits using latches, flip-flops, and registers.	L4-Analyze L6- Create	Analytical Thinking, Hardware Design
CO3	Analyze and design counters and finite state machine-based sequential systems.	L4-Analyze L6- Create	System Analysis ,Design & Development
CO4	Analyze computer architecture, arithmetic operations, processor organization, and performance evaluation techniques.	L2-Understand L4-Analyze L5-	Conceptual Understanding , System Analysis, Performance

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

		Evaluate	Evaluation
CO5	Evaluate memory systems, I/O organization, and alternative computer architectures.	**L4-Analyze L5-Evaluate**	Critical Analysis, Technology Assessment & Decision Making

Details of the Course:

Unit 1:

Contact Hours/Teaching Hours: 10

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-Multiplexers, code conversion. Introduction to HDL description of combinational logic

Self-Learning Hours = 3

Students shall Practice Boolean simplification, K-map problems and HDL tutorials.

Team Activity Hours = 2

Design and simulate a combinational logic circuit.

Unit 2:

Contact Hours/Teaching Hours: 08

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.

Self-Learning Hours = 2

Study timing diagrams and storage element applications.

Team Activity Hours = 2

Group presentation on sequential storage elements.

Unit 3:

Contact Hours/Teaching Hours: 8

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.

Self-Learning Hours = 2

FSM design exercises and counter applications.

Team Activity Hours = 2

Team-based synchronous counter/FSM design project.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Unit 4: **Contact Hours/Teaching Hours: 10**

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.

Self-Learning Hours = 2

Study modern processors, RISC vs CISC and performance evaluation.

Team Activity Hours = 2

Comparative seminar and case study discussion.

Unit 5:

Contact Hours/Teaching Hours: 09

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).
Cache design, replacement algorithms and memory performance analysis.
Group case study on cache architecture and Flynn classification.

Self-Learning Hours = 2

Cache design, replacement algorithms and memory performance analysis.

Team Activity Hours = 2

Group case study on cache architecture and Flynn classification.

<u>Course Articulation Matrix</u>

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO01	1	3	-	-	-	-	-	-	1	1	-	1	2	-	-
CO02	1	2	3	-	-	-	1	1	2	1	-	1	2	2	1
CO03	1	2	2	3	3	1	-	1	1	2	1	2	3	2	2
CO04	1	2	3	3	-	1	1	1	1	1	1	1	3	2	2
CO05	1	2	-	-	-	-	1	1	1	1	-	1	2	2	-
Avg.	1	2.2	1.6	1.2	0.6	0.4	0.6	0.8	1.2	1.2	0.4	1.2	2.4	1.6	1

Text Books:

1. M. Morris Mano, Digital Logic and Computer Design, Pearson, 1st edition, 2016.
2. W. Stalling, Computer Organization and Architecture, Pearson, 11th edition, 2022.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Reference Books:

1. Charles H. Roth Jr., Fundamentals of Logic Design, Wadsworth Publishing., 5th edition, 2005.
2. John P Hayes, Computer Architecture and Organization, McGraw Hill,3rd edition, 2017.

Online Resources:

1. **ePathshala – Digital Electronics**
   https://epgp.inflibnet.ac.in/Home/ViewSubject?catid=uUIVj2W71X+8mppiIHe0+A==
2. **NPTEL – Switching Circuits and Logic Design**
   https://nptel.ac.in/courses/106105185
3. **NPTEL – Introduction to Digital Systems Design**
   https://nptel.ac.in/courses/117105080
4. **NPTEL – Computer Organization and Architecture**
   https://nptel.ac.in/courses/106106166