Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Operating Systems**	
Course Code	:	**TCS 502**	
Semester	:	**V**	
Pre-requisite	:	**Data structures with C (TCS-302).**	

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- Understand the concepts, architecture, services, and design principles of operating systems.
- Analyse process management, CPU scheduling, synchronization, and deadlock handling techniques.
- Understand memory management and virtual memory techniques in modern operating systems.
- Apply system calls for process, memory, and file management in UNIX/Linux operating systems.
- Analyze file systems, secondary storage management, protection mechanisms, and operating system implementation techniques.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Explain the concepts, architecture, services, UNIX environment, system calls, and design issues associated with operating systems	L2 – Understand	Conceptual Understanding, Analytical Thinking
CO2	Analyse process management, process synchronization, CPU scheduling, and deadlock prevention, avoidance, detection, and recovery techniques in operating systems	L4 – Analyze	Process Analysis, Problem Solving, Critical Thinking
CO3	Explain memory management strategies, paging, segmentation, virtual memory, and page replacement techniques used in modern operating systems	L2 – Understand	Memory Management, Analytical Thinking, Problem Solving
CO4	Apply system calls for process management, memory management, and file management in UNIX/Linux operating systems	L3 – Apply	UNIX/Linux Programming, System Administration, Practical Implementation

CO5	Analyse file systems, secondary storage management, protection mechanisms, and operating system data structures and algorithms	L4 – Analyze	File System Management, Storage Management, System Security

---

Details of the Course:

Unit 1: Introduction to Operating Systems and UNIX **Contact Hours/Teaching Hours: 8**

What operating systems do, Operating system structure, Operating system services, Operating system classification, User–Operating System interface, System calls, Types of system calls, Operating system structure. UNIX: Command structure, Internal and external commands, Filters, vi editor.

Self-Learning Hours = 6

Students should study the functions and services of operating systems, compare different operating system classifications, understand system calls and user interfaces, practice UNIX commands, learn text processing using filters, and gain hands-on experience with the vi editor.

Team Activity Hours = 3

Students should work in teams to perform basic UNIX file and directory operations, execute internal and external commands, use filters for text processing, and demonstrate the use of the vi editor. Each team should present the role of operating system services and system calls through practical examples.

Unit 2: Process Management and Process Synchronization **Contact Hours/Teaching Hours: 10**

Process concept, Operations on processes, Multithreading models, Threading issues. Process Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple-processor scheduling, Thread scheduling. Process Synchronization: Inter-process communication, Critical section problem, Peterson's solution, Synchronization hardware, Semaphores, Classical synchronization problems**.**

Self-Learning Hours = 8

Students should study process life cycles, compare CPU scheduling algorithms, understand multithreading models, learn and analyse synchronization mechanisms, and practice solving problems related to semaphores and classical synchronization**.**

Team Activity Hours = 4

Students should work in groups to simulate CPU scheduling algorithms and compare their performance. Each team should also implement a synchronization problem such as Producer–Consumer or Dining Philosophers and discuss the synchronization techniques used.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Unit 3: Deadlocks **Contact Hours/Teaching Hours: 8**

Deadlocks: System model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Resource trajectories, Safe and unsafe states, Banker's algorithm, Deadlock detection, Recovery from deadlock.

Self-Learning Hours = 6

Students should study the necessary conditions for deadlocks, analyse prevention and avoidance techniques, solve Banker's algorithm problems, and understand deadlock detection and recovery mechanisms.

Team Activity Hours = 3

Students should work in teams to analyze resource allocation scenarios, identify deadlocks, determine safe sequences using Banker's algorithm, and present different deadlock handling techniques.

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

Memory Management Strategies: Background, Swapping, Contiguous memory allocation, Paging, Structure of page table, Segmentation. Virtual Memory Management: Background, Demand paging, Page replacement, Allocation of frames, Thrashing.

Self-Learning Hours = 8

Students should study memory allocation techniques, practice numerical problems on paging and segmentation, compare page replacement algorithms, and understand virtual memory management and thrashing.

Team Activity Hours = 4

Students should work in groups to simulate page replacement algorithms, compare their performance, analyse memory utilization techniques, and present the advantages and limitations of different memory management strategies.

Unit 5: File System Management **Contact Hours/Teaching Hours: 8**

File System: File concept, Access methods, Directory structure, Protection, File system structure, Directory implementation, Allocation methods, Free space management. Secondary Storage Structures: Mass storage structures, Disk structure, Disk scheduling, Disk management, Swap space management. Protection: Goals of protection, Principles of protection, Access matrix.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Self-Learning Hours = 6

Students should study file organization and directory structures, analyze file allocation and free space management techniques, practice disk scheduling problems, and understand protection mechanisms and access control models.

Team Activity Hours = 3

Students should work in teams to compare disk scheduling algorithms and file allocation methods, analyze file system protection mechanisms, and present practical solutions for efficient and secure file system management.

Course Articulation Matrix

Course Outcome	PO1	PO 2	PO 3	PO4	PO5	PO6	PO7	PO8	PO9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3
CO_{01}	3	2	1		2					1		1	2	2	1
CO_{02}	3	3	2	2	2					1		1	3	3	2
CO_{03}	3	3	2	2	2							1	2	3	2
CO_{04}	2	2	2	1	3					2		1	3	3	2
CO_{05}	2	3	2	2	3			1		1		2	3	3	2
Avg.	3	3	2	2	2			1		1		1	3	3	2

Text Books:

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne: Operating System Principles, 7th edition, Wiley India, 2006.
2. Unix concepts and applications – Sumitabha Das

Reference Books:

1. Andrew S Tanenbaum: Operating Systems: Design and Implementation, 3rd edition, Prentice Hall, 2006.
2. Stuart E. Madnick, John Donovan: Operating Systems, Tata McGraw Hill, 2008.

Online Resources:

https://www.geeksforgeeks.org/operating-systems/operating-systems

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	**:**	**Database Management System**
Course Code	**:**	**TCS503**
Semester	**:**	**V**
Pre-requisite	**:**	**Data Structures with C(TCS302)**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To understand the fundamental concepts of Database Management System, including data models, database architecture, and the advantages of DBMS over traditional file systems.
- To develop the ability to design and model relational databases using Entity-Relationship (ER) diagrams, relational algebra, normalization techniques, and schema design principles.
- To acquire proficiency in SQL to create, manipulate, retrieve, and manage data efficiently using Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), Data Query Language (DQL), and Transaction Control Language (TCL).
- To understand advanced database concepts, including transaction management, concurrency control, recovery mechanisms, indexing, query processing, and database security to ensure data integrity and efficient database operations.
- Apply database concepts to real-world applications by designing, implementing, and managing database solutions that support software applications, decision-making, and enterprise information systems.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Understand the different issues involved in the design and implementation of a database system.	**L2 – Understand**	Understanding the concepts of database modelling, Analytical Thinking
CO2	Study the physical and logical database designs, database modelling, relational, hierarchical, and network models.	**L3 – Apply**	Designing of database, Query Processing
CO3	Understand and use data manipulation language to query, update, and manage a database.	**L3 – Apply**	Database Design, Implementation
CO4	Develop an understanding of essential DBMS concepts such as: database security, integrity, and concurrency.	**L4 – Analyze**	Database Constraints Analysis and Implementation
CO5	Design and build a simple database system and demonstrate competence with the fundamental tasks	**L3 – Apply**	Designing of database using DBMS software

	involved with modelling, designing, and implementing a DBMS.		

Details of the Course:

Unit 1: Introduction **Contact Hours/Teaching Hours: 09**
Introduction: An overview of DBMS; Advantages of using DBMS approach; Database systems vs File Systems, Database system concepts and architecture Data models, schemas, and instances; Three-schema architecture and data independence; Database languages and interfaces; The database system environment; Centralized and client-server architectures; Classification of Database Management systems.

Self-Learning Hours = 6
Students shall understand the concept of database modelling and architecture of DBMS. Also analyse the benefits of having DBMS over traditional file system.

Team Activity Hours = 3
Group discussion on understanding the schema design of a relational database system.

Unit 2: Entity-Relationship Model **Contact Hours/Teaching Hours: 09**
Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes and Keys; Relationship types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types; Refining the ER Design; ER Diagrams, Naming Conventions and Design Issues; Relationship types of degree higher than two.

Self-Learning Hours = 6
Students shall understand the conceptual level of database design using ER model and various parameters of ER model. Moreover, they will also understand the conversion of ER model into relational model using query language.

Team Activity Hours = 3
Group-based activity on solving practical problems like building of various ER models and conversion of them into relational model using SQL.

Unit 3: Relational Model and Relational Algebra **Contact Hours/Teaching Hours: 11**
Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, Transactions and dealing with constraint violations; Unary Relational Operations: SELECT and PROJECT; Relational Algebra and Calculus Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION; Additional Relational Operations. Examples of Queries in Relational Algebra; Relational Database Design Using ER- to-Relational Mapping.

SQL – 1: SQL Data Definition and Data Types; Specifying basic constraints in SQL; Schema change statements in SQL; Basic queries in SQL; More complex SQL Queries.

Insert, Delete and Update statements in SQL; Specifying constraints as Assertion and Trigger; Views (Virtual Tables) in SQL; Additional features of SQL; Database programming issues and techniques;

Embedded SQL, Dynamic SQL; Database stored procedures. Optimization of SQL Queries through Indexes, Concepts of NoSQL.

Self-Learning Hours = 6

Students shall implement various SQL queries like DDL, DML, DCL, TCL, and DQL. Moreover, they will also understand the conversion of relational algebra expression into SQL query.

Team Activity Hours = 3

Group programming activity to develop an efficient database design and to apply various constraints like integrity constraint, key constraints, domain constraints, referential integrity constraints etc. Moreover, they will also design a relational database using optimize queries and by applying the concept of triggers, and view.

