An Autonomous Institute

Shree Warana Vibhag Shikshan Mandal's

Tatyasaheb Kore Institute of Engineering And Technology, Warananagar

Department of Computer Science & Engineering

Third Year Bachelor of Technology (B. Tech) in Computer Science & Engineering

w.e.f. AY- 2025-26

T.Y.B.Tech. (Sem-V) Computer Science & Engineering

Syllabus Structure and Curriculum as per NEP 2020

SWVSM's

Tatyasaheb Kore Institute of Engineering and Technology, Warananagar

An Autonomous Institute

Department of Computer Science & Engineering

Vision

To become center of excellence in the field of Computer Science and Engineering and develop competent IT technocrats

Mission

- To develop engineering graduates with high degree of processional excellence
- To excel in academics and research through contemporary and real world problems
- To enhance graduate employability through work based learning in social entrepreneurship
- To encourage industrial and nationally recognized institutes collaboration
- To create an environment to nurture lifelong learning

Program Educational Objectives (PEOs)

Graduates will be,

- Able to design and develop computing system using modern technologies by adapting business intelligence and challenges.
- Able to acquire capabilities with aptitude for higher education and entrepreneurship
- Able to function effectively as professionals having excellent interpersonal skills with ethical and social obligations.
- Able to work efficiently in multidisciplinary and multicultural environment
- Able to lead in their respective domain and contribute positively to the needs of society.

Program Specific Outcomes (PEOs)

Graduate will be able to

- Identify, design and develop solution for real world problems by implementing phases of software development process model
- Analyze and apply the computer science engineering solutions in societal and human context
- Demonstrate the skills and knowledge of contemporary issues in the field of Computer science and Engineering

Quality Policy

- To promote excellence in academic and training activities by inspiring students for becoming competent professionals to cater industrial and social needs.

SWVSM'S

Tatyasaheb Kore Institute of Engineering and Technology, Warananagar

An Autonomous Institute

Abbreviations

S N	Acronym	Definition
1	ISE	In-Semester Examination
2	ESE	End-Semester Examination
3	ISA	In-Semester Assessment (Term Work)
4	L	Lecture
5	T	Tutorial
6	P	Practical
7	CH	Contact Hours
8	C	Credits

Course Bucket Terminologies

Sr. No.	Acronym	Definition
1	PCC	Professional Core Course
2	MDM	Multidisciplinary Minor
3	OE	Open Elective Course
4	HSSM	Humanities Social Science and Management
5	ELC	Experiential Learning Course
6	VSEC	Vocational and Skill Enhancement Course
7	AEC	Ability Enhancement Course

Course/ Subject Code

F.Y.B.Tech Syllabus change year	UG/PG	Course Category with number	Separator	Branch	Semester	Course Number	
23	UG	PCC	-	CSE	5	0	1

Course Term work and POE Code

CSE	5	0	1	T / P / A
Branch	Semester	Course Number		T - Term work P - POE A - Audit Course

Third Year B. Tech.

(Semester – V)

in

Computer Science & Engineering

Curriculum Structure, Credits and Evaluation Scheme as per NEP 2020

(to be implemented from AY 2025-26)

SWVSM's

Tatyasaheb Kore Institute of Engineering and Technology, Warananagar

Third Year B. Tech. (Computer Science and Engineering)

Semester-V

(To be implemented from 2025 - 26)

Curriculum Structure, Credit and Evaluation Scheme as per NEP 2020

Sr. No.	Category	Sub Category	Course Code	Course Title	Teaching Scheme					Examination & Evaluation Scheme			
					L	T	P	C	CH	Component	Marks	Min for Passing	
1	Program Core Courses	PCC	23UGPCC-CSE501	Java Programming	3	--	--	3	3	ISE	40	16	40
										ESE	60	24	
2		PCC	23UGPCC-CSE502	Database Engineering	3	--	--	3	3	ISE	40	16	40
										ESE	60	24	
3		PCC	23UGPCC-CSE503	Computer Algorithms	3	--	--	3	3	ISE	40	16	40
										ESE	60	24	
4	Program Elective Course	PEC-1	23UGPEC1-CSE504-X	Program Elective-1	3	--	--	3	3	ISE	40	16	40
										ESE	60	24	
5	Multidisciplinary Courses	MDM-3	23UGMDM3-CSE505	Fundamentals of Machine Learning	4	--	--	4	4	ISE	40	16	40
										ESE	60	24	
6		OE-2	23UGOE2-CSE506-1	Cyber Security and Cyber Laws	3	--	--	3	3	ISE	40	16	40
										ESE	60	24	
7	Program Core Courses	PCC	23UGPCC-CSE501P	Java Programming Lab	--	--	4	2	4	ISA	25	10	
										POE	50	10	
8		PCC	23UGPCC-CSE502P	Database Engineering Lab	--	--	2	1	2	ISA	25	10	
										POE	25	10	
9		PCC	23UGPCC-CSE503T	Computer Algorithms	--	1	--	1	1	ISA	25	10	
10	Program Elective Course	PEC-1	23UGPEC1-CSE504-XP	Program Elective-I Lab	--	--	2	1	2	ISA	25	10	
										OE	25	10	
					19	**1**	**8**	**24**	**28**	**--**	**800**	**--**	

****Additional contact hours are provided for the courses without any credit**

Note: In theory examination there will be separate passing for ESE and ISE

SWVSM's

Tatyasaheb Kore Institute of Engineering and Technology, Warananagar

Third Year B. Tech. (Computer Science and Engineering)

Semester-V

(To be implemented from 2025 - 26)

Multidisciplinary Minor (MDM), Open Elective Course (OE) & Program Elective Courses Basket

Multidisciplinary Courses Course Basket Sem -V			
Multidisciplinary Minor (MDM) and Open Elective Course (OE)			
Category	**Sub Category**	**Course Code**	**Name of Course**
Multidisciplinary Courses	**MDM-3**	23UGMDM3-CSE505	Fundamentals of Machine Learning
	Open Elective – OE2	23UGOE2-CSE506-1	Cyber Security and Cyber Laws

Program Electives Courses (PEC) Basket

PEC – 1			
Category	**Sub Category**	**Course Code**	**Name of Course**
Program Elective Course	**PEC - 1**	23UGPEC1-CSE504-1	System Software and Compiler Design
		23UGPEC1-CSE504-2	Cryptography and Network Security
		23UGPEC1-CSE504-3	UNIX and Shell Scripting

23UGPCC-CSE501 – Java Programming

Click for Syllabus Structure

Teaching Scheme			Evaluation Scheme		
Lectures	**:**	2 Hrs/Week	**ISA**	**:**	25 Marks
Credits	**:**	2			
Practical	**:**	4	**POE**	**:**	25 Marks

Course Objective: The objective of this course is to	
1	To introduce the concept of object oriented programming using java.
2	To learn how to implement reliable and secure application using exception handling and package concept
3	Have the ability to write program to perform file operations.
4	To understand how to design components with java Swing API and present mechanism of multithreading
5	To understand how to familiarize database connectivity through JDBC and learn the collection framework and explore the concept of networking

Course Outcomes:		
COs	**At the end of successful completion of the course, the students will be able to**	**Bloom's Taxonomy**
CO1	Articulate the principle of object oriented problem solving & programming.	Remember
CO2	Illustrate code reusability, security and abstraction using inheritance, package and interface.	Understand
CO3	Develop reliable and user friendly applications using exception handling and file handling.	Apply
CO4	Create desktop apps using SWING and event handling and also illustrate multithreading concepts.	Apply and Understand
CO5	Use JDBC & collection framework and network programming concept.	Apply and Understand

Course Description :

This course introduces students to object-oriented programming using Java, a widely used, platform-independent programming language. The course also covers exception handling, file I/O, collections framework, GUI development using Swing, multithreading and networking. Through hands-on assignments, students will gain practical experience in building robust, reusable, and maintainable software applications in Java.

