Major Core Courses (MCC)

Course Code	Course Name	Level	L	T	P	C	CIE	SEE	Total	Pre-requisite
2501MA01	Linear algebra &Calculus	FC	2	1		3	50	50	100	-
2501PH02	Modern Physics	FC	2		1	3	50	50	100	-
2501CS01	Programming for Problem Solving Using C	FC	2		2	4	50	50	100	-
2501IT01	Business Intelligence Lab	FC			2	2	50	50	100	-
2501ME01	Engineering Graphics	FC	1		2	3	50	50	100	-
2501MA02	Differential Equations &Vector Calculus	FC	2	1		3	50	50	100	-
2501CS03	Data Structures	FC	2		2	4	50	50	100	PPSC
2501IT42	Programming with Python	FC	2		2	4	50	50	100	PPSC
2501CS71	Computer Organization & Architecture	FC	2	1		3	50	50	100	DLD
2501MA08	Discrete Mathematics	IC	2	1		3	50	50	100	LAC
2501CS08	Object Oriented Programming through C++	IC	2		2	4	50	50	100	PPSC
2501CS11	Theory of Computation	IC	2	1		3	50	50	100	DMS
2501IT05	Database Management Systems	IC	2		2	4	50	50	100	PPSC
2501IT07	Agile Software Engineering	IC	2		1	3	50	50	100	PPSC
2501MA09	Probability & Statistics	IC	2	1		3	50	50	100	LAC
2501IT06	Java Programming	IC	2		2	4	50	50	100	PPSC
2501CS13	Operating Systems	IC	2		1	3	50	50	100	-
2501AI02	Artificial Intelligence	IC	2		1	3	50	50	100	-
2501CS07	Computer Networks	IC	2		1	3	50	50	100	-
2501AI03	Data Mining	IC	1		2	3	50	50	100	-
2501CS10	Advanced Data Structures & Algorithm Analysis	AC	2		1	3	50	50	100	DS
2501IT04	Compiler Design	AC	2		1	3	50	50	100	TOC
2501AI05	Machine Learning	AC	2		2	4	50	50	100	DAE
2501AI04	Big Data Analytics	AC	2		1	3	50	50	100	DM
2501IT03	Cryptography & Network Security	AC	2		1	3	50	50	100	CN
2501CS15	Software Architecture	AC	2		1	3	50	50	100	ASE
	Total		**47**	**6**	**30**	**84**				

Linear Algebra & Calculus

(Common to CE,EEE,ME,ECE,CSE,IT,AIML,CSE(DS),PT&Min.E)

Course Code: 2501MA01	L	T	P	C
	2	1	0	3

Course Outcomes:
At the end of the Course, Student will be able to:

CO1: Solve the system of Linear equations
CO2: Calculate Eigen values and Eigen vectors
CO3: Apply differential calculus for one and several variable functions
CO4: Calculate the Maximum value and Minimum value of a function of several variables
CO5: Compute areas and volumes using multiple integrals

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2									
CO2	3	2									
CO3	3	2									
CO4	3	2									
CO5	3	2									

UNIT – I

System of linear equations: Vector Space,Linear Independence,Rank of a matrix by echelon form, normal form, Inverse of Non-singular matrices by Gauss-Jordan method, Solutions of Linear Systems: Existence, Uniqueness,Solving the system by Gauss elimination method.

Practice(Using any computational tool)

1. Variables, arithmetic operations, elementary mathematical functions.
2. Defining row vector, column vector, Arithmetic operations on matrices
3. finding transpose of a matrix, inverse of a matrix, determinant of a matrix
4. rank of a matrix, solving system of linear equations.

UNIT – II

Eigenvalues, Eigvectors : Eigenvalues and properties(without proof), Eigenvectors, Diagonalization of a matrix, Cayley-Hamilton Theorem (without proof), Quadratic forms, Reduction of Quadratic form to canonical forms by Orthogonal Transformation,Nature of Quadratic forms.

Practice(Using any computational tool):

1. Computing eigen values and eigen vectors, matrix diagonalization.

UNIT-III

One Variable Calculus:Cauchy's mean value theorem, Taylor's and Maclaurin theorems with remainders (without proof), Problems and applications on the above theorems.

Several Variable Calculus: Limit, Continuity, partial derivatives and their geometrical interpretation.