Unit 4: Database Design – 1 **Contact Hours/Teaching Hours: 09**

Informal Design Guidelines for Relation Schemas; Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form Properties of Relational Decompositions; Algorithms for Relational Database Schema Design; Multivalued Dependencies and Fourth Normal Form; Join Dependencies and Fifth Normal Form; Inclusion Dependencies; Other Dependencies and Normal Forms.

Self-Learning Hours = 6

Students shall implement the concept of normalization to design an efficient relational model. They will also implement the concept of keys and functional dependency.

Team Activity Hours = 3

Group activity on analysing the different normal forms and to achieve the best suitable normal form for removing the redundancies from the relational database.

Unit 5: Transaction Management **Contact Hours/Teaching Hours: 10**

The ACID Properties: Transactions and Schedules; Concurrent Execution of Transactions; Lock- Based Concurrency Control; Performance of locking; Transaction support in SQL; Introduction to crash recovery; 2PL, Serializability and Recoverability; Lock Management; Log Files; Checkpointing; Recovering from a System Crash; Media Recovery

Self-Learning Hours = 6

Students shall implement transaction control languages like rollback, commit, and save point etc., They will also implement the concurrent transaction management.

Team Activity Hours = 3

Case study and group presentation on real-world applications of Database Management System, SQL, Trigger, View, Normalization, Constraints, and Indexing etc.

Course Articulation Matrix

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

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

CO05	2	1	2	2	3	-	-	-	-	-	-	-	3	3	2
Avg.	**1.2**	**2**	**2**	**1.4**	**2.5**	**1**	**1**	**-**	**1**	**-**	**1**	**1**	**2.2**	**2.2**	**1.6**

Text Books:

1. McGraw-Hill. Date K., Swamynathan S. An Introduction to Database Systems. Eight Edition Pearson. 2012.
2. Elmasri R. and Navathe S.B., Fundamentals of Database Systems. 2nd 2012.

Reference Books:

1. Singh S.K., Database Systems- Concepts, Designs and Application. 2nd Edition. Pearson 2011.
2. C.J. Introduction to Database Systems (Vol I & II) 8th Edition. Addison- Wesley. 8th 2004.
3. Silberschatz A. Korth H. F. Sudarshan S., Database System Concepts. Sixth Edition 1st 2014.

Online Resources:

1. **GeeksforGeeks – DBMS**
   DBMS Tutorial - GeeksforGeeks
2. **NPTEL – Database Management Systems**
   https://nptel.ac.in/
3. **SQL by W3school:**
   SQL Tutorial

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**System Design**
Course Code	:	**TCS504**
Semester	:	**V**
Pre-requisite	:	**TCS101, TCS307/TCS346/TCS408(OOPs)**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- **Develop a strong understanding of software design principles** by applying clean coding practices, object-oriented programming concepts, UML modeling, and machine coding techniques for developing maintainable software systems.
- **Analyze and apply object-oriented design principles and design patterns** to build modular, reusable, extensible, and scalable software solutions for real-world applications.
- **Design Low-Level Designs (LLD)** of software systems by identifying classes, relationships, responsibilities, and interactions using appropriate object-oriented modeling techniques.
- **Understand the fundamentals of High-Level Design (HLD)** including software architecture, distributed systems, databases, caching, networking, scalability, and asynchronous communication for designing large-scale applications.
- **Design end-to-end scalable software architectures** by integrating Low-Level Design, High-Level Design, modern architectural components, security mechanisms, and real-world case studies to solve complex engineering problems.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Understand clean coding practices, object-oriented design principles, UML, ER diagrams, and machine coding techniques to develop modular, maintainable, and reusable software systems.	**L2 - Understand**	**Software Design Fundamentals,** Object-Oriented Modeling, Technical Documentation
CO2	Apply object-oriented concepts, SOLID principles, design patterns, UML modeling, and machine coding practices to design scalable and maintainable Low-Level Design (LLD) solutions for real-world applications.	**L3 - Apply**	**Problem Solving**, Object-Oriented Design, Software Development, Code Reusability
CO3	Analyze High-Level Design (HLD) concepts including software architecture, scalability, networking, caching, database design, load	**L3 - Apply**	**Software Architecture,** High-Level Design,

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

	balancing, and distributed system fundamentals.		Scalable System Design, Distributed System Fundamentals
CO4	Analyze and apply NoSQL databases, asynchronous messaging, authentication, authorization, and security mechanisms for building scalable distributed applications.	**L4 - Analyze**	**Distributed Systems**, Data Engineering, Asynchronous System Design, Application Security
CO5	Evaluate and implement end-to-end large-scale software systems by integrating LLD and HLD concepts, NoSQL databases, asynchronous messaging, authentication, security mechanisms, and practical case studies to address real-world engineering problems.	**L5 - Evaluate**	**Large-Scale System** Design, Software Architecture Evaluation, System Integration, Engineering Decision Making

Details of the Course:

Unit 1: Clean Coding, Object Oriented Principles & Their applications, UML & ERD & Machine Coding

Contact Hours/Teaching Hours: 9

Clean coding mindset; readable, modular and maintainable code; naming conventions; avoiding code smells; scalable code structure; Object-oriented thinking: encapsulation, abstraction, inheritance vs composition, polymorphism; UML: class diagrams, sequence diagrams; ER diagrams; Machine coding: requirement analysis, class design, relationships, modular code implementation

Self-Learning Hours = 6

Students shall study clean coding guidelines, prepare UML diagrams for real-world applications, and implement small machine coding problems using object-oriented programming concepts.

Team Activity Hours = 3

Group activity to analyze poorly designed code, refactor it using clean coding practices, and prepare UML diagrams for a real-world application.

Unit 2: **Deciphering Design principles (SOLID), Design Patterns and Case Studies**

Contact Hours/Teaching Hours: 10

SOLID principles: SRP, OCP, LSP, ISP, DIP using bad vs refactored designs; Creational patterns: Singleton, Factory Method, Abstract Factory, Builder; Behavioral patterns: Strategy, Observer, Command; Structural patterns: Adapter, Decorator, Proxy, Bridge.

Low Level Design Case Studies

Tic Tac Toe/Chess, LRU Cache, Parking Lot, ATM, Elevator, Movie Ticket Booking

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Self-Learning Hours = 6

Students shall implement selected design patterns and prepare Low-Level Designs for small software systems using UML and object-oriented concepts.

Team Activity Hours = 3

Group-based machine coding exercises to design and implement an LLD solution for a real-world application such as Parking Lot or ATM System.

Unit 3: Foundations of High-Level Design

Contact Hours/Teaching Hours: 9

Architecture: Monolithic, N-tier, vertical vs horizontal scaling, latency vs throughput; Networking: REST, RPC, GraphQL, WebSockets; Load balancing (L4/L7, Round Robin, Least Connections, Consistent Hashing); Caching: Cache-aside, Read-through, Write-through, Write-back, Write-around, LRU, LFU, Redis/Memcached; Databases: ACID vs BASE, CAP theorem, replication, sharding.

Self-Learning Hours = 6

Students shall study popular system architectures and compare different caching, scaling, and database design strategies used in modern software systems.

Team Activity Hours = 3

Group discussion on designing scalable architectures for applications such as e-commerce, video streaming, and online banking.

Unit 4: Data Handling & Asynchronous Processing

Contact Hours/Teaching Hours: 9

NoSQL databases: Key-value, Document, Columnar, Graph databases and use cases; Async processing, RabbitMQ, Kafka, Pub/Sub, backpressure, dead-letter queues; Identity & Security: Authentication vs Authorization, Session vs JWT, OAuth2, OpenID Connect, HTTPS/SSL.

Self-Learning Hours = 6

Students shall explore NoSQL databases, message brokers, and authentication mechanisms by studying real-world system architectures.

Team Activity Hours = 3

Group activity to design an event-driven architecture using Kafka/RabbitMQ and discuss security mechanisms for modern web applications.

Unit 5: Advanced Case Studies

Contact Hours/Teaching Hours: 9

Food Delivery System Design, Ride-Sharing System Design, Payment System Design, URL Shortener System Design, Stock Trading System Design, E-Commerce System Design (Amazon Lite), Social Media Platform Design (Facebook/Instagram/Twitter), Real-Time Chat Application Design (WhatsApp/Discord).

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Self-Learning Hours = 6

Students shall analyze architecture documents, study large-scale distributed systems, and compare different architectural approaches used by leading technology companies.

Team Activity Hours = 3

Group project to design and present the complete architecture of a real-world software system, including HLD, LLD, database schema, APIs, caching strategy, messaging, and security considerations.