Prerequisites :	**1**	C++
	2	Operating System
	3	Computer Networks

Section – I

Unit-1	**Fundamental Programming in Java**	
	Fundamental Programming in Java: The Java Buzzwords, The Java Programming Environment- JVM, JIT Compiler, Byte Code Concept, HotSpot, A Simple Java Program, Source File Declaration Rules, Comments,	

	Data Types, Variables, Operators, Strings, Input and Output, Control Flow, Big Numbers, Arrays-Jagged Array. **Objects and Classes:** Object-Oriented Programming Concepts, Declaring Classes, Declaring Member Variables, Defining Methods, Constructor, Passing Information to a Method or a Constructor, Creating and using objects, Controlling Access to Class Members, Static Fields and Methods, this keyword, Object Cloning, Class Design Hints.	**07 Hours**
Unit-2	**Inheritance and Interface** **Inheritance:** Definition, Super classes, and Subclasses, Overriding and Hiding Methods, Polymorphism, Inheritance Hierarchies, Super keyword, Final Classes and Methods, Abstract Classes and Methods, casting, Design Hints for Inheritance, Nested classes & Inner Classes, finalization and garbage collection. **Interfaces:** Defining an Interface, Implementing an Interface, Using an Interface as a Type, Evolving Interfaces, and Default Methods.	**06 Hours**
Unit-3	**Package and Exception** **Packages:** Class importing, Creating a Package, Naming a Package, Using Package Members, Managing Source and Class Files. Developing and deploying (executable) Jar File. **Exception:** Definition, Dealing with Errors, The Classification of Exceptions, Declaring Checked Exceptions, Throw an Exception, Creating Exception Classes, Catching Exceptions, Catching Multiple Exceptions, Re-throwing and Chaining Exceptions, finally clause, Advantages of Exceptions, Tips for Using Exceptions.	**07 Hours**
Section – II		
Unit-4	**I/O Streams and Multithreading** **I/O Streams:** Byte Stream – InputStream, OutputStream, FileInputStream, FileOutputStream, Character Streams- FileReader, FileWriter, BufferedStream, Scanner, File, RandomAccesFile. **Multithreading:** Processes and Threads, Runnable Interface and Thread Class , Thread Objects, Defining and Starting a Thread, Pausing Execution with Sleep, Interrupts, Thread States, Thread Properties, Joins, Synchronization, Deadlock	**06 Hours**
Unit-5	**GUI using Swing and Collections** **Graphical User Interfaces using Swing:** Introduction to the Swing, Swing features, Swing Top Level Containers, Creating a Frame, Positioning a Frame, Displaying Information in a Panel, The Model-View-Controller Design Pattern, The JComponent Class. **Layout Management:** Introduction to Layout Management, APIs for Border Layout, Flow Layout, Grid Layout **Event Handling:** Basics of Event Handling, The AWT Event Hierarchy, Semantic and Low- Level Events in the AWT, Low-Level Event Types **User Interface Components:** Text Input, Choice Components, Menus, Dialog Boxes Setting the Look and Feel of UI, Introduction to JApplet	**07 Hours**

	Collections: Collection Interfaces, Concrete Collections- List, Queue, Set, Map, the Collections Framework, Lambda Expressions and Annotations	
Unit-6	**JDBC and Networking** **JDBC:** The Design of JDBC, The Structured Query Language, JDBC Installation, Basic JDBC Programming Concepts, Query Execution, Statement, Prepared Statement, Scrollable and Updatable Result Sets **Networking:** Overview of Networking, Networking Basics, Working with URLs, Creating a URL, Parsing a URL, Reading Directly from a URL, Connecting to a URL, Reading from and Writing to a URL Connection, Sockets, Reading from and Writing to a Socket, Writing the Server Side of a Socket, Datagrams, Writing a Datagram Client and Server.	**06 Hours**

Proposed List of Experiments

Expt. No.	Experiment Title	Bloom's Taxonomy
1	**Use of instance variables, constructors, getter, setters** Create a class called Employee that includes three pieces of information as instance variables- first name, a last name and a monthly salary. Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again.	Understand
2	**Use of static members of the class.** Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by 12this interest should be added to savingsBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs 2000.00 and Rs 3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.	Apply
3	**Implementation of abstract class** Create abstract class Shape which has instance variables side, area and perimeter And methods calculateArea(), calculatePerimeter() as abstract methods and display() as concrete method. Write subclasses which extend Shape class like Triangle, Rectangle, Circle , Cube and Squere and override abstract methods and display methods in subclass take instance variable if needed as per the formula. And use parameterized constructor to initialize instance variables using "this" reference variable Write Test class and Create a reference variable of Shape which will hold the objects of all the sub classes and calculate respective area, perimeter and display the results.	Apply
4	**Implementation of Interface** Create the interface stack which has variable size, abstract methods push(),pop(),display(), overflow() and underflow(). We need to implement 3 subclasses IntegerStack, StringStack and DoubleStack respectively by implementing interface. All the methods in interface are declared for string. And in subclass for integerStack convert string to integer. Same thing to all other. Create one test class and check for the working of all the	Understand