Practice(Using any computational tool): Basics of plotting,Plot graphs of single variable functions

UNIT – IV

Functions of several variables: Total differential and differentiability,derivatives of composite and implicit functions,derivatives of higher order and their commutativity,Euler's theorem on homogeneous functions , Taylor's and Maclaurin's expansion of functions of two variables. Jacobians, maxima and minima, constrained maxima/minima problems using Lagrange's method of multipliers.

Practice(Using any computational tool): Plot graphs of various multi variable functions

UNIT – V

Multiple Integrals: Double integrals, triple integrals, change of order of integration, change of variables to polar, cylindrical and spherical coordinates. Finding areas (by double integrals) and volumes (by double integrals and triple integrals).

Practice(Using any computational tool): Plotting the region of Integration

Students are advised to use any computational / AI Tool like Wolfrum Alpha, Symbolab, Mathway, Desmos, Geogebra etc., for the practice

Text Books:

1. Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Alpha Science International Ltd., 5th Edition (9th reprint), 2021· ISBN 978-8184875607
2. Advanced Modern Engineering Mathematics, Glyn James, Pearson publishers, 5th Edition, 2018. ISBN-13. 978-1292174341

Reference Books:

1. Advanced Engineering Mathematics, Michael Greenberg, Pearson publishers, 9th edition. ISBN-13. 9788177585469
2. Higher Engineering Mathematics, H. K. Dass, Er. R. Verma, S-Chand publishers, 3rd edition 2023.ISBN 9788121938907

Web Links:

1. https://archive.nptel.ac.in/courses/111/104/111104137/
2. https://archive.nptel.ac.in/courses/111/107/111107108/
3. https://www.khanacademy.org/math/linear-algebra/
4. https://www.khanacademy.org/math/multivariable-calculus

Modern Physics

(Common to EEE, ECE, CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code:2501PH02	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the Course, Student will be able to:

- **CO1** Apply the principles of interference and diffraction to design and enhance the resolving power of grating
- **CO2** Familiarize the concepts of LASER's and Optical fibres with applications.
- **CO3** Explain the fundamental concepts of Quantum behavior of matter
- **CO4** Differentiate various electron theories to understand the properties of solids.
- **CO5** Explain the basic concepts of Semiconductors and identify the type of semiconductors using Hall effect.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	1	-	-	-	-	-	1	1	-	-
CO2	2	1	-	-	-	-	-	1	1	-	-
CO3	2	1	-	-	-	-	-	1	1	-	-
CO4	2	1	-	-	-	-	-	1	1	-	-
CO5	2	1	-	-	-	-	-	1	1	-	-

UNIT–I

Wave Optics

Interference: Introduction - Principle of Superposition - Interference of light- Conditions for sustained Interference- Interference in thin films (reflected geometry) – Colours in thin films – Newton's Rings (reflected geometry) – Determination of wavelength and refractive index – Applications.

Diffraction: Introduction – Fresnel and Fraunhofer Diffraction - Diffraction due to Single slit (quantitative), Double slit (qualitative) - N–Slits Intensity distribution curves (qualitative) – Grating spectrum–Rayleigh's criterion - Resolving powers of grating (qualitative) - Applications.

Practice:

1. Determination of Radius of curvature of a plano-convex lens using Newton's Rings.
2. Measurement of Width or thickness of a thin wire by forming interference fringes.
3. Determination of Wavelength of light source using Diffraction grating by normal incidence method.

4. Resolving power of grating.
5. Determination of wavelength of light by using prism.

UNIT–II

Lasers & Optical Fibers

Lasers: Introduction - Characteristics of Lasers – Spontaneous and Stimulated emission of radiation – population inversion- Einstein's coefficients – Relation - significance - Pumping Mechanisms - Ruby laser – Helium-Neon laser - Applications.

Optical Fibers: Introduction – Principle of propagation in Optical Fiber – Angle of acceptance – Expression for Numerical Aperture and condition for propagation – Classification of Optical fibers - Applications.

Practice:

1. Determination of Wavelength of He-Ne laser source by using diffraction grating.

UNIT–III

Quantum Mechanics

Introduction – Matter waves – de Broglie's hypothesis – Davisson and Germer Experiment - Heisenberg's Uncertainty Principle – interpretation of wave function – Schrödinger Time dependent and Time Independent wave equations– Particle in a potential box.

Unit–IV

Free electron Theory

Free Electron Theory: Introduction–Classical free electron theory (merits and demerits only) -Quantum Free electron theory – Electrical conductivity- Fermi energy state–Fermi Dirac distribution function - Temperature dependence – Density of states.