Course Articulation Matrix

CO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2	PSO 1	PSO 2	PSO 3
CO1	3	2	2	–	2	–	–	1	1	2	–	1	3	2	1
CO2	3	3	3	–	3	–	–	1	1	1	1	2	3	3	2
CO3	3	3	3	2	3	–	–	1	1	1	2	2	3	3	3
CO4	3	3	2	2	3	–	–	2	1	1	2	2	3	3	3
CO5	3	3	3	2	3	–	–	2	2	2	2	3	3	3	3
Avera ge	3	3	3	2	3	0	0	1	1	1	1	2	3	3	2

Text Books:

1. Head First Design Patterns (2nd Edition), Eric Freeman, Elisabeth Robson, O'Reilly
2. Clean Architecture: A Craftsman's Guide to Software Structure and Design, Robert C. Martin, Pearson
3. Designing Data-Intensive Applications, Martin Kleppmann, O'Reilly Media

Reference Books:

1. System Design Interview – Volume 2, Alex Xu, Sahn Lam, ByteByteGo

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

				L	T	P	Credits
Course Name	:	**Machine Learning**		3	0	0	3
Course Code	:	**TCS509**					
Semester	:	**V**					
Pre-requisite	:	**Basic knowledge of Python programming (variables, loops, functions, lists, dictionaries, and file handling) along with familiarity with data handling concepts.**					

Course Learning Objectives:

- Describe fundamental statistical concepts, machine learning paradigms, and Python programming constructs required for machine learning applications.
- Perform exploratory data analysis, data preprocessing, data transformation, and data visualization to prepare datasets for machine learning tasks.
- Apply supervised machine learning algorithms to solve classification and regression problems using appropriate tools and techniques.
- Analyze datasets using clustering and dimensionality reduction techniques to identify patterns and improve data representation.
- Evaluate and optimize machine learning models using performance metrics, cross-validation methods, and hyperparameter tuning techniques

---

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 structures, algorithms, pointers, dynamic memory allocation, arrays, and analyze their time and space complexity for solving computational problems.	L2 – Understand	Understanding, Analytical Thinking
CO2	Apply stack and queue data structures to solve real-world problems, perform expression conversion and evaluation, and implement various queue types using arrays.	L3 – Apply	Data Analysis, Problem Solving, Data Preparation
CO3	Design and implement singly, circular, and doubly linked lists, and perform insertion, deletion, traversal, searching, and implementation of stacks and queues using linked lists.	L4 – Analyze	Machine Learning Modeling, Analytical Thinking, Problem Solving
CO4	Implement searching and sorting algorithms, apply hashing techniques with collision resolution	L4 – Analyze	Data Analysis, Critical Thinking, Pattern Recognition

	strategies, and evaluate the efficiency of different searching and sorting methods.		
CO5	Construct and manipulate binary trees, binary search trees, threaded binary trees, and Huffman trees, and perform tree traversals, insertion, deletion, and Huffman coding applications.	**L5-Evaluate**	Model Evaluation, Optimization, Decision Making

Details of the Course:

Unit 1: Review of Statistical Concepts: **Contact Hours/Teaching Hours: 8**

Mean, Median, Mode, Outliers, Range, Average Deviation, Absolute Deviation, Squared Deviation, Standard Deviation, Total Sum of Squares. **Introduction to Machine Learning:** What is Machine Learning, Introduction to ML's three approaches: Supervised, Unsupervised and Reinforcement Learning. **Introduction to Python:** Basic Operations**,** Lists, Tuples, Dictionaries, Flow Control, Strings, File handling, Numpy, Scikit-learn

Self-Learning Hours = 6

Students should practice calculating statistical measures and dispersion metrics on sample datasets, explore real-world applications of supervised, unsupervised, and reinforcement learning, and write Python programs using basic data structures, file handling, NumPy, and Scikit-learn to solve simple machine learning problems.

Team Activity Hours = 3

work in groups to analyze a real-world dataset by computing statistical measures, identifying outliers, and discussing suitable machine learning approaches. Each team will also develop a simple Python program using NumPy and Scikit-learn and present their findings to the rest.

Unit 2:**Introduction to Exploratory Data Analysis**

Contact Hours/Teaching Hours: 08

Introduction to Exploratory Data Analysis (EDA) –Steps in EDA, Data Types: Numerical Data – Discrete data, continuous data – Categorical data. **Data Transformation** Transformation Techniques: Performing data deduplication – replacing values – Discretization and binning. Introduction to Missing data, handling missing data. Data Visualization using Matplotlib, Seaborn

Self-Learning Hours = 6

Students should independently perform Exploratory Data Analysis (EDA) on sample datasets by identifying data types, handling missing values, applying data transformation techniques, and creating visualizations using Matplotlib and Seaborn to derive meaningful insights.

Team Activity Hours = 3

work in groups to analyze a real-world dataset, perform data cleaning and preprocessing (deduplication, missing value handling, discretization), and develop visualizations to present key patterns and findings to the rest.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Unit 3: Supervised Learning Algorithms:

Contact Hours/Teaching Hours: 8

Linear Regression, Logistic. Regression, Decision Trees, Random Forest, Support Vector Machine. K-Nearest Neighbors, CN2 Algorithm, Naive Bayes

Self-Learning Hours = 6

Students should implement and compare Linear Regression, Logistic Regression, Decision Trees, Random Forest, SVM, KNN, CN2, and Naïve Bayes algorithms on sample datasets using Python and evaluate their performance using appropriate metrics.

Team Activity Hours = 3

work in groups to solve a real-world classification or prediction problem by applying multiple machine learning algorithms, comparing their results, and presenting the most suitable model along with justification based on performance analysis.

Unit 4: Clustering: **Contact Hours/Teaching Hours: 8**

K-means, Silhoutte Scores, Hierarchical Clustering, Fuzzy c-means, DBScan. **Dimensionality Reduction:** Low Variance Filter, High Correlation Filter, Backward Feature Elimination, Forward Feature Selection, Principle Component Analysis, Projection Methods.

Self-Learning Hours = 6

Students should apply clustering techniques such as K-Means, Hierarchical Clustering, Fuzzy C-Means, and DBSCAN on real datasets, evaluate cluster quality using Silhouette Scores, and practice dimensionality reduction methods to identify the most relevant features for machine learning models.

Team Activity Hours = 3

work in groups to perform clustering and feature selection on a real-world dataset, compare the effectiveness of different clustering and dimensionality reduction techniques, and present their findings through visualizations and performance analysis.

Unit 5:

Contact Hours/Teaching Hours: 08

Model Evaluation and Selection: Cross-validation, model evaluation metrics, model selection, and hyper parameter tuning. **Hyper parameter Optimization Techniques** Manual Search, Random Search, Grid Search. Case study in Python for Hyper parameter Tuning

Self-Learning Hours = 6

Students should practice evaluating machine learning models using different performance metrics and cross-validation techniques, and implement manual search, random search, and grid search methods in Python to optimize model hyperparameters.

Team Activity Hours = 3

work in groups to develop and optimize a machine learning model on a real-world dataset, compare different hyperparameter tuning techniques, and present the impact of tuning on model performance and selection.

Course Articulation Matrix

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

Text Books:

1. "Machine Learning For Dummies", John Paul Mueller and Luca Massaron
2. "A Course in Machine Learning", Hal Daumé III.
3. "Learning scikit-learn: Machine Learning in Python", Raúl Garreta and Guillermo Moncecchi

Reference Books:

1. "Programming Collective Intelligence: Building Smart Web 2.0 Applications", Toby Segaran
2. "Building Machine Learning Systems with Python", WilliRichert and Luis Pedro Coelho
3. "Machine Learning in Action", Peter Harrington

Online Resources:

Google Developers – Machine Learning Learning Path: [https://developers.google.com/machine-learning]

Geeks For Geeks:

https://www.geeksforgeeks.org/machine-learning/machine-learning/

Kaggle Learn – Intro to Machine Learning: [https://www.kaggle.com/learn/intro-to-machine-learning]

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name : **Computer Networks**
Course Code : **TCS511**
Semester : **V**
Pre-requisite : **Algorithms, Programming in C, Operating Systems.**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- Explain the fundamental concepts, architectures, and services of computer networks and the functions of different network layers.
- Evaluate the performance and suitability of application layer protocols and network technologies for different communication requirements.
- Analyze the operation of transport layer protocols and mechanisms for reliable data transfer, flow control, and congestion control.
- Apply network addressing techniques, subnetting, and routing algorithms to design and analyze network communication systems.
- Comprehend mechanisms for error detection, flow control and multiple access.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Apply and characterize computer networks from the view point of components and from the view point of services.	**L3 – Apply**	Conceptual Understanding, Classification and Interpretation
CO2	Display good understanding of the flow of a protocol in general and a network protocol in particular	**L2 – Understand**	Understanding, Analytical Thinking
CO3	Analyse reliable data transfer protocols and incrementally develop solutions for the requirements of Transport Layer.	**L4 – Analyze**	Problem-Solving, Critical Thinking, Programming
CO4	Describe the essential principles of Network Layers and use IP addressing to create subnets for any specific requirements.	**L2 – Understand**	Knowledge and Understanding, Numerical and Logical Skills
CO5	Classify and select the appropriate technology to meet Data Link Layer requirements and design a framework to implementing TCP/IP protocol suite.	**L2 – Understand**	Knowledge and Understanding, Numerical and Logical Skills

---

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Details of the Course:

Unit 1: Introduction **Contact Hours/Teaching Hours: 11**

Computer Networks and the Internet; What is internet: a nuts-and-bolts description, a services description, what is a protocol, The Network edge: Access Networks and Physical Media, The Network Core: Packet and Circuit Switching, Difference between Virtual Circuits and Datagram networks,Delays: Processing, Queuing, Transmission and Propagation, Delay in Circuit-Switched Networks and Packet-Switched Networks, End-to-End Delay, Throughput in Computer Networks, Protocol Layers and Their Service Models: Layered Architecture, Encapsulation, The OSI Reference Model and the TCP/IP protocol stack

Self-Learning Hours = 6

Students shall practice numerical examples in delay in networks, working of layered architecture.

Team Activity Hours = 3

Group discussion on how computer network works and working of various network components.

Unit 2: Application Layer

Contact Hours/Teaching Hours: 11

Principles of Network Applications: Network Application Architectures, Processes Communicating, Transport Services Available to Applications, transport Services Provided by the Internet, Application-Layer Protocols, The Web and HTTP: Overview of HTTP, Non- Persistent and Persistent Connections, HTTP Message Format, conditional GET, HTTP/2 and HTTP/3, User-Server Interaction: Cookies, Web Caching,File Transfer Protocol, Electronic Mail in the Internet: SMTP, mail message formats, mail access protocols, DNS-The Internet's Directory Service: Services Provided by DNS, Overview of How DNS Works, DNS Records and Messages, Peer-to-Peer File Distribution: BitTorrent, Video Streaming and Content Distribution Networks: Internet Video, HTTP Streaming and DASH, Content Distribution Networks, Case Studies: Netflix and YouTube

Self-Learning Hours = 6

Students shall use various network related CLI commands to understand the transfer of data from one place to other over a computer network.