	classes.	
5	**Implementation of package and creation of jar file.** Develop a mathematical package for Statistical operations like Mean, Median, Average, Standard deviation. Create a sub package in the math package -convert. In "convert" package provide classes to convert decimal to octal, binary, hex and vice-versa. Develop application program to use this package, and build executable jar file of it.	Apply
6	**Implementation of Exception handling. (Pre Defined Exception)** Develop application which can handle any 5 combination of predefined compile time and runtime exceptions using multiple catch blocks. Use throws and finally keywords as well.	Analayse
7	**Implementation of Exception handling. (User defined Exception)** Develop a BankAccount class which should contain all methods of Bank i.ebalanceEnquery(), withdraw(), transfer() and deposit(). You should create at least two objects of BankAccount using array and do all operations mentioned above. Also generate user defined exception LowBalanceException, NegetiveNumberException and PasswordMismatchException whenever required. To transfer amount from one account to another use two BankAccount objects.	Develop
8	**Simple file handling program in java.** Take file name as input to your program through command line, If file is existing the open and display contents of the file. After displaying contents of file ask user – 1.do you want to add the data at the end of file or 2.replace specified text in file by other text. Based on user's response, then accept data from user and append it to file. If file in not existing then create a fresh new file and store user data into it. Also. User should type exit on new line to stop the program. Do this program using Character stream classes.	Develop
9	**Implementation of reading student data and store it into file. (Serialization)** Take Student information such as name, age, weight, height, city, phone from user and store it in the file using DataOutputStream and FileOutputStream and Retrive data using DataInputStream and FileInputStream and display the result. Use Serialization concept and Bytestream classes.	Develop
10	**Implementation of GUI programming (Design of Calc).** Develop a Swing GUI based standard calculator program. Use event handling, Layout of swing package.	Design
11	**Implementation of GUI programming and multithreading. (Design of Stopwatch).** Create Stop Watch with Swing GUI and Multithreading. Provide Facility for Lap Counting.	Develop
12	**Implementation of GUI using Networking** Create GUI Based chat application using TCP or UDP.	Design
13	**Implementation of Collection (ArrayList,LinkedList).** Write a program to read a text file one line at a time. Read each line as a	Apply

	String and place that String object into a LinkedList. Print all of the lines in the LinkedList in reverse order.	
14	**Implementation of Collections (Hashmap,LinkedHashMap).** Fill a HashMap with key-value pairs. Print the results to show ordering by hash code. Extract the pairs, sort by key, and place the result into a LinkedHashMap. Show that the insertion order is maintained.	Apply
15	**Implementation of Student registration and login using GUI.** Write a GUI based program to create a student registration and Login. Store Registration data in Database and take Login information from Database.	Design

Mapping of POs & COs:

PO CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	If Applicable		
													PSO1	PSO2	PSO3
CO1	-	2	-	3	-	-	-	1	-	1	-	-	1	-	1
CO2	-	2	2	-	2	-	-	1	-	-	-	-	1	-	1
CO3	-	-	-	2	2	1	1	1	-	1	-	1	1	1	1
CO4	2	2	3	-	2	-	-	1	-	-	1	1	1	-	1
CO5	-	2		1	-	-	2	-	-	1	-	-	1	1	1

References	
Text Books :	
1	Core Java- Volume I Fundamentals - Cay Horseman and Gary Cornell, Pearson, Eight edition
Reference Books :	
1	JAVA-The Complete Reference - Herbert Schildt - Mcgraw Hill, Oracle Press Tenth edition.
2	Head First Java - Eric Freeman Elisabeth Robson Bert Bates Kathy Sierra - O'Reilly Publication - 3 rd edition
SWAYAM Courses (Operational Timestamp: Saturday,10-May-2025 on 3:45PM)	
1	SWAYAM Courses (Programming in Java: https://onlinecourses.nptel.ac.in/noc22_cs47/preview [IIT, Kharagpur]

23UGPCC CSE501P – System Software and Compiler Design Lab

Teaching Scheme			Evaluation Scheme		
Practical	**:**	2 Hrs/Week	**ISA**	**:**	25 Marks
Credits	**:**	1	**POE**	**:**	25 Marks

Expt. No.	Title of Experiment	Bloom's Taxonomy
1	Implement some macros in any assembly language and show the interpretation of the macro assembler, Linker and loader	Apply
2	Explore the basic structure and functionality of Lex and YACC tools.	Understand
3	Write a Lex program to identify and count vowels and consonants in a given input string.	Apply
4	Write a Lex program to identify letters and words from a given input string.	Apply
5	Write a Lex program to recognize letters, digits, words, operators, and special symbols.	Apply
6	Write a Lex program to identify positive/negative integers and positive/negative fractions from input.	Apply
7	Write a Lex program to identify signed integers and fractions using file handling.	Apply
8	Write a Lex program to tokenize input from a file.	Apply
9	Write a Lex program to convert `printf` and `scanf` statements to `write` and `read` using file handling.	Apply
10	Write a YACC program to convert infix expressions to postfix notation.	Apply
11	Write a YACC program to implement a calculator that evaluates valid arithmetic expressions.	Analyze
12	Write a YACC program to check whether a given input string is a palindrome.	Analyze

23UGPEC1-CSE504-2P - Cryptography and Network Security Lab

Teaching Scheme

Practical : 2 Hrs/Week

Credits : 1

Evaluation Scheme

ISA : 25 Marks

Expt. No.	Experiment Title	Bloom's Taxonomy
1	Perform encryption, decryption using the Ceaser Cipher substitution technique.	Apply
2	Perform encryption, decryption using the Hill Cipher substitution technique	Apply
3	Perform encryption, decryption using the Rail Fence transposition technique	Apply
4	Perform encryption, decryption using the Row and Column transposition technique	Apply
5	Implement DES (Data Encryption Standard) Algorithm	Apply
6	Implement RSA Algorithm	Apply
7	Implement the Diffie-Hellman Key Exchange Algorithm	Apply
8	Demonstrate intrusion detection system (ids) using any tool eg. Snort or any other software.	Analyze
9	Defeating Malware i) Building Trojans ii) Rootkit Hunter	Analyze
10	Configuring S/MIME for e-mail communication	Analyze
11	Port Scanning and Monitoring Using Nmap.	Analyze
12	Perform surveillance through packet sniffer tool like Wireshark &TCP Dump	Analyze

23UGPEC1-CSE504-3P-UNIX and Shell Programming Laboratory

Click for Syllabus Structure

Teaching Scheme
Practical : 2 Hrs/Week
Credits : 1

Evaluation Scheme
ISA : 25 Marks

Expt. No.	Experiment	Bloom's Taxonomy
1	Installing, partitioning, and configuring GNU/Linux OS distribution	Understand & Apply
2	Essential Linux/Unix commands (a) General Purpose Utilities (b) File System (file handling) Commands (c) Directory related commands (d) Process control commands (e) Text Processing commands (f) Filtering using Regular Expression (g) Networking Commands	Apply
3	Implementation of various operations on Files (creat, open, read, write, append, fstat, dup)	Apply
4	Implementation of various system call (OPEN,READ,WRITE)	Apply
5	Study and implementation of pipe() system call (Reading and Writing through Named and Unnamed Pipe)	Apply
6	Signal Handling and writing user defined function as signal handler	Apply
7	Use of fork() and exec()	Apply
8	Use of important filters for grep ,seed and features of Unix regular expression	Apply
9	Shell Scripting Programming Scenario 1	Apply
10	Shell Scripting Programming Scenario 2	Apply