Band theory of solids: Bloch Theorem – Origin of energy bands in crystalline solids – classification of crystalline solids.

Practice:

1. Study the variation of magnetic field along the axis of a circular coil carrying current by using Stewart and Gee's apparatus.
2. Determination of Frequency of electrically maintained tuning fork by Melde's apparatus.

UNIT–V

Semiconductor Physics

Semiconductors Introduction – Intrinsic semiconductors – density of charge carriers (Qualitative only) – Electrical conductivity –Fermi level - extrinsic semiconductors - P-type & N-type semiconductors-Density of charge carriers (Qualitative only) - Dependence of Fermi energy on carrier concentration and temperature–Hall effect-Hall coefficient - Applications of Hall effect–Drift and Diffusion currents - Einstein's equation

Semiconductor Devices Working of PN junction diode – Forward and reverse bias Zener diode -Metal-Oxide-Semiconductor (MOS) structure- Capacitance-voltage characteristics- MOSFET structure - I–V characteristics.

Practice:

1. Determination of V-I characteristics and Breakdown voltage of a Zener diode.
2. Determination of Energy band gap of a semiconductor by using P-N junction diode.
3. Study the relation between Temperature and resistance and finding the constants A & B of a thermistor.
4. Determine the resistivity of a semiconductor by four probe method.

Textbooks:

1. Engineering Physics, M N Avadhanulu & T.V.S. Arun Murthy, S Chand &Company Ltd, 1st edition 2024. ISBN: 978-93-5870-932-3
2. Engineering Physics, Satyendra Sharma and Jyotshna Sharma, Pearson publications. ISBN: 978-81-3151-178-7

Reference Books:

1. Concepts of Modern Physics, Authur Beiser, Shobhit Mahajan and S Rai Choudhary, McGraw Hill (2017). ISBN: 9789351341857
2. Engineering Physics, M.R. Srinivasan, New Age international publishers (2009). ISBN: 978-1848290501
3. Optics, Ajoy Ghatak, 6th Edition McGraw Hill Education, 2017. ISBN: 978-9390113590

Web Links:

1. http://nptel.ac.in/courses/122107035/11
2. http://nptel.ac.in/courses/115102023/-
3. https://phet.colorado.edu/en/simulations/category/physics
4. http://physicsgecg.blogspot.in/p/reading-materials.html

Programming for Problem Solving using C
(Common to all)

	L	T	P	C
Course Code: 241CS001	**2**	**0**	**2**	**4**

Course Outcomes: At the end of the Course, Student will be able to:

- **CO1:** Demonstrate basics of computer, algorithm and flow chart for problem solving.
- **CO2:** Make use of an appropriate control structures to solve given problems.
- **CO3:** Solve complex problems using arrays and strings.
- **CO4:** Develop modular programming using functions.
- **CO5:** Demonstrate dynamic memory allocations and file handling using file operations.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	3	2	1	3	-	-	2	-	-	1
CO2	2	3	2	1	3	-	-	2	-	-	1
CO3	2	3	2	1	3	-	1	2	-	-	1
CO4	2	3	2	1	3	-	1	2	-	-	1
CO5	2	3	2	1	3	-	1	2	-	-	1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	2
CO2	2	2
CO3	2	2
CO4	2	2
CO5	2	2

UNIT – I

Introduction to Programming and Problem Solving

Introduction to Programming Languages, Basics of a Computer Program- Algorithms, Algorithmic approach, characteristics of algorithm, Problem solving strategies: Top-down approach, Bottom-up approach, Time and space complexities of algorithms, flowcharts (Using Dia Tool), pseudo code. Structure of C Program, Introduction to Compilation and Execution, Primitive Data Types, Variables, and Constants, Basic Input and Output, operators, keywords, identifiers, Type Conversion, and Casting.

Practice 1:

1. Explore different platforms
   a. Basic linux environment and its editors like Vi, Vim & Emacs etc.
   b. Exposure to turbo C, gcc.
   c. Explore to hacker rank or any other Online coding platform and compiler environment.
   d. “Hello world” in C.
   e. Objective: Learn about the syntax of reading from stdin and writing to stdout. https://www.hackerrank.com/challenges/hello-world-c/problem?isFullScreen=true

f. Write a simple program to read int, float, char and string using scanf() and display using printf() in all the above given platforms.