Team Activity Hours = 3

Group-based activity is on solving practical problems of establishing a LAN and to configure DNS servers.

Unit 3: Transport Layer

Contact Hours/Teaching Hours: 6

Transport-Layer Services: Relationship Between Transport and Network Layers, Overview of the Transport Layer in the Internet, Multiplexing and Demultiplexing, Connectionless Transport-UDP: Segment structure, checksum in UDP, QUIC, Transport Layer Protocols: Simple Protocol, Stop-and-Wait Protocol, Go-Back-N (GBN), Selective Repeat (SR); Connection-Oriented Transport-TCP: The TCP Connection, TCP Segment Structure, TCP Timers, Round-Trip Time Estimation and Timeout, Reliable Data Transfer, Flow Control, Error Control, TCP Connection Management; Principles of Congestion Control, TCP Congestion Control

Self-Learning Hours = 6

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Students shall calculate the round trip time; understand the fields in TCP and UDP headers.

Team Activity Hours = 3

Group-based activity is to design a program for a client-server program interaction between two different processes present in different computers connected over a network.

Unit 4: Network Layer

Contact Hours/Teaching Hours: 6

Packet Forwarding and Routing, Delivery and Forwarding of IP Packets, Structure of a Router, The Internet Protocol(IPv4): Datagram format, Fragmentation, Addressing-Classful and Classless, subnets, DHCP, Network Address Translation(NAT), Internet Control Message Protocol(ICMP), ARP, IPv6:Datagram format, Moving from IPv4 to IPv6: Dual Stack, Tunneling, Header translation, Generalized Forwarding and SDN: Match, Action.
Inter and Intra domain routing, Routing Algorithms: Adaptive and Non-adaptive, The Link State(LS) Routing Algorithm, The Distance Vector (DV) Routing Algorithm, Introduction to Routing in the Internet: RIP, OSPF, BGP; Introduction to Broadcast and Multicast Routing, SNMP

Self-Learning Hours = 6

Students shall implement NAT, DHCP and routing protocols.

Team Activity Hours = 3

Group activity on subnetting and supernetting by distributing the available IP dresses among various users.

Unit 5: Link Layer

ContactHours/Teaching Hours: 08

Link Layer and Local Area Networks- Introduction to Link Layer and its services, Where Link Layer is implemented? Error detection and correction techniques: Parity checks, Checksum, CRC; Multiple Access protocols: Channel Partitioning, Random Access (Slotted Aloha, Aloha, CSMA), Taking Turns; Link Layer Addressing: MAC addresses, Ethernet, Link Layer Switches, VLANS

Self-Learning Hours = 6

Students shall implement Parity checks, Checksum, CRC with help of C/Java programming.

Team Activity Hours = 3

Case study and group presentation on real-world applications of LANs, MANs, WANs and VLANs.

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		1	2	1	2
CO02	3	2			2					1		1	3	1	2
CO03	3	3	2	2	2					1		1	3	2	3
CO04	3	3	2		2					1		1	3	2	2
CO05	3	3	3	2	3				1	1		1	3	2	3
Avg.	3	2.6	2	2	2.2				1	1		1	2.2	2.6	1.4

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Text Books:

1. Ross and Kurose, Computer Networking: "A Top Down Approach (8th edition)", 8/e, Pearson/Addison-Wesley, 2022

Reference Books:

1. Andrew Tanenbaum and David Wetherhall, Computer Networks, 5/e, Prentice Hall, 2010
2. Peterson and Davie, Computer Networks: A System Approach, 4/e, Elsevier India, 2007
3. Forouzan, Data Communication and Networking, 5/e, McGraw Hill Education, 2013
4. William Stallings, Data and Computer Communication, 8/e, Pearson/Addison-Wesley, 2007

Online Resources:

1. Cisco Networking Academy
   https://www.netacad.com/networking

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Computer Networks - I**
Course Code	:	**TCS 514**
Semester	:	**V**
Pre-requisite	:	**Fundamental of Computer & Introduction to Programming (TCS 101), Data Structures with C (TCS 302)**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- Explain the fundamentals of computer networks, Internet architecture, protocol layering, and the OSI and TCP/IP reference models.
- Apply application layer protocols and socket programming concepts to support network communication and Internet-based applications.
- Analyze the operation of UDP and TCP protocols for reliable data transfer, flow control, and congestion control.
- Analyze IP addressing, subnetting, routing, DHCP, NAT, and IPv4/IPv6 mechanisms for efficient network communication.
- Apply networking concepts and tools to design and evaluate communication solutions for real-world networking 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 computer networks, Internet architecture, protocol layering, and the OSI and TCP/IP reference models.	**L2 – Understand**	Basic networking knowledge and analytical skills.
CO2	Apply application layer protocols (HTTP, FTP, SMTP, DNS, BitTorrent) and socket programming concepts to analyze network applications and communication services.	**L3 – Apply**	Protocol knowledge and programming skills.
CO3	Analyze transport layer protocols, including UDP and TCP, to ensure reliable data transfer, flow control, and congestion control in computer networks.	**L4 - Analyze**	Analytical and problem-solving skills.
CO4	Analyze IP addressing, subnetting, routing, DHCP, NAT, IPv4/IPv6, and network layer services for efficient packet forwarding and communication.	**L4 – Analyze**	Networking and analytical skills.
CO5	Apply networking concepts and tools to design and evaluate network communication solutions for real-world applications.	**L3 – Apply**	Networking and design skills.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Details of the Course:

Unit 1: Introduction to Computer Network **Contact Hours/Teaching Hours: 11**

Introduction: Computer Networks and the Internet, Overall view: As components and as services; What is a protocol, what is a network protocol, Access Networks and Physical Media, Circuit and Packet Switching, Internet Backbone, Delays: Processing, Queing, Transmission and Propagation delays.
The Layered Architecture: Protocol Layering, The OSI Reference Model and the TCP/IP protocol stack, History of Computer Networking and the Internet.

Self-Learning Hours = 7

Study networking fundamentals, protocol layering, and compare the OSI and TCP/IP models through examples.

Team Activity Hours = 3

Discuss network architecture, switching techniques, and analyze network delay scenarios in groups.

Unit 2: Application Layer **Contact Hours/Teaching Hours: 12**

Application Layer: Principles and Architectures of Network Applications, Client and Server processes, the idea of socket, Transport services available to Application Layer especially in the Application Layer Protocols: The Web and http: Persistent and Nonpersistent connections, http message format, cookies, proxy server, conditional GET.
File Transfer Protocol.
Email: smtp, mail message formats, mail access protocols: pop3, imap, MIME.
DNS: Services, How it works, Root, Top-Level and Authoritative DNS servers, Resource Records, DNS messages.
A simple introduction to p2p file distribution: BitTorrent.

Self-Learning Hours = 8

Study application layer protocols and analyze HTTP, FTP, Email, DNS, and BitTorrent through real-world examples.

Team Activity Hours = 3

Demonstrate the working of application layer protocols and discuss client-server communication in groups.

Unit 3: Transport Layer **Contact Hours/Teaching Hours: 06**

Transport Layer: Introduction and Services, The Transport layer in internet, Difference between Connection Oriented and Connectionless services.
UDP: Segment structure, checksum in UDP

Self-Learning Hours = 4

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Study transport layer services, UDP communication, and practice checksum calculations.

Team Activity Hours = 2

Compare connection-oriented and connectionless services and solve UDP checksum examples in groups.

Unit 4: Transport Layer 2 **Contact Hours/Teaching Hours: 06**

Transport Layer2: The principles behind connection oriented data transfer, designing a connection oriented protocol, stop-and-wait, Go Back N, Selective Repeat.
TCP: Connection Establishment, TCP header, Sequence and acknowledgement numbers, Round Trip Time, Flow Control, Congestion Control.

Self-Learning Hours = 4

Study reliable data transfer protocols, TCP features, and solve problems on flow control, congestion control, and RTT estimation.

Team Activity Hours = 2

Analyze TCP communication and compare Stop-and-Wait, Go-Back-N, and Selective Repeatprotocols through group discussions.

Unit 5: Network Layer 1 **Contact Hours/Teaching Hours: 10**

Network Layer I: Introduction, Packet Forwarding and Routing, Difference between Virtual Circuits and Datagram networks, The internals of a router: Input ports, output ports, switching architecture.
The Internet Protocol(IP), Datagram format, IP fragmentation, IPv4 addressing, subnets, CIDR, classful addressing, DHCP, Network Address Translation(NAT), Universal Plug and Play as a provider of NAT, Internet Control Message Protocol(ICMP), IPv6 Header, Moving from IPv4 to IPv6: tunnelling, A brief discussion on IP security.

Self-Learning Hours = 6

Study IP addressing, subnetting, routing, NAT, and IPv6 concepts through examples and numerical problems.