23UGPCCSE502 Database Engineering

Teaching Scheme

Click Syllabus Structure

Lectures : 3 Hrs/Week
Credits : 2
Practical : --

Evaluation Scheme
ISE : 40 Marks
ESE : 60 Marks

Course Objective : The objective of this course is	
1	Understand the Fundamental Concepts of Database Systems
2	Develop Proficiency in Database Design Techniques
3	Gain Practical Skills in SQL Programming
4	Explore Data Storage, Indexing, and File Organization Techniques
5	Understand Transaction Management and Recovery Mechanisms

Course Outcomes :

COs	**At the end of successful completion of the course, the students will be able to**	**Bloom's Taxonomy**
CO1	Describe the fundamentals of Database and Relational Database Management Systems	Understand
CO2	Identify appropriate Entities and their Relationship, also Convert the same to Relational Database for a given problem	Understand
CO3	Identify the use of functional dependency and normalization in database design.	Apply
CO4	Write SQL queries to design & manage the database	Analyze
CO5	Illustrate Transactions, Concurrency and Recovery apply to database system	Analyze

Course Description :

This Course is designed to understand the internals of Database System, with elaboration from Database Design, Using Relational Database (using SQL) and the transaction concepts

Prerequisites :	1	Set Theory
	2	Operating System
	3	Data Structures

Section – I

Unit-1	**Introduction to Databases**	
	Database System Applications, Purpose of Database Systems, View of Data, Database Languages, Specialty Databases, Database Users & Administrators, Structure of Relational Databases, Database Schema, Keys, Relational Query Languages, Relational Operations	**6 Hrs**
Unit-2	**Database Design**	
	E-R Model: The Entity-Relationship Model, Constraints, Entity-Relationship Diagrams, Reduction to Relational Schemas **Normalization:** Data Redundancies & Update Anomalies, Functional Dependencies, The Process of Normalization, First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form	**7 Hrs**

Unit-3	**Structured Query Language (SQL)**	
	Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Aggregate Functions, and Nested sub Queries, Modification of Databases, Joins and Views and DCL	**6 Hrs**
Section – II		
Unit-4	**Data Storage & Indexing**	
	File Organization, Organization of records in File, Data Dictionary Storage, Basic Concepts indexing & hashing, Ordered Indices, B+ Tree Index files, Static Hashing	**7 Hrs**
Unit-5	**Transaction Management**	
	Transaction Concept, A Simple Transaction Model, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Lock-Based Protocols, Deadlock Handling, Timestamp-Based Protocols, Validation-Based Protocol	**7 Hrs**
Unit-6	**Recovery System**	
	Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm, Failure with Loss of Non-volatile Storage, Remote Backup Systems	**6 Hrs**

Mapping of POs & COs:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PSO 1	PSO 2	PSO 3
CO1	2	--	2	--	--	--	--	--	--	--	--	1	2	1	2
CO2	2	3	2	1	1	--	--	--	--	--	--	--	3	1	2
CO3	2	2	3	3	--	--	--	--	--	--	--	1	3	--	2
CO4	1	--	1	--	2	--	--	--	--	--	--	--	3	--	3
CO5	2	1	-	1	-	--	--	--	--	--	--	1	2	1	2

References	
Text Books :	
1	Database System Concepts, A. Silberschatz, H.F. Korth, S. Sudarshan, 6th Edition, Mc Graw Hill Education [for Unit No. I, 2.1, III,IV,V,VI]
2	Database Systems - A practical approach to Design, Implementation and Management Thomas Connolly, Carolyn Begg, 3rd Edition, Pearson Education [for Unit No. II, 2.2]
Reference Books :	
1	Database Systems – Design, Implementation and Management, Rob & Coronel 5th Edition, Thomson Course Technology
2	Fundamentals of Database Systems, Ramez Elmasri, Shamkant B. Navathe, 4th Edition, Pearson Education

23UGPCC-CSE503 - Computer Algorithms

Teaching Scheme
Lectures : 3 hrs/week
Credits : 3
Tutorials : 1

Evaluation Scheme:
ISE : 40 Marks
ESE : 60 Marks

Course Objectives: The objective of this course is to

1. To introduce basic concepts and importance of algorithms.
2. To explain design techniques like Divide and Conquer, Greedy, Dynamic Programming.
3. To understand algorithms for sorting, searching, and graph traversal.
4. To explore backtracking and its use in problem solving.
5. To give an overview of NP problems and computational limits.

Course Outcomes:

COs	At the end of successful completion of the course the student will be able to	Blooms Taxonomy
CO1	Understand basics of algorithms and performance analysis.	Understand
CO2	Solve problems using divide & conquer, greedy, and dynamic programming.	Apply
CO3	Implement and analyze sorting, searching, and graph algorithms.	Analyze
CO4	Solve optimization problems using backtracking.	Analyze
CO5	Classify problems by complexity: P, NP, NP-Complete, NP-Hard.	Evaluate

Description

This Course is designed to understand the various key aspects and algorithm design techniques to solve the real-world problems.

Prerequisites:	1:	Data Structures
	2:	Discrete Mathematics
	3:	Engineering Mathematics
	4:	Programming Concepts

Section - I		
Unit 1	**Divide and Conquer**	
	Algorithm-Algorithm Specification, Performance Analysis, Randomized Algorithms. **Divide and Conquer**: The General Method, Binary Search, Finding the Maximum and Minimum, Merge Sort, Quick Sort, Selection.	**9 Hrs**
Unit 2	**The Greedy Method**	
	The General Method, Knapsack Problem, Job sequencing with deadlines, Minimum-Cost Spanning Trees – Prim's and Kruskal's Algorithms, Optimal Storage on tapes, Optimal Merge Patterns- Huffman code, Single Source Shortest Paths.	**8 Hrs**

Unit 3	**Dynamic Programming**	
	The General Method, Multistage Graphs, All Pair Shortest Paths, Reliability design, Traveling Sales Person Problem.	**8 Hrs**
Section – II		
Unit 4	**Basic Traversal and Search Techniques**	
	Techniques for Binary Trees, Techniques for Graphs – Breadth First Search & Traversal, Depth First Search & Traversal and Spanning Trees, connected components and spanning tree, Biconnected Components and DFS	**6 Hrs**
Unit 5	**Backtracking**	
	The General Method, Sum of Subsets, n-queen problem, Hamiltonian Cycle, Graph Coloring	**5 Hrs**
Unit 6	**NP Hard and NP Complete**	
	Basic Concepts, P, NP, NP-Complete, NP-Hard Problems, Introduction to NP Hard Graph Problems.	**3 Hrs**

Mapping of POs & COs:

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

References:

Text Books	
1	Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni, Sanguthevar Rajasekaran, 2nd Edition,University Press.
2	Introduction to Algorithms, Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein,3rd Edition,PHI.