2. Basics and Operators
   a. Sum and Difference of 2 numbers.
      Objective: Learn int and float data types.
      https://www.hackerrank.com/challenges/sum-numbers-c/problem?isFullScreen=true
   b. Playing with Characters.
      Objective: Learn how to take a character, a string and a sentence as input in C.
      https://www.hackerrank.com/challenges/playing-with-characters/problem?isFullScreen=true
   c. Bitwise Operators
      Objective: Learn how to work with bits (0,1) and bitwise operators.
      https://www.hackerrank.com/challenges/bitwise-operators-in-c/problem?isFullScreen=true
   d. Conversion of Fahrenheit to Celsius and vice versa.
   e. Distance travelled by an object.
   f. Calculate Simple interest and compound interest.

3. Operators and Expressions, Variables and Type conversions.
   a. Evaluate the following expressions
      i. a/b*c-b+a*d/3
      ii. j = (i++) + (++i)
   b. Square root of a given number.
   c. Find the area of circle, square, rectangle and triangle.
   d. Find the maximum of three numbers using conditional operator.
   e. Take marks of 5 subjects in integers, find the total in integer and average in float.

UNIT – II

Control Structures

Simple sequential programs, Conditional Statements (if, if-else, else if ladder, switch), Loops (for, nested for loop, while, do-while), break and continue, goto statement.

Practice:

1. Conditional Statements
   a. Objective: Understand if and else Conditional statements in C.
      https://www.hackerrank.com/challenges/conditional-statements-in-c/problem?isFullScreen=true
   b. Roots of a Quadratic Equation.
   c. Generate electricity bill.
   d. Simulate a calculator using switch case.
   e. Find the given year is a leap year or not.
2. Loops
   a. Objective: Learn the usage of the for loop in C.
      https://www.hackerrank.com/challenges/for-loop-in-c/problem?isFullScreen=true
   b. Sum of the digits of a 5-digit number.
      Objective: Learn the usage of while loop and usage of operators - % and /.
      https://www.hackerrank.com/challenges/sum-of-digits-of-a-five-digit-number/problem?isFullScreen=true

c. Given number is a prime or not. (Also Prime numbers between a given range.)
d. Armstrong Number or not.
e. Palindrome or not.
f. Objective: Print a pattern of numbers using Loops.
https://www.hackerrank.com/challenges/printing-pattern-2/problem?isFullScreen=true
g. Construct a Pyramid pattern.

UNIT – III

Arrays: Arrays indexing, Accessing programs with array of integers, two dimensional arrays, Introduction to Strings, string handling functions.
Sorting Techniques: bubble sort, selection sort.
Searching Techniques: linear, Binary search.

Practice:

1. 1-Demensional Arrays
   a. Objective: Print the sum and free the memory where the array is stored.
   https://www.hackerrank.com/challenges/1d-arrays-in-c/problem?isFullScreen=true
   b. Objective: Working with indices in array.
   https://www.hackerrank.com/challenges/reverse-array-c/problem?isFullScreen=true
   c. Search an element in array (Linear Search).
   d. Find min and max elements in array.
   e. Insert an element into array.
   f. Eliminate duplicate elements from array.
   g. Sorting of elements in an array using Bubble sort.
2. 2-Demensional Arrays
   a. Sum of two 2-D arrays.
   b. Multiplication of two 2-D arrays.
   c. Transpose of a Matrix.
   d. Trace of a Matrix.
   e. Lower Triangular Matrix.
3. Hacker Rank
   a. Objective: print each word of the sentence in a new line.
   https://www.hackerrank.com/challenges/printing-tokens- /problem?isFullScreen=true
   b. Count number of alphabets (lowercase, uppercase, consonants, vowels) and digits Lowercase to Uppercase, Uppercase to Lowercase, Toggle case, Sentential case
   c. Objective:Digit Frequency find the frequency of each digit in the given string.
   https://www.hackerrank.com/challenges/frequency-of-digits-1/problem?isFullScreen=true
   d. Find string length, concatenate 2 strings, reverse a string using built-in and without built-in string functions.

UNIT – IV

Functions: Introduction to Functions, Function Declaration and Definition, Function call Return **Types and Arguments**, arrays as parameters, Scope and Lifetime of Variables, **storage classes**, **recursion**, functions and arrays.