Team Activity Hours = 3

Solve subnetting exercises and discuss routing, DHCP, and NAT implementation using real-world scenarios.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO01	3	2			1							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	3	2	3	2	3				2	2		2	3	3	2
Avg.	3	2	3	2	3	0	0	0	2	2	0	1	3	3	2

Text Books:

1. Computer Networking: “A Top Down Approach (5th edition)”, Ross and Kurose, Pearson/Addison-Wesley

Reference Books:

1. Andrew Tanenbaum and David Wetherhall, “Computer Networks(5th edition)”, Prentice Hall.
2. Peterson and Davie, “Computer Networks: A System Approach (4th edition)”, Elsevier
3. Forouzan, “Data Communication and Networking (4th edition)”, McGraw Hill
4. William Stallings: “Data and Computer Communication”, 8th Edition, Pearson Education, 2007.
5. Nader F. Mir:” Computer and Communication Networks”, Pearson Education, 2007.

Online Resources:

1. **GeeksforGeeks – Computer Networks**
   https://www.geeksforgeeks.org/computer-networks/basics-computer-networking/
2. **Coursera – The Bits and Bytes of Computer Networking**
   https://www.coursera.org/learn/computer-networking
3. **NPTEL – Emergence of Network and Reference Model**
   https://nptel.ac.in/courses/106105081

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 : **Natural Language Processing and Computer Vision**
Course Code : **TCS564**
Semester : **V**
Pre-requisite : **Python programming (TCS 346), Fundamentals of Machine Learning, Basics of Artificial Intelligence (TCS 364)**

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To introduce the fundamental concepts of Natural Language Processing (NLP), including text preprocessing, text normalization, and common NLP applications.
- To explain syntax analysis, language modeling, and feature extraction techniques used in Natural Language Processing.
- To develop the ability to analyze and implement sentiment analysis techniques and NLP applications using modern computational approaches.
- To provide an understanding of Computer Vision fundamentals, digital image processing, feature detection, and image segmentation techniques.
- To enable students to evaluate and implement deep learning-based computer vision techniques, object detection frameworks, transfer learning, and advanced vision 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 Natural Language Processing (NLP), including text preprocessing, text normalization, regular expressions, and common NLP applications.	**BL1, BL2**	Understanding, Text Processing
CO2	Apply NLP techniques such as part-of-speech tagging, syntactic and dependency parsing, language modeling, and feature extraction using Bag of Words, TF-IDF, and word embeddings.	**BL2, BL3**	NLP Programming,

			Problem Solving
CO3	Analyze sentiment analysis techniques and implement NLP applications including machine translation, speech recognition, and chatbots using appropriate methods and tools.	**BL3, BL4**	AI Programming, Application Development
CO4	Describe the fundamentals of Computer Vision, image processing techniques, feature detection, image segmentation, and image representation using various color spaces.	**BL2, BL3**	Image Processing, Analytical Thinking
CO5	Evaluate and implement object detection, deep learning-based computer vision models (CNNs, R-CNN, YOLO, SSD), transfer learning, and advanced vision applications such as object tracking, 3D vision, and autonomous systems.	**BL4, BL5**	Deep Learning, Model Development

Details of the Course:

UNIT 1: Introduction to NLP, Text Processing

Contact Hours/Teaching Hours: 9

Overview of NLP and its significance, Key challenges in NLP, Common NLP tasks and applications, Text normalization: tokenization, stemming, lemmatization, Regular expressions for pattern matching, Stop words removal and text cleaning.

Self-Learning Hours – 5

Students shall practice text preprocessing techniques using Python libraries on real-world datasets.

Team Activity Hours – 2

Group discussion on identifying suitable preprocessing techniques for different NLP applications.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

UNIT 2: Syntax and Parsing, Language Models

Contact Hours/Teaching Hours: 10

Part-of-speech tagging, Syntactic parsing, Dependency parsing, Introduction to n-grams, Smoothing techniques, Introduction to neural language models Feature Extraction from Text Bag of words and TF-IDF, Word embeddings: Word2Vec, GloVe- Contextual embeddings from transformers.

Self-Learning Hours – 5

Students shall implement feature extraction and word embedding techniques using Python libraries.

Team Activity Hours – 2

Group activity to compare the performance of different feature extraction and embedding methods.

UNIT 3: Sentiment Analysis **Contact Hours/Teaching Hours: 9**

Techniques for sentiment analysis, Rule-based and machine learning approaches, Application of sentiment analysis in business and social media NLP Applications, Machine translation, Speech recognition, Chatbots and conversational agents

Self-Learning Hours – 5

Students shall build a basic sentiment analysis model using publicly available datasets.

Team Activity Hours – 2

Group discussion on real-world NLP applications and presentation of sentiment analysis case studies.

UNIT 4: Introduction to Computer Vision, Image Processing Basics

Contact Hours/Teaching Hours: 10

Overview of computer vision and its significance, Key challenges in computer vision, Common applications of computer vision, Digital image fundamentals, Image manipulation techniques (filtering, transformations), Color spaces and conversions, Feature Detection and Matching, Image Segmentation

Edge detection algorithms, Corner and interest point detection, Feature matching and object recognition, Thresholding techniques, Region-based segmentation, Clustering methods in segmentation (k-means, mean shift).

Self-Learning Hours – 6

Students shall implement basic image processing and feature detection techniques using OpenCV.

Team Activity Hours – 2

Group activity involving comparison of different image segmentation techniques on sample datasets.

UNIT 5: Object Detection and Recognition

Contact Hours/Teaching Hours: 10

Introduction to object detection frameworks, Haar cascades and HOG features, Modern approaches: R-CNN, Fast R-CNN, YOLO, and SSD, Deep Learning for Vision, Convolutional neural networks (CNNs) for vision, Transfer learning in vision tasks, Applications in real-time video processing, Advanced Topics and Applications, 3D vision and depth estimation, Motion analysis and object tracking, Computer vision in autonomous vehicles and drones.

Self-Learning Hours – 6

Students shall implement object detection models using pre-trained deep learning frameworks and analyze their performance.

Team Activity Hours – 2

Group project involving the development and demonstration of a real-time computer vision application using modern object detection techniques.

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	3	3	2	2	3	–	–	–	1	1	1	–	3	3	2
CO03	3	3	2	2	3	–	–	1	2	1	1	–	3	3	2
CO04	3	3	3	2	3	–	–	1	2	1	1	–	3	3	2
CO05	3	3	3	3	3	–	1	1	2	2	2	1	3	3	3
Avg.	3.0	2.8	2.2	2.0	2.8	0.0	0.2	0.6	1.6	1.2	1.2	0.2	2.8	2.8	2.0

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

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Text Book:

1. Steven Bird, Ewan Klein, and Edward Loper, Natural Language Processing with Python, 3rd Edition, O'Reilly Media, Inc., USA, 2021.
2. Daniel Jurafsky and James H. Martin, Speech and Language Processing, 3rd Edition, Pearson Education Limited, USA, 2020.
3. Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd Edition, Springer Science, New York, USA, 2011.
4. Joseph Howse, Joe Minichino, and Prateek Joshi, Learning OpenCV 4: Computer Vision with Python 3, 1st Edition, Packt Publishing Ltd., UK, 2019.

References:

1. Steven Bird, Ewan Klein, and Edward Loper, Natural Language Processing with Python, 3rd Edition, O'Reilly Media, Inc., USA, 2021.
2. Daniel Jurafsky and James H. Martin, Speech and Language Processing, 3rd Edition, Pearson Education Limited, USA, 2020.
3. Richard Szeliski, Computer Vision: Algorithms and Applications, 2nd Edition, Springer Science, New York, USA, 2011.
4. Joseph Howse, Joe Minichino, and Prateek Joshi, Learning OpenCV 4: Computer Vision with Python 3, 1st Edition, Packt Publishing Ltd., UK, 2019.

Online Resources:

1. **GeeksforGeeks – Natural Language Processing (NLP)**
   https://www.geeksforgeeks.org/natural-language-processing-nlp-tutorial/
2. **NPTEL – Natural Language Processing**
   https://nptel.ac.in/
3. **OpenCV Official Documentation**
   https://docs.opencv.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	**:**	**Foundation of Quantum Computing**	**L**	**T**	**P**	**Credits**
Course Code	**:**	**TCS584**	**3**	**1**	**0**	**3**
Semester	**:**	**III**				
Pre-requisite	**:**	**Programming for problem solving, Python Programming**				

Course Learning Objectives:

- To develop an understanding of the fundamental principles, mechanisms, and computational paradigms of quantum computing.
- To enable learners to analyze quantum computing architectures, quantum gates, circuits, and algorithms for solving computational problems.
- To provide practical exposure to quantum programming languages and software frameworks for implementing quantum algorithms.
- To develop the ability to apply quantum computing techniques for solving problems in cryptography, optimization, machine learning, and other emerging application domains.
- To foster the skills required to design and develop quantum computing solutions by integrating quantum algorithms, programming methodologies, and application-specific requirements.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Understand techniques and mechanism of quantum computing.	**L2 – Understand**	Understanding, Conceptual Learning
CO2	Understand the working principles of various quantum computing techniques.	**L3 – Apply**	Understanding, Analytical Thinking
CO3	Analyze different quantum computing techniques and algorithms.	**L3 – Apply**	Analytical Thinking, Critical Thinking

CO4	Use quantum programming languages to implement quantum computing techniques and algorithms.	**L4 – Analyze**	Programming Skills, Computational Thinking, Problem Solving
CO5	Apply quantum computing techniques to solve problems in various application domains.	**L3 – Apply**	Problem Solving, Application Skills, Decision Making
CO6	Develop quantum computing schemes for real-world computational applications.	**L6 – Create**	Design Thinking, Innovation, Solution Development

Details of the Course:

Unit 1: Introduction to Quantum Mechanics for Quantum Computing

Contact Hours/Teaching Hours: 9

History of quantum computation and quantum information, Classical vs. quantum systems, Key quantum phenomena: Superposition, entanglement, and interference, Dirac notation (Bra-Ket notation, inner product), Postulates of quantum mechanics, Qubits and quantum state representation, Single-qubit gates (Pauli-X, Pauli-Y, Pauli-Z, Hadamard).