Reference Books	
1	Introduction to The Design and Analysis of Algorithms", Anany Levitin, 3rd Edition,Pearson.`
2	Fundamentals of Algorithms Gilles Brassard, Paul Bratley Pearson Education

SWAYAM Courses:	
1	https://onlinecourses.nptel.ac.in/noc22_cs71/preview [CMI]
2	https://onlinecourses.swayam2.ac.in/cec22_cs13/preview [UoK]
Additional Web-link:	
1	https://docs.oracle.com/cd/E19059-01/stud.10/819-0493/OtherTools.html
2	http://www.uniquecareer.in/computer-algorithm/ [TKIET]

Term Work & Tutorial List	
1	**Algorithm Basics and Performance:** • Explain the importance of algorithms and how to analyze their time complexity. • Solve: Analyze the time complexity of binary search and linear search algorithms.
2	**Divide and Conquer:** • Describe the divide and conquer approach with examples like finding maximum/minimum in an array. • Solve: Perform Merge Sort on the array [38, 27, 43, 3, 9, 82, 10].
3	**Binary Search Algorithm:** • Explain binary search with its recursive and iterative forms. • Solve: Trace the steps of binary search for finding 43 in the sorted array [10, 20, 30, 40, 43, 50].
4	**Greedy Method and Knapsack Problem:** • Explain the greedy method using the fractional knapsack problem. • Solve: Given weights and profits, use the greedy approach to maximize profit in fractional knapsack.
5	**Minimum Spanning Tree Algorithms:** • Discuss Prim's and Kruskal's algorithms and their applications. • Solve: Find MST for a given graph with vertices and edges (provide a simple weighted graph).
6	**Dynamic Programming Introduction:** • Explain the concept of overlapping subproblems and optimal substructure. • Solve: Use dynamic programming to find the shortest path in a multistage graph (simple example).
7	**All Pair Shortest Path (Floyd Warshall):** • Explain Floyd Warshall algorithm and its time complexity. • Solve: Apply Floyd Warshall to the given adjacency matrix and find shortest paths.
8	**Tree and Graph Traversal:** • Explain BFS and DFS with examples in graphs and binary trees. • Solve: Perform BFS and DFS traversal on the provided graph and write the traversal order.
9	**Backtracking Method:** • Explain backtracking with the N-Queens problem and sum of subsets. • Solve: Write steps to place 4 queens on a 4x4 chessboard without conflict.
10	**Hamiltonian Cycle Problem:** • Define Hamiltonian Cycle and explain why it is NP-Complete. • Solve: Try to find a Hamiltonian cycle in a small given graph (example with 5 vertices).
11	**NP Problems Overview:** • Differentiate between P, NP, NP-Complete, and NP-Hard problems with examples. • Write a short note on why solving NP-Complete problems efficiently is challenging.

23UGPCC CSE501 – System Software and Compiler Design

Click for Syllabus Structure

Teaching Scheme			Evaluation Scheme		
Lectures	**:**	2 Hrs/Week	**ISE**	**:**	40 Marks
Credits	**:**	2			
Tutorials	**:**	--	**ESE**	**:**	60 Marks

Course Objective: The objective of this course is to	
1	Introduce the fundamental concepts of language processors, including assemblers, linkers, loaders, and macro processors.
2	Explain the structure and working of compilers and their constituent phases.
3	Design and implement the analysis phases of a compiler such as lexical and syntax analysis.
4	Develop and implement the synthesis phases of a compiler, including intermediate code generation and code optimization.

Course Outcomes:		
COs	**At the end of successful completion of the course, the students will be able to**	**Bloom's Taxonomy**
CO1	Explain the translation process of assembly language and describe the functions of assemblers.	Understand
CO2	Employ macro processing techniques and demonstrate understanding of linking and loading mechanisms.	Apply
CO3	Interpret the structure and functioning of a compiler, emphasizing lexical and syntax analysis.	Understand
CO4	Construct top-down and bottom-up parsers for context-free grammars using suitable parsing strategies.	Apply
CO5	Formulate intermediate code using syntax-directed translation techniques and generate basic optimizations with three-address instructions.	Apply

Course Description :

This course introduces system software components such as assemblers, loaders, and compilers, with a focus on compiler phases including lexical analysis, syntax analysis, intermediate code generation, and optimization.

Prerequisites :		
	1	Operating Systems
	2	Finite Automata and Formal Languages
	3	Microprocessor Architecture

Section – I

Unit-1	**Language Processors and Assemblers** Introduction to Language Processors, Elements of Assembly Language Programming, Advanced Assembler Directives, Pass-I of the Assembler, Intermediate Code Forms, Pass-II of the Assembler	**06 Hours**
Unit-2	**Macros, Linkers and Loaders** Macro Definition and Calls, Macro Expansion and Nested Macro Calls,	**07 Hours**

	Advanced Macro Facilities, Relocation and Linking Concepts, Self-Relocating Programs, Loaders	
Unit-3	**Lexical Analysis** Structure of a Compiler, Role of the Lexical Analyzer, Lexical Errors, Lexemes, Tokens, and Patterns, Input Buffering and Transition Diagrams, Lexical Analyzer Generator: LEX	**07 Hours**
Section – II		
Unit-4	**Syntax Analysis** Introduction, Context Free Grammar (CFG), Top-Down Parsing: Recursive Descent Parsing Technique and Predictive Parsing Technique: LL (1), Bottom up Parsing: LR(0), SLR (1).	**07 Hours**
Unit-5	**Syntax Directed Translation and Intermediate code generation** Syntax-Directed Translation (SDD and SDT), Synthesized and Inherited Attributes, S-Attributed and L-Attributed Definitions, Intermediate Code Generation: Triples, Quadruples, and Three-Address Instructions, TAC for Boolean Expressions and Assignment Statements	**07 Hours**
Unit-6	**Code Optimization and Code Generation** Introduction to Code Optimization, Principal Sources of Optimization, Peephole Optimization, Issues in Code Generation	**06 Hours**

Mapping of POs & COs:

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

References	
Text Books :	
1	System Programming and Operating System, D. M. Dhamadhere, 2nd Edition-McGraw Hill Education India.(Unit I & II)
2	Compilers: Principles, Techniques and Tools, Jeffery D. Ullman, Alfred V Aho, Ravi Sethi-Pearson Education India.(Unit III, IV, V & VI)

Reference Books :	
1	Compiler Construction, D.M. Dhamdare, Mc-Millan
SWAYAM Courses (Operational Timestamp: Saturday,10-May-2025 on 3:45PM)	
1	SWAYAM Courses (Operational Timestamp: https://onlinecourses.nptel.ac.in/noc25_cs13/preview [IIT, Kharagpur]

.

23UGPEC1-CSE504-2: Cryptography and Network Security

Teaching Scheme
Lectures : 2 Hrs/Week
Credits : 2
Tutorials : --

Evaluation Scheme
ISE : 40 Marks
ESE : 60 Marks

Course Objective: The objective of this course is	
1	Explain different types of symmetric and asymmetric security techniques.
2	Compare different types of cryptographic algorithms to ensure data integrity.
3	Explain different types of security protocols in TCP/IP protocol suite.
4	Understanding different types of security threats for computer system.