Practice: Functions in C

1. Objective: Learn simple usage of functions.
   https://www.hackerrank.com/challenges/functions-in-c/problem?isFullScreen=true

2. Objective: Fibonacci Numbers using recursive function.
   https://www.hackerrank.com/challenges/ctci-fibonacci-numbers/problem
3. Objective: N^{th} factorial using recursion.
   https://www.hackerrank.com/contests/ccc-veltech-practice-set-ende/challenges/factorial-using-recursion-1
4. Objective: Find the super digit of the integer.
   https://www.hackerrank.com/challenges/recursive-digit-sum/problem
5. Implement LCM
6. Objective: Calculate the Nth term of series.
   https://www.hackerrank.com/challenges/recursion-in-c/problem?isFullScreen=true

UNIT – V

Pointer: Introduction to Pointers, dereferencing and address operators, pointer and address arithmetic, array manipulation using pointers, **functions & pointers** modifying parameters inside functions using pointers, Command line Arguments, Dynamic memory allocation, Null Pointer, generic pointer, dangling pointer.
File Handling:-Introduction to Files, Using Files in C, Reading from Text Files, Writing to Text Files, Random File Access.

Practice:

1. Pointers
   a. Objective: learn to implement the basic functionalities of pointers in C.
      https://www.hackerrank.com/challenges/pointer-in-c/problem?isFullScreen=true
   b. Objective: Learn using Pointers with Arrays and Functions.
      https://www.hackerrank.com/challenges/students-marks-sum/problem?isFullScreen=true
   c. Objective: sort a given array of strings into lexicographically increasing order or into an order in which the string with the lowest length appears first.
      https://www.hackerrank.com/challenges/sorting-array-of-strings/problem?isFullScreen=true
   d. Find the sum of a 1D array using malloc().
   e. Swap two numbers using functions and pointers - call by value and reference.
   f. Objective: Dynamic Handling requests by a Librarian to place the books in the shelves.
      https://www.hackerrank.com/challenges/dynamic-array-in-c/problem?isFullScreen=true

2. File handling concepts
   a) Write text into and read text from a file.
   b) Write text into and read text from a binary file using fread() and fwrite().
   c) Copy the contents of one file to another file.
   d) Merge two files into the third file using command-line arguments
   e) Find no. of lines, words and characters in a file.

Additional Practice:

1. Variadic functions in C
   Objective: Understanding variable number of arguments
   https://www.hackerrank.com/challenges/variadic-functions-in-c/problem?isFullScreen=true
2. Small Triangles, Large Triangles
   Objective: Print sorted by their areas

https://www.hackerrank.com/challenges/small-triangles-large-triangles/problem?isFullScreen=true

3. Permutations of Strings
   Objective: print all strings permutations in strict lexicographical order
   https://www.hackerrank.com/challenges/permutations-of-strings/problem?isFullScreen=true

Text Books:

1 Programming in C, Rema Theraja, Oxford, 2nd edition. ISBN 93-5497-9
2 "The C Programming Language", Brian W. Kernighan and Dennis M. Ritchie, Prentice-Hall. ISBN 13: 9780131103627

Reference Books:

1. Computing fundamentals and C Programming, Balagurusamy, E., McGraw-Hill Education. ISBN.No: 9352604172
2. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill. ISBN No. 0071367993
3. Let Us C Yashwanth,Kanetkar, Eighth edition, BPB Publications.ISBN No. 1934015253
4. Programming in C A-Practical Approach, Ajay Mittal. Pearson Education.ISBN No. 9788131729342
5. R G Dromey How to Solve It by Computer (Prentice-Hall International Series in Computer Science.ISBN-13 : 978-0134340012

Web Links:

1 https://www.hackerrank.com/
2 https://onlinecourses.nptel.ac.in/noc22_cs40/preview
3 https://archive.nptel.ac.in/courses/106/104/106104128/

Business Intelligence Lab
(Common to CE,EEE,ME,ECE,CSE,IT,AIML,CSE(DS),PT&Min.E)

	L	T	P	C
Course Code: 2501IT01	**0**	**0**	**2**	**2**

Course Outcomes:
At the end of the course, student will be able to:

- **CO1:** Utilize Excel and Power BI for data analysis, visualization, and reporting.
- **CO2:** Apply various data analysis techniques in Excel and Power BI to extract meaningful insights from datasets.
- **CO3:** Create clear and compelling visualizations using Excel and Power BI to communicate data-driven insights.
- **CO4:** Develop data models in Power BI to organize and analyze data efficiently.
- **CO5:** Design interactive dashboards in Power BI to facilitate data exploration and decision-making.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	2			2				1	1	1
CO2	2	3			2				1	1	2
CO3	2	2			2				1	1	2
CO4	1	2	1		2				1	1	2
CO5	1	2	1		2				1	1	2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	1	2
CO2	2	1
CO3	2	2
CO4	1	2
CO5	2	2

Practice:

1. Introduction to Excel

a. Overview of Excel and its capabilities for data analysis.
b. Basics of Excel: Navigating the interface, entering data, formatting cells.
c. Introduction to functions and formulas: SUM, AVERAGE, IF, VLOOKUP, etc.
d. Importing data into Excel from different sources: CSV, text files.