Self-Learning Hours = 6

Students shall study the fundamentals of quantum mechanics, represent quantum states using Dirac notation, and analyze the behavior of single-qubit quantum gates.

Team Activity Hours = 3

Students shall compare classical and quantum computing paradigms, demonstrate the concepts of superposition and entanglement through simulations, and present their observations.

Unit 2: **Mathematical Foundation of Quantum Computing**

Contact Hours/Teaching Hours: 10

Linear algebra for quantum computing (vectors, matrices, tensor products), Hilbert spaces and unitary operators, Quantum measurement (Born rule, projective measurement), Mixed states and density matrices, No-cloning theorem.

Self-Learning Hours = 6

Students shall solve problems involving vector spaces, tensor products, and unitary transformations, and explore quantum measurement principles through numerical examples.

Team Activity Hours = 3

Students shall collaboratively solve linear algebra problems related to quantum computing, discuss the significance of the no-cloning theorem, and present their findings.

Unit 3: Quantum Circuits and Multi-Qubit Systems

Contact Hours/Teaching Hours: 09

Multi-qubit systems and entanglement (Bell states, EPR pairs, applications), Two-qubit gates (CNOT, SWAP, Toffoli), Quantum circuit model, Quantum parallelism, Quantum teleportation and superdense coding, Quantum teleportation protocol.

Self-Learning Hours = 6

Students shall design basic quantum circuits, simulate multi-qubit operations, and analyze quantum teleportation and superdense coding protocols.

Team Activity Hours = 3

Students shall develop quantum circuits using simulation tools, compare different quantum gates, and demonstrate quantum communication protocols through group activities.

Unit 4: Quantum Algorithms

Contact Hours/Teaching Hours: 10

Deutsch-Jozsa algorithm, Bernstein-Vazirani algorithm, Simon's algorithm, Grover's search algorithm (overview and applications), Shor's factoring algorithm (conceptual framework).

Self-Learning Hours = 6

Students shall implement introductory quantum algorithms using quantum programming frameworks and compare their computational advantages over classical algorithms.

Team Activity Hours = 3

Students shall analyze the working of major quantum algorithms, compare their computational complexity, and present suitable real-world applications.

Unit 5: Quantum Error Correction and Applications

Contact Hours/Teaching Hours: 10

Challenges in quantum computing (decoherence, noise), Basics of quantum error correction (QEC), Stabilizer codes (Shor code, surface codes), Overview of quantum hardware (NISQ era, superconducting qubits, trapped ions), Current and future applications: Quantum cryptography (BB84 protocol), Optimization, Quantum machine learning (Variational Quantum Eigen solver (VQE)).

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Self-Learning Hours = 6

Students shall study quantum error correction techniques, explore current quantum hardware platforms, and investigate emerging applications of quantum computing.

Team Activity Hours = 3

Students shall review recent developments in quantum technologies, compare quantum hardware platforms and error correction techniques, and present case studies on real-world quantum computing applications.

Course Articulation Matrix

CO	Statement	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
TCS584.1	Explain techniques and mechanism of quantum computing	2										1	1	2	1
TCS584.2	Know the working of various quantum computing techniques		2	1		1						1	2	1	1
TCS584.3	Analyze the different quantum computing techniques		2	1		1						1	2	1	1
TCS584.4	Use programming language to implement quantum	2			1	2						1	1	2	1

	computing techniques/algorithms														
TCS584.5	Apply quantum computing techniques to various applications		2	1								1	2	1	1
TCS584.6	Develop quantum computing schemes	2	1								1	2	1	1	1
TCS584		2	1.75	1.00	1.00	1.33	-	-	-	-	1.00	1.17	1.50	1.33	1.00

Text Books:

1. Quantum Computation and Quantum Information, Nielsen, M. A., & Chuang, I. L. (2010). *Quantum Computation and Quantum Information* (10th Anniversary Edition). Cambridge University Press.
2. Quantum Computing: An Applied Approach, Hidary, J. D. (2021). *Quantum Computing: An Applied Approach* (2nd Edition). Springer.

Reference Books:

1. Programming Quantum Computers, Johnston, E. R., Harrigan, N., & Gimeno-Segovia, M. (2019). *Programming Quantum Computers: Essential Algorithms and Code Samples*. O'Reilly Media.
2. An Introduction to Quantum Computing, Kaye, P., Laflamme, R., & Mosca, M. (2007). *An Introduction to Quantum Computing*.
3. Quantum Computer Science: An Introduction
Mermin, N. D. (2007). *Quantum Computer Science: An Introduction*. Cambridge University Press.

Online Resources:

1. **NPTEL – Introduction to Quantum Computing**
   https://onlinecourses.nptel.ac.in
2. **SWAYAM – Quantum Computing and Emerging Technologies Courses**
   https://swayam.gov.in

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Exploratory Data Analysis**
Course Code	:	**TCS 572**
Semester	:	**V**
Pre-requisite	:	**NA**

L	T	P	Credits
3	1	0	3

Course Learning Objectives:

- To introduce students to the fundamental concepts, goals, assumptions, and significance of Exploratory Data Analysis and develop an understanding of the consequences of violations of underlying data assumptions.
- To develop the ability to apply graphical and quantitative EDA techniques for data visualization, pattern identification, relationship analysis, statistical hypothesis testing, and outlier detection.
- To enable students to analyze probability distributions and estimate distribution parameters using statistical estimation methods, probability plots, and probability distribution tables.
- To build competence in analyzing and summarizing time-series and multidimensional data using statistical measures, grouping, resampling, data transformation, deduplication, and missing-data handling techniques.
- To develop the ability to apply and evaluate dimensionality-reduction and nonlinear learning techniques, and explore reinforcement-learning concepts and EDA methods through real-world case studies.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO01	Explain the fundamental concepts, goals, assumptions, and significance of Exploratory Data Analysis and interpret the consequences of violations of underlying data assumptions.	**L2 – Understand**	Statistical Understanding, Data Interpretation
CO02	Apply appropriate graphical and quantitative EDA techniques to analyze data, identify patterns, examine relationships, test statistical hypotheses, and detect outliers.	**L3 – Apply**	Data Visualization, Statistical Analysis
CO03	Analyze probability distributions and estimate their parameters using suitable statistical methods, probability plots, and distribution tables.	**L4 – Analyze**	Probability Modelling, Parameter Estimation
CO04	Analyze and summarize time-series and multidimensional data using statistical measures, grouping, resampling, contingency tables, data transformation, deduplication, and missing-data handling techniques.	**L4 – Analyze**	Data Preprocessing, Multidimensional Data Analysis
CO05	Apply and evaluate dimensionality-reduction and nonlinear learning techniques, and examine reinforcement-learning concepts and EDA methods through real-world case studies.	**L5 – Evaluate**	Feature Reduction, Analytical Evaluation

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Details of the Course:

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

What is EDA?. How Does Exploratory Data Analysis differ from Classical Data Analysis? : Model, Focus, Techniques, Rigor, Data Treatment, Assumptions. How Does Exploratory Data Analysis Differ from Summary Analysis? What are the EDA Goal. Software tools for EDA. EDA Assumptions: Underlying Assumptions. Importance. Techniques for Testing Assumptions. Interpretation of 4-Plot. Consequences: Consequences of Non-Randomness, Consequences of Non-Fixed Location Parameter, Consequences of Non-Fixed Variation Parameter, Consequences Related to Distributional Assumptions.

Self-Learning Hours = 4

Students shall explore a real-world dataset using suitable EDA software tools, examine underlying assumptions, and interpret four-plot visualizations to identify non-randomness, changes in location and variation, and deviations from distributional assumptions.

Team Activity Hours = 2

Group discussion and presentation comparing EDA with classical and summary analysis, followed by interpretation of assumption violations in sample datasets

Unit 2: EDA Techniques

Contact Hours/Teaching Hours: 10

Introduction. Graphical Techniques: Histogram, Box Plot, Scatter Plot, Q–Q/Normal Plot, Correlation Plots, Contour Plot, Violin Plot, Autocorrelation, Lag, Spectral, Run-Sequence.

Quantitative Techniques: Measures of Location (mean, median, mode), Confidence Limits for the Mean, Two-Sample *t*-Test for Equal Means, One-Factor ANOVA, Multi-factor Analysis of Variance, Measures of Scale, Chi-Square Test for the Standard Deviation, Levene Test for Equality of Variances, Measures of Skewness and Kurtosis, Kolmogorov-Smirnov Goodness-of-Fit Test, Grubbs' Test for Outliers.

Self-Learning Hours = 5

Students shall apply graphical techniques such as histograms, box plots, scatter plots, Q–Q plots, violin plots, correlation plots, autocorrelation plots, and quantitative statistical tests on suitable datasets.

Team Activity Hours = 3

Group-based analysis of a real-world dataset using graphical and quantitative EDA techniques to identify patterns, relationships, distribution characteristics, variance differences, and outliers.

Unit 3: Probability Distributions:

Contact Hours/Teaching Hours: 11

What is a Probability Distribution, Related Distributions, Families of Distributions, Location and Scale Parameters, Estimating the Parameters of a Distribution: Method of Moments, Maximum Likelihood, Least Squares, PPCC and Probability Plots.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Gallery of Distributions: Normal Distribution, Uniform Distribution, t Distribution, F Distribution, Chi-Square Distribution, Binomial Distribution, Poisson Distribution, Exponential Distribution, Weibull Distribution, Lognormal Distribution, Gamma Distribution, Beta Distribution.

Tables for Probability Distributions: CDF of Standard Normal Distribution, Upper Critical Values of the Student's-t Distribution, Upper Critical Values of the F Distribution, Critical Values of the Chi-Square Distribution, Critical Values of the t-* Distribution, Critical Values of the Normal PPCC Distribution.