Course Outcomes:		
COs	**At the end of successful completion of the course, the students will be able to**	**Bloom's Taxonomy**
CO1	*Explain* classical encryption techniques and the basic model of network security.	Understand
CO2	*Use* symmetric and asymmetric key cryptographic algorithms including DES, RSA, and Diffie-Hellman	Apply
CO3	*Examine* cryptographic authentication mechanisms such as hash functions, MACs, and digital signature schemes	Analyze
CO4	*Implement* key management and user authentication protocols like Kerberos and X.509	Apply
CO5	Use internet and system security tools to detect and prevent threats	Apply

Course Description:

This course is designed to provide a fundamental understanding of Information and Network Security. It covers core concepts such as cryptographic techniques, authentication mechanisms, key management, secure communication protocols, and system security to protect data and network infrastructure against various threats.

Prerequisites:	1	Basic knowledge of Communication system

Section – I

Unit-1	**Classical Encryption Techniques**	
	Overview – The OSI Security Architecture, Security Attacks, Services and Mechanism, A Model for Network Security, Symmetric Cipher Model Classical Encryption Techniques – Substitution Techniques, Transposition Techniques.	**5 Hours**
Unit-2	**Symmetric and Asymmetric Key Cryptography**	
	Block Ciphers and the Data Encryption Standard: Block Cipher Structure, Data Encryption Standard (DES), A DES Example, Strength of DES, Block Cipher Design Principles. **Public Key Cryptography:** Principles of Public-Key Cryptosystems, RSA Algorithm, Other Public key, Cryptosystems - Diffie-Hellman Key Exchange.	**7 Hours**

Unit-3	**Cryptographic Authentication Functions**	
	Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Two Simple Hash Functions, Hash Functions Based on Cipher Block Chaining. **Message Authentication Codes:** Message Authentication Requirements, Message Authentication Functions, Requirements for MAC and Security of MACs, MACs Based on Hash Functions. **Digital Signatures:** Digital Signatures, ElGamal Digital Signature Scheme, Schnorr Digital Signature Scheme, Digital Signature Standard (DSS)	**8 Hours**
Section – II		
Unit-4	**Key Management and User Authentication**	
	Key management: Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key ,Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates, Public Key Infrastructure **User Authentication Protocol:** Remote User-Authentication Principles, Remote User-Authentication Using Symmetric Encryption, Kerberos, Remote User Authentication Using Asymmetric Encryption	**7 Hours**
Unit-5	**Internet Security Protocols**	
	Transport-Level Security: Web Security Issues, Secure Sockets Layer (SSL), Transport Layer Security (TLS), SSH, **Electronic Mail Security:** Pretty Good Privacy (PGP). **IP Security:** IP Security Overview, IP Security Policy, Encapsulating Security Payload. **Case Study:** Perform surveillance through packet sniffer tool like Wireshark &TCP Dump.	**6 Hours**
Unit-6	**System Security**	
	Intruders: Intrusion Detection, Password Management, **Malicious Software:** Viruses and Related Threat, Countermeasures. **Firewalls:** Firewall Design Principles, Trusted Systems.	**6 Hours**

CO-PO-PSO Mapping Table:

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

References	
Text Books :	
1	Williams Stallings – Cryptography and Network Security Principles and Practices Pearson Education (LPE), 6th Edition (For Unit I to V)
2	Williams Stallings – Cryptography and Network Security Principles and Practices Pearson Education (LPE), 4th Edition(For Unit VI)
Reference Books :	
1	Cryptography & Network Security B.A. Forouzan McGrawHill
2	Cryptography and network security – AtulKahate (TMGH)
3	Handbook of Applied Cryptography - Menezes, an Oorschot, and S.A. Vanstone
SWAYAM Courses (Operational Timestamp: Sat,16-Jul-2022 on 7:00 AM)	
1	https://onlinecourses.swayam2.ac.in/cec22_cs15/preview

23UGPEC1-CSE504-3 – (PEC-I) UNIX and Shell Programming

Click for Syllabus Structure

Teaching Scheme			Evaluation Scheme		
Lectures	**:**	2 Hrs/Week	**ISE**	**:**	40 Marks
Credits	**:**	2			
Tutorials	**:**	--	**ESE**	**:**	60 Marks

Course Objective: The objective of this course is to	
1	To provide knowledge to the students about Fundamental architecture of UNIX/Linux operating system fundamentals
2	To understand File subsystem and related functions (system calls)
3	To understand Process Control subsystem and related functions (system calls) and memory allocation
4	To understand the Unix/Linux process and Inter Process Communication using pipes, signals
5	To provide a comprehensive introduction to Shell programming

Course Outcomes:		
COs	**At the end of successful completion of the course, the students will be able to**	**Bloom's Taxonomy**
CO1	To understand Fundamental architecture of UNIX and its components	Understanding
CO2	To Demonstrate various UNIX system calls(functions) for file subsystem handling and its data structures	Apply
CO3	To Demonstrate various UNIX system calls(functions) for Process control subsystem handling and IPC	Apply
CO4	Demonstrate UNIX signals, important filters and features of UNIX regular expression.	Apply
CO5	Learn UNIX shell programming constructs and write the shell scripts for given scenario	Apply

Course Description:

This course provides understanding of the Unix, GNU Linux operating system and describes the application programming interface of the UNIX family of operating systems programming environment

Prerequisites:		
	1	Data Structures and Algorithms
	2	Operating System Course (CSE404)
	3	Concepts of Operating System Process

Section – I

Unit-1	**UNIX System Overview** Introduction, UNIX Architecture, User Perspective : File System, Processing Environment, Building Block Primitives. Architecture of Unix OS, Introduction to System Concepts : Overview of File Subsystems, Processes. Logging, Files and Directories.	**06 Hours**

Unit-2	**File I/O, Files and Directories**	**07 Hours**
	File I/O : Introduction File Descriptors, ***open()***, data structures after ***open()***, data structure after two processes open files, ***read()***, ***write()***, ***creat()***, ***close()***, data structure after a processes close the file. ***lseek()***, File Sharing, ***dup()***, ***dup2()***. Files and Directories : Introduction ***stat()***, ***fstat()***, File Types, File Access Permissions, File Systems, Symbolic Links, Hard Links	
Unit-3	**Process Environment, Control**	**06 Hours**
	Process Environment : Introduction, ***main()*** function, Process Termination, Command-Line arguments, Environment List, Memory Layout of 'C' Program, Shared Libraries, Memory Allocation. Process Control: Process Identifiers, ***fork()***, ***vfork()***, ***exit()***, ***wait()***, ***exec(), system()***	
Section – II		
Unit-4	**Process Relationships and IPC**	**06 Hours**
	Process Relationships : Terminal Logins, Linux Terminal Logins, Network Logins, Linux Network Login, Job Control. IPC : Pipes, FIFOs	
Unit-5	**Signals and Filters using Regular Expression**	**06 Hours**
	Signals : Signal Concepts, Signal Function, ***kill()***, ***raise()***, ***alarm()***, ***pause()***. Filters Using Regular Expression – ***grep***: Searching for a Pattern, Basic Regular Expression(BRE), Extended Regular Expression (ERE) and ***egrep***, ***sed*** : The Stream Editor, Line Addressing, Using Multiple Instructions, Context Addressing, Writing Selected Lines to a File, Text Editing, Substitution, Basic Regular Expressions Revisited	
Unit-6	**Essential Shell Programming**	**07 Hours**
	Shell Scripts, ***read*** : Making Scripts Interactive, Using Command-Line Arguments, ***exit*** and Exit Status of Command, The Logical Operators && and ||,The if conditional, Using ***test*** and [] to Evaluate Expressions, The case Conditional, ***expr*** : Computation and String Handling, $0 : Calling a Script by Different Names, ***while*** : Looping, ***for*** : Looping with a List, ***set*** and ***shift***, The Here Document, ***trap*** : Interrupting a Program	

Mapping of POs & COs:

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

References	
Text Books :	
1	Advanced Programming in the UNIX Environment, W.Richard Stevens, 2nd Edition, Pearson
2	The Design of UNIX Operating System, Maurice Bach, PHI
3	Unix Concepts and Applications, Sumitabha Das, 3rd Edition, Tata McGraw Hill
Reference Books :	
1	Understanding the LINUX Kernel, Daniel P. Bovet and Marco Cesati, 3rd Edition, O'Reilly
2	LINUX In a Nutshell, Ellen Siever, Stephen Figgins, Robert Love, Sixth Edition, O'Reilly
3	Linux: The Complete Reference, Sixth Edition, Richard Petersen, McGraw-Hill
SWAYAM Courses (Operational Timestamp: Saturday, 27-May-2025 on 3:45PM)	
1	https://onlinecourses.nptel.ac.in/noc22_cs78/preview [IIT, Madras]
2	https://onlinecourses.swayam2.ac.in/cec22_cs23/preview [University of Madras]

23UGMDM3-CSE505 – Fundamentals of Machine Learning

Lectures : 4 Hrs/week
Credits : 4
Tutorial : --

Evaluation Scheme:
ISE : 40 Marks
ESE : 60 Marks

Course Objectives: The objective of this course is to

1. Introduce basic concepts and types of machine learning.
2. Explain data handling and visualization techniques.
3. Help students understand simple ML models.
4. Introduce classification and clustering with real-world examples.
5. ML applications, trends, and use across science and engineering fields.

Course Outcomes:

COs	At the end of successful completion of the course the student will be able to	Blooms Taxonomy
CO1	Explain basic concepts and terms in machine learning.	Understand
CO2	Classify different learning types with relevant examples.	Understand
CO3	Apply suitable ML techniques to basic real-world problems.	Apply
CO4	Interpret outcomes of simple models.	Analyze
CO5	Describe ML applications in science and engineering fields.	Understand

Description

This course introduces the basic concepts of Machine Learning in a simplified and practical way for students from non-computer science backgrounds. It focuses on real-world applications, intuitive understanding of learning models like regression and classification, and how ML can help solve problems in various engineering fields without requiring programming knowledge.

Prerequisites:	1:	Basic understanding of mathematical and statistical concepts.
	2:	Basics of programming languages.

Unit 1	**Introduction to Machine Learning**	
	Basics of Machine Learning, Types of Learning: Supervised, Unsupervised, Real-life examples of ML (e.g., spam filter, movie recommendations), Steps in ML process.	**5Hrs**
Unit 2	**Working with Data**	
	Basics of data collection, types of data (numeric, categorical), importance of data cleaning. Introduction to data visualization using line charts, bar charts, pie charts, histograms, scatter plots, and box plots.	**7Hrs**

Unit 3	**Understanding Predictions – Regression**	
	Concept of prediction with real-life examples, Introduction to simple linear regression and multiple regression, understanding the line of best fit and visual interpretation. Basics of evaluating model accuracy using simple error metrics (e.g., Mean Absolute Error).	**6Hrs**
Unit 4	**Understanding Categories – Classification**	
	Concept of classification and linear classifiers, introduction to logistic regression, real-life applications of classification, introduction to decision trees, relevant terminologies and decision boundaries.	**6Hrs**
Unit 5	**Introduction to Clustering**	
	Differences between supervised and unsupervised learning; grouping data based on similarity without predefined labels; introduction to the concept of cluster centroids and K-Means clustering.	**5Hrs**
Unit 6	**Applications and Trends in ML**	
	Overview of recommendation systems in OTT platforms, streaming, e-commerce, and social networking; introduction to image, text and voice analysis (natural language processing); discussion on responsible ML use, ethics, limitations, and future career opportunities.	**7Hrs**

Mapping of POs & COs:

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

References:

Text Books	
1	Tom M. Mitchell, "*Machine Learning*", McGraw-Hill, 1997.

Reference Books

1	Aurélien Géron, *"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow"*, O'Reilly Media, 2019.
2	Stuart Russell and Peter Norvig, *"Artificial Intelligence: A Modern Approach"*, Pearson, 4th Edition, 2020.

SWAYAM Courses:	
1	https://onlinecourses.nptel.ac.in/noc23_cs18/preview [IIT Madras]
2	https://onlinecourses.swayam2.ac.in/imb24_mg126/preview [ATLAS SkillTech University]
Additional Web-link:	
1	https://www.coursera.org/learn/machine-learning [Coursera]
2	https://www.udemy.com/course/machinelearning/?couponCode=IND21PM [Udemy]

23UGOE2-CSE506-1 - Cyber Security and Cyber Laws

Click Syllabus Structure

Teaching Scheme
Lectures : 3 Hrs/Week
Credits : 3
Tutorials : --

Evaluation Scheme
ISE : 40 Marks
ESE : 60 Marks

Course Objective: The objective of this course is	
1	To gain knowledge about securing both clean and corrupted systems, protect personal data, and secure computer network
2	To examine secure software development practice
3	To understand key terms and concepts in Information Technology Act
4	To incorporate approaches for incident analysis and response

Course Outcomes:		
COs	**At the end of successful completion of the course, the students will be able to**	**Bloom's Taxonomy**
CO1	Comprehend computer security fundamentals, including threats and key terminology.	Remember
CO2	Examine different types of cybercrimes such as fraud, identity theft, DoS attacks, and malware, along with their mitigation techniques	Analyze
CO3	Use hacking methods like reconnaissance and penetration testing, and implement security tools such as firewalls, IDS, and VPNs.	Apply
CO4	Describe India's cyber laws, including the IT Act 2000, its amendments, and legal challenges related to cybercrime.	Understand
CO5	Examine the concepts of digital forensics & incident management	Analyze

Course Description:

This course is designed to provide a foundational understanding of Information and Network Security. It covers key concepts including types of threats, cybercrimes, malware, and hacking techniques. Students will also explore security technologies, legal perspectives, and digital forensics practices. The course equips learners with essential knowledge to identify, prevent, and respond to cyber threats.