Self-Learning Hours = 5

Students shall explore and compare different probability distributions, estimate distribution parameters using suitable methods, and interpret probability plots and distribution tables

Team Activity Hours = 3

Group activity to identify and fit suitable probability distributions to different real-world datasets and present the results using probability plots, parameter estimates, and goodness-of-fit analysis.

Unit 4: Analysis and Summarization:

Contact Hours/Teaching Hours: 7

Fundamentals of Time series analysis – characteristics, grouping, resampling. Statistical summary measures, data elaboration, 1-D Statistical data analysis, 2-D Statistical data Analysis, contingency tables, n-D Statistical data analysis. Transformation: Performing data deduplication - replacing values Introduction to Missing data, handling missing data.

Self-Learning Hours = 4

Students shall perform time-series grouping and resampling, compute statistical summaries, analyze one-dimensional and multidimensional data, and apply techniques for handling duplicate and missing values.

Team Activity Hours = 2

Group-based data cleaning and summarization activity using a real-world dataset involving missing-value treatment, deduplication, transformation, contingency tables, and statistical interpretation.

Unit 5: Dimensionality Reduction:

Contact Hours/Teaching Hours: 05

Principal Component Analysis (PCA), Singular Value Decomposition, Factor Analysis, Intrinsic Dimensionality. Non-Linear methods: Multidimensional Scaling, Manifold Learning, Self-Organizing Maps. Understanding reinforcement learning: Difference between supervised and reinforcement learning, Applications of reinforcement learning. EDA Case Studies.

Self-Learning Hours = 5

Students shall apply and compare PCA, SVD, Factor Analysis, Multidimensional Scaling, and Manifold Learning techniques on suitable datasets and study introductory reinforcement-learning applications.

Team Activity Hours = 2

Case study and group presentation comparing dimensionality-reduction and nonlinear learning techniques on a real-world dataset, including interpretation and visualization of the reduced feature space.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO01	3	2			2	-	-	-	-	-	-	2	2	2	1
CO02	2	3	2	3	3	-	-	-	-	-	-	2	2	3	2
CO03	3	3	2	2	2	-	-	-	-	-	-	2	2	3	2
CO04	2	3	2	3	3	-	-	-	-	-	-	2	3	3	2
CO05	2	3	3	3	3	-	-	-	-	-	-	2	2	3	3
Avg.	3	3	2	2	3	0	0	0	0	0	0	2	2	3	2

Text Books:

1. Engineering Statistics handbook, NIST 1st edition, 2012.
2. Suresh Kumar Mukhiya, Usman Ahmed, "Hands-On Exploratory Data Analysis with Python", Packt Publishing, 1st edution, 2020.

Reference Books:

1. Martinez, W , Martinez A & J.L. Solka : Exploratory Data Analysis with MATLAB, CRC Press, A Chapman & Hall Book ,3rd Edition (2017).
2. Michael Jambu, "Exploratory and multivariate data analysis", Academic Press Inc. 1st edition, 1991.

Online Resources:

1. **GeeksforGeeks – EDA**
   https://www.geeksforgeeks.org/data-analysis/exploratory-data-analysis-in-python/

2. **NPTEL – EDA with R**
   https://nptel.ac.in/

3. **Kaggle- EDA Learning Path**
   https://www.kaggle.com/discussions/general/570044

4. **Coursera- EDA**
   https://www.coursera.org/courses?query=exploratory%20data%20analysis

5. **IIT Kanpur – online Course Material-EDA**
   https://home.iitk.ac.in/~shalab/onlinecoursematerial

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name : **Security Audit and Compliance-I**
Course Code : **TCS 595**
Semester : **V**
Pre-requisite : Fundamental of Cyber Security (TCS-492).

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To enable students to describe the foundational principles of cybersecurity, identify various types of malware and evolving cyber threats, analyse social engineering tactics, and implement basic preventive security practices to secure personal and organizational environments.
- To familiarize students with legal and ethical frameworks in cyberspace, international information security standards (such as ISO/IEC 27001, NIST), and compliance policies, enabling them to evaluate cyber laws and execute effective information security risk assessment and management practices.
- To equip students with the skills required to implement cryptographic techniques, configure robust authentication and authorization mechanisms, design access control models, and deploy secure communication protocols to protect data in transit and restrict unauthorized access
- To train students to design and maintain secure database architectures, manage data security throughout its lifecycle, and establish robust continuous monitoring, auditing, backup, and disaster recovery strategies to ensure business continuity.
- To empower students to systematically evaluate systems and databases for vulnerabilities using security scanning tools, apply standard penetration testing methodologies, perform security audits, and document compliance reports to mitigate organizational security risks.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Explain the fundamentals of cybersecurity, cyber threats, malware, social engineering attacks, and preventive security practices.	**L2 – Understand**	Articulation of security risk vectors.
CO2	Interpret cybersecurity governance, cyber ethics, cyber laws, information security standards, compliance frameworks, and information security risk management practices.	**L2 – Understand**	Conceptual mapping of exploit mechanics
CO3	Apply secure communication techniques including authentication, authorization, access control mechanisms, encryption methods, and secure communication protocols.	**L3: Apply**	Critical evaluation, identifying architectural weaknesses

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

CO4	Analyze database security mechanisms, data lifecycle management, auditing, monitoring, backup, recovery, and disaster recovery strategies.	**L3: Apply**	Secure coding practices
CO5	Evaluate database security through auditing, vulnerability assessment, penetration testing methodologies, and security scanning tools to ensure organizational compliance and security.	**L4 – Analyse**	Debugging security controls

---

Details of the Course:

Unit 1: Introduction to Cybersecurity : **Contact Hours/Teaching Hours: 09**

Definition of Netizen, Malware and its existence, Definition of Security hole, Security Patch, Viruses, Worms, Trojan Horses, Bot Networks, Social Engineering, Avoiding Malwares, Spyware, Adware, Keyboard Loggers, Rogue Software and Scareware, Ransomware, White Hat Search Engine Optimization, Current and Fulltime Threats, Hackers, Hacker's tools, E-Mail and SPAM, Spoofing, Spammer's tools, SPIM, Cyberbullies, Online Reputation Attacks, Phishing, Cyber stalkers
.

Self-Learning Hours = 6

Students shall independently explore the characteristics of modern malware strains (such as ransomware and keyloggers) and analyze case studies of high-profile social engineering and phishing attacks.

Team Activity Hours = 3

In groups, students will conduct a passive open-source intelligence (OSINT) gathering exercise on a mock target entity to blueprint potential vectors for social engineering.

Unit 2: Cyber Ethics, Cyber Law , Cyber Policy and Compliance:

Contact Hours/Teaching Hours: 11

Cybersecurity Management Concepts, Security governance, Management models, roles, and functions, Enterprise Roles and Structures, Information security roles and positions, Strategic planning and security strategy, Information Security Standards & Laws, Information Security Standards-ISO, IT Act, Copyright Act, Patent Law, IPR, Cyber Laws in India; IT Act 2000 Provisions, Intellectual Property Law: Copy Right Law, Software License, Patent Law; GDPR, HIPAA, SOX, NIST, ISO27001
.

Self-Learning Hours = 6

Students shall independently research the regulatory requirements and organizational impact of global and national compliance frameworks (including the IT Act 2000, GDPR, HIPAA, and NIST CSF)..

Team Activity Hours = 3

Groups will be assigned a hypothetical corporate data breach scenario. One part of the group will act as corporate security officers mapping the breach against ISO 27001/GDPR checklists to defend their strategy, while the other part acts as compliance auditors analyzing structural gaps in the enterprise's security governance, strategic planning, and legal liabilities.

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Unit 3: Secured Communication: **Contact Hours/Teaching Hours: 10**

Access Control and Authentication, User authentication methods (e.g., passwords, biometrics, multi-factor authentication), Role-based access control (RBAC), Access control lists (ACLs), Data Encryption, Encryption of data at rest and in transit, Transparent Data Encryption (TDE), Encryption - Symmetric vs. asymmetric encryption; SSL/TLS encryption for web traffic, Encryption algorithms (e.g., AES, RSA).

Self-Learning Hours = 6

Students shall independently configure local Access Control Lists (ACLs) and implement Role-Based Access Control (RBAC) matrices in a simulated environment.

Team Activity Hours = 3

A collaborative lab exercise where groups design a multi-factor authentication (MFA) and data-in-transit encryption architecture for a remote banking application..

Unit 4: Database Security: **Contact Hours/Teaching Hours: 08**

Data lifecycle: Data roles, classification, maintenance, retention, collection, destruction, Database Auditing and Monitoring, Database activity monitoring (DAM), Audit trails and logs, Database security policies and procedures.
Backup and Recovery, Secure backup and recovery procedures, Disaster recovery planning, Off-site backup storage
.

Self-Learning Hours = 6

Students will research enterprise-grade Database Activity Monitoring (DAM) systems and draft a standardized Disaster Recovery (DR) policy featuring backup retention schedules and off-site storage logistics.

Team Activity Hours = 3

Teams will participate in a interactive simulation of a catastrophic ransomware attack that compromises a production database system. The groups must work together under a ticking clock to execute a Disaster Recovery Plan, determining data classification priorities, validating audit trail integrity, and managing backup recovery strategies under constrained timelines..

Unit 5: Auditing & Monitoring: **Contact Hours/Teaching Hours: 07**

Database Auditing and Monitoring, Database activity monitoring (DAM), Audit trails and logs, Database security policies and procedures, Database Vulnerability Assessment and Penetration Testing, Identifying and remediating database vulnerabilities, Penetration testing methodologies, Security scanning tools.