Prerequisites:		
	1	Fundamental knowledge of Data Communication
	2	Networking and Information Security

Section – I

Unit-1	**Introduction to Computer Security**	
	Introduction, Identifying Types of Threats, Basic Security Terminology, Concepts and Approaches, Online Security Resources.	**4 Hours**
Unit-2	**Cyber Frauds, DoS, Viruses**	
	Introduction to Cyber Security, Cyber Stalking, Fraud, and Abuse: Introduction, How Internet Fraud Works, Identity Theft, Cyber Stalking, Protecting Yourself Against Cyber Crime. **Denial of Service Attacks**: Introduction, DoS, Illustrating an Attack **Malware**: Introduction, Viruses, Trojan Horses, The Buffer-Overflow Attack. The Sassier Virus/Buffer Overflow, Spyware, Other Forms of Malware, Detecting and Eliminating Viruses and Spyware.	**8 Hours**

Unit-3	**Hacking Methods and Strategies**	
	Introduction, Basic Terminology, The Reconnaissance Phase, Actual Attacks, Malware Creation, Penetration Testing	**6 Hours**
Section – II		
Unit-4	**Computer Security Technology**	
	Introduction, Virus Scanners, Firewalls, IDS, Digital Certificates, SSL/TLS, Virtual Private Networks, Wi-Fi Security.	**6 Hours**
Unit-5	**The Legal Perspectives of Cyber Crime**	
	Need of cyber laws, The Indian context, The Indian IT ACT: Admissibility of Electronic records, Amendments made in Indian ITA 2000, Positive Aspects and weak areas of ITA 2000, Challenges to Indian law and cybercrime scenario in India, Digital signatures and the Indian ITA act.	**6 Hours**
Unit-6	**Introduction to Digital Forensics**	
	Introduction, General Guidelines, Finding Evidence on the PC, Finding Evidence in System Logs, Getting Back Deleted Files, Operating System Utilities, Mobile Forensics: Cell Phone Concepts.	**6 Hours**

Mapping of POs & COs:

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

References	
Text Books :	
1	Computer Security Fundamentals - Chuck Easttom, Pearson, Third edition(Unit I to IV &VI)
2	Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Nina Godbole and Sunil Belapure, Wiley INDIA.(Unit V)
Reference Books :	
1	Jason Luttgens, Matthew Pepe, Kevin Mandia, Incident Response & Computer Forensics, McGraw-Hill Osborne Media, 3rd edition , 2014
2	Keith J. Jones, Richard Bejtlich, Curtis W. Rose, Real Digital Forensics: Computer Security and Incident Response, Paperback – Import, 2005.

3	Handbook of Applied Cryptography - Menezes, an Oorschot, and S.A. Vanstone
SWAYAM Courses (Operational Timestamp: Thu,29-May-2025 on 7:00 AM)	
1	https://onlinecourses.swayam2.ac.in/cec22_cs21/preview [AIHSHEW, Coimbatore]
2	https://onlinecourses.swayam2.ac.in/nou22_ge67/preview [DBAOU,Gujrat]

23UGPCC-CSE502P Database Engineering Lab

Teaching Scheme			Evaluation Scheme		
Practical	**:**	2 Hrs/Week	**ISA**	**:**	25 Marks
Credits	**:**	1	**POE**	**:**	25 Marks

Expt. No.	Experiment Title	Bloom's Taxonomy
1	Draw an E-R Diagram of any organization.	Analyse
2	Reduce above mentioned E-R Diagram into tables	Analyse
3	Normalize any database from first normal form to Boyce-Codd Normal Form (BCNF)	Understand
4	Use DDL Queries to create, alter (add, modify, rename, drop) & drop Tables	Understand
5	Use DML Queries to insert, delete, update & display records of the tables.	Analyze
6	Create table with integrity constraints like primary key, check, not null and unique	Analyze
7	Create table with referential integrity constraints with foreign key, on delete cascade and on delete set null	Understand
8	Display the results of set operations like union, intersections & set difference.	Understand
9	Display the results of Join Operations like cross join, self join, inner join, natural join, left outer join, right outer join and full outer join.	Understand
10	Display the records using Aggregate functions like min, max, avg, sum & count. Also use group by, having clauses	Understand
11	Display the results using String operations.	Apply
12	Demonstrate use of SQL Sub Queries.	Apply
13	Create & Update views (materialized and non-materialized) for any created table.	Apply
14	Create indices using SQL	Apply
15	Study of B+ tree indexing.	Apply

23UGPCC-CSE503T - Computer Algorithms

Teaching Scheme
Tutorial : 1 Hrs/week
Credits : 1

Evaluation Scheme:
ISA : 25 Marks

Term Work & Tutorial List	
1	**Algorithm Basics and Performance:** • Explain the importance of algorithms and how to analyze their time complexity. • Solve: Analyze the time complexity of binary search and linear search algorithms.
2	**Divide and Conquer:** • Describe the divide and conquer approach with examples like finding maximum/minimum in an array. • Solve: Perform Merge Sort on the array [38, 27, 43, 3, 9, 82, 10].
3	**Binary Search Algorithm:** • Explain binary search with its recursive and iterative forms. • Solve: Trace the steps of binary search for finding 43 in the sorted array [10, 20, 30, 40, 43, 50].
4	**Greedy Method and Knapsack Problem:** • Explain the greedy method using the fractional knapsack problem. • Solve: Given weights and profits, use the greedy approach to maximize profit in fractional knapsack.
5	**Minimum Spanning Tree Algorithms:** • Discuss Prim's and Kruskal's algorithms and their applications. • Solve: Find MST for a given graph with vertices and edges (provide a simple weighted graph).
6	**Dynamic Programming Introduction:** • Explain the concept of overlapping subproblems and optimal substructure. • Solve: Use dynamic programming to find the shortest path in a multistage graph (simple example).
7	**All Pair Shortest Path (Floyd Warshall):** • Explain Floyd Warshall algorithm and its time complexity. • Solve: Apply Floyd Warshall to the given adjacency matrix and find shortest paths.
8	**Tree and Graph Traversal:** • Explain BFS and DFS with examples in graphs and binary trees. • Solve: Perform BFS and DFS traversal on the provided graph and write the traversal order.
9	**Backtracking Method:** • Explain backtracking with the N-Queens problem and sum of subsets. • Solve: Write steps to place 4 queens on a 4x4 chessboard without conflict.
10	**Hamiltonian Cycle Problem:** • Define Hamiltonian Cycle and explain why it is NP-Complete. • Solve: Try to find a Hamiltonian cycle in a small given graph (example with 5 vertices).
11	**NP Problems Overview:** • Differentiate between P, NP, NP-Complete, and NP-Hard problems with examples. • Write a short note on why solving NP-Complete problems efficiently is challenging.