Self-Learning Hours = 6

Students shall independently install and configure open-source security scanning tools to run non-intrusive vulnerability assessments on target database instances. Students will review industry-standard database penetration testing methodologies (such as OWASP or OSSTMM) and practice interpreting database audit logs to isolate evidence of anomalous user activity or privilege escalation.

Team Activity Hours = 3

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Operating in red-team and blue-team pairs, students will simulate a database vulnerability assessment. The attacking team uses scanning tools to discover misconfigurations and weak access controls, while the defending team monitors DAM logs to track the attack footprints.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
TCS595.1	3	2		1	2	-	-	2	-	1	-	2	2	2	2
TCS595.2	2	3		2	2	2	-	3	-	2	-	2	1	2	3
TCS595.3	3	3	3	2	3	-	-	2	-	1	-	2	2	3	2
TCS595.4	3	3	2	3	3	-	-	2	-	1	-	2	3	2	3
TCS595.5	3	3	2	3	3	1	-	2	1	2	-	3	2	2	3

Text Books:

1. Principles of Information Security
- **Authors: Michael E. Whitman and Herbert J. Mattord**
- **Publisher: Cengage Learning**
- **Year of Publication: 2021 (7th Edition)**

2. Database Security
- **Authors: Alfred Basta and Melissa Zgola**
- **Publisher: Cengage Learning**
- **Year of Publication: 2011 (1st Edition)**

---

1. **Reference Books:**

1. IT Governance: An International Guide to Data Security and ISO27001/ISO27002
- **Authors:** Alan Calder and Steve Watkins
- **Publisher:** Kogan Page
- **Year of Publication:** 2019 (7th Edition)

2. The Security Risk Assessment Handbook: A Complete Guide for Performing Security Risk Assessments
- **Author:** Douglas Landoll
- **Publisher:** CRC Press
- **Year of Publication:** 2021 (3rd Edition)

Online Resources:

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

1. **Indian School of Business (ISB)**
   - URL: https://online-em.isb.edu/cybersecurity-for-leaders
2. **Tata Institute of Social Sciences (TISS)**
   - URL: https://sve.tiss.edu/TISS/index.php?p=viewProgramme&programmeId=65
3. **Massachusetts Institute of Technology (MIT)**
   - URL: https://ocw.mit.edu/courses/6-857-network-and-computer-security-spring-2014/
4. **IIT Kharagpur (NPTEL)**
   - URL: https://onlinecourses.nptel.ac.in/noc24_cs139/preview
5. **OWASP Foundation**
   - URL: https://cheatsheetseries.owasp.org/

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Course Name	:	**Computer System Security**
Course Code	:	**TCS 597**
Semester	:	**V**
Pre-requisite	:	Fundamental of Cyber Security (TCS-492).

L	T	P	Credits
3	0	0	3

Course Learning Objectives:

- To understand the core fundamentals of system security, memory vulnerabilities like buffer/integer overflows, and the mechanisms of program analysis tools to detect security vulnerabilities.
- To master the mechanics of classic software exploits (e.g., Return-to-libc, Race Conditions, Dirty COW, Shellshock) and evaluate effective countermeasures, defenses, and privilege separation models.
- To analyse common web vulnerabilities including Cross-Site Scripting (XSS), SQL Injection, and Cross-Site Request Forgery (CSRF), and implement robust session management, HTTPS, and Content Security Policies (CSP).
- To evaluate and contrast mobile operating system architectures (Android vs. iOS), studying unique threat vectors like rootkits, repackaging attacks, and advanced malware detection techniques.
- To explore hardware-level side-channel threats such as Meltdown and Spectre, assess Access Control Lists (ACLs), and investigate systemic security challenges within supply chains and critical SCADA infrastructures.

Course Outcomes:

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

Course Outcome	Statement	Bloom's Level	Skill
CO1	Explain different security threats and attacks.	**L2 – Understand**	Articulation of security risk vectors.
CO2	Know the working of different attacks and security protocols.	**L2 – Understand**	Conceptual mapping of exploit mechanics
CO3	Analyse the different security protocols.	**L3: Apply**	Critical evaluation, identifying architectural weaknesses
CO4	Use programming to implement security protocols.	**L3: Apply**	Secure coding practices
CO5	Develop system security protocols	**L4 – Analyse**	Debugging security controls

---

Details of the Course:

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Unit 1: Introduction to System security: **Contact Hours/Teaching Hours: 10**

Control hijacking attacks buffer overflow, integer overflow, bypassing browser memory protection, Sandboxing and Isolation, Tools and techniques for writing robust application software, Security vulnerability detection tools, and techniques program analysis (static, concolic and dynamic analysis), Privileges, access control, and Operating System Security, Exploitation techniques, and Fuzzing

Self-Learning Hours = 6

Students shall analyze the exact mechanics of control-flow hijacking by tracing how integer overflows bypass length checks to trigger stack-based buffer overflows.

Team Activity Hours = 3

Group designs and compiles a command-line C/C++ application containing intentional, hard-to-detect buffer overflow and integer overflow vulnerabilities.

Unit 2: Software security

Contact Hours/Teaching Hours: 09

Vulnerabilities, Attacks, and Countermeasures: Privileged programs (Set-UID programs) and vulnerabilities & Privilege Separation, Buffer Overflow vulnerability and defences, Return-to-libc attack, Race, Condition vulnerability and attack, Dirty COW attack, Format String vulnerability and attack, Shellshock attack, Heartbleed attack Interactivity, Annotation, and Arrangement.

Self-Learning Hours = 6

Students shall analyze source code variants of historical vulnerabilities (e.g., Shellshock, Heartbleed, and Dirty COW), trace memory allocation patterns during format string exploits, and practice defensive safe-string manipulation and compilation-level defenses (like ASLR and Stack Canaries).

Team Activity Hours = 3

Group code-review session where teams exchange deliberately vulnerable Set-UID C programs and demonstrate how to apply the Principle of Privilege Separation to neutralize the flaws.

Unit 3: Web Security:

Contact Hours/Teaching Hours: 10

Same origin Policy, Cross site scripting attack, Cross site request forgery attack, Sql Injection attack, Clickjacking attack, Content Security Policies (CSP) in web, Web Tracking, Session Management and User Authentication, Session Integrity, Https, SSL/TLS, Threat Modelling.

Self-Learning Hours = 6

Students shall build and deploy localized test web applications to safely test SQL Injection (SQLi), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF) exploits.

Team Activity Hours = 3

Group programming activity to identify session management vulnerabilities, and design an end-to-end secure architecture using HTTPS/TLS controls.

Unit 4: Smartphone Security:

Contact Hours/Teaching Hours: 09

Android vs. ioS security model, threat models, information tracking, rootkits, Access control in Android operating system, Rooting android devices, Repackaging attacks, Attacks on apps, Whole- disk encryption, hardware protection, Viruses, spywares, and keyloggers and malware detection.

Self-Learning Hours = 6

Students shall examine mobile application permission manifests, research real-world mobile rootkit behaviour, track information leakage mechanisms in third-party libraries, and study hardware-backed whole-disk encryption standards on mobile devices.

Team Activity Hours = 3

Group activity to contrast the safety of Google's open ecosystem access controls against Apple's strict app-vetting lifecycle regarding app repackaging attacks.

Unit 5: Hardware and system security:

Contact Hours/Teaching Hours: 08

Meltdown Attack, spectre attack, Authentication and password, Access control concept, Access control list, Capability, Sandboxing, Threats of Hardware Trojans and Supply Chain Security, Side Channel Analysis based Threats, and attacks. Issues in Critical Infrastructure and SCADA Security.

Self-Learning Hours = 6

Students shall study the operational mechanics of microarchitectural optimization flaws (Meltdown and Spectre) , research the systemic risks introduced by Hardware Trojans within tech supply chains.

Team Activity Hours = 3

Group case study analysis investigating notable historical attacks on critical SCADA infrastructure and industrial networks.

Course Articulation Matrix

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1
CO1	3	1	-	-	-	2	1	2	-	1	-	1	-
CO 2	3	2	-	-	-	1	-	1	-	-	-	1	-
CO 3	2	3	1	2	-	-	-	-	-	-	-	1	1
CO 4	2	2	3	-	3	-	-	-	1	1	-	1	3
CO 5	2	3	3	2	2	1	1	1	1	1	1	2	3

Text Books:

Authors Name	Title	Edition	Publisher, Country	Year

Department of Computer Science and Engineering

B.Tech. Computer Science and Engineering

Charles P Pfleeger and Shari Lawrence Pfleeger	Security in Computing	5th	Pearson/Addison-Wesley, American	2011
Principles and Practice, Book by William Stallings	Cryptography and Network Security	7th	Pearson/Addison-Wesley, American	1998

1. Reference Books:

Authors Name	Title	Edition	Publisher, Country	Year
W. Stallings	Network Security Essentials	6th	Prentice Hall, India	2017
Ch. P. Pfleeger, S. L. Pfleeger	*Security in Computing*	4th	Prentice Hall, India	2006

Online Resources:

1. **MIT OpenCourseWare (Massachusetts Institute of Technology)**

URL: https://ocw.mit.edu/courses/6-858-computer-systems-security-fall-2014/

2. **NPTEL Cryptography and Network Security (IIT Kharagpur)**

URL: https://onlinecourses.nptel.ac.in/noc24_cs139/preview

3. **OWASP Web Security Academy (PortSwigger)**

URL: https://portswigger.net/web-security

4. **NPTEL Introduction to Information Security (IIT Madras)**

URL: https://nptel.ac.in/courses/106106129

5. **SANS Institute (Critical Infrastructure / SCADA Security)**

URL: https://www.sans.org/industrial-control-systems-security/