2. Data Analysis with Excel

a. Data manipulation techniques: Sorting, filtering, and grouping data.
b. Advanced functions and formulas: INDEX/MATCH, SUMIFS, COUNTIFS, etc.
c. Data visualization with Excel: Creating charts and graphs.
d. Using PivotTables for data summarization and analysis.

3.Advanced Excel Features

a. Introduction to Excel tables and structured references.
b. Working with named ranges and dynamic ranges.
c. Excel data validation techniques.

4. Introduction to Power BI

a. Overview of Power BI and its advantages over Excel for large datasets.
b. Installing Power BI Desktop.
c. Understanding the Power BI interface: Navigation, ribbons, and panes.
d. Importing data into Power BI Desktop from various sources.

5. Data Preparation in Power BI

a. Introduction to Power Query for data transformation.
b. Cleaning, shaping, and filtering data in Power Query Editor.

6. Data Preparation in Power BI

a. Combining data from different sources.
b. Loading data into Power BI model.

7. Data Modeling in Power BI

a. Understanding relationships between tables.
b. Creating calculated columns and measures using DAX.

8. Data Modeling in Power BI

a. Introduction to DAX functions: CALCULATE, FILTER, RELATED, etc.
b. Working with date and time functions in DAX.

9. Visualization Basics in Power BI

a. Creating basic visualizations: Bar charts, line charts, pie charts, etc.
b. Customizing visualizations: Formatting, titles, legends, etc.

10. Visualization Basics in Power BI

a. Using slicers and filters to interact with visualizations.
b. Adding drill-down capabilities to visualizations.

11. Advanced Visualizations and Dashboards in Power BI

a. Exploring advanced visualizations: TreeMap, Waterfall chart, KPIs, etc.
b. Creating custom visuals from the marketplace.

12. Advanced Visualizations and Dashboards in Power BI

a. Designing effective dashboards: Layout, arrangement, and organization.
b. Adding interactivity with bookmarks and drill-through.

Additional Practice:

1. **Basic Data Analysis:** Import a dataset into Excel and perform basic data analysis tasks such as sorting, filtering, and creating simple charts to visualize the data.
2. **Expense Tracker:** Create a spreadsheet to track your expenses. You can have columns for date, item description, category, and amount. Use formulas to calculate totals and analyze your spending habits.
3. **Data modeling and extracting statistics from dataset:** Connecting Power BI to local data files and cloud servers (COVID19 dataset will be imported into the Power BI for visualization).

Text Books:

1	Learn Power BI - Second Edition: A comprehensive, step-by-step guide for beginners to learn real-world business intelligence 2nd Edition, ISBN: 9781801811958.
2	Power BI Beginner: Zero to Hero in Power BI Desktop by Philip Seamark, ISBN 1691641227.

Reference Books:

1	Power BI Quick Start Guide: Build dashboards and visualizations to make your data come to life" by Devin Knight and Siddharth Mehta.
2	Learn Power BI: A Beginner's Guide to Analyzing Data and Creating Reports with Power BI" by Murilo Miranda.

Web Links:

1	https://learn.microsoft.com/en-us/power-bi/
2	https://support.microsoft.com/en-us/excel
3	https://cce.sydney.edu.au/course/MSE1
4	https://cce.sydney.edu.au/course/PBBA

Engineering Graphics

(Common to CE,EEE,ME,ECE,CSE,IT,AIML,CSE(DS),PT&Min.E)

	L	T	P	C
Course Code: 2501ME01	**1**	**0**	**2**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Apply the principles of engineering drawing to construct Engineering curves
CO2: Construct projections of points and lines.
CO3: Demonstrate visualization skills of projections of planes.
CO4: Demonstrate visualization skills of projections of solids and development of surfaces
CO5: Construct isometric and orthographic views of simple solids.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2	1		2				1	1	
CO2	3	2	1		2				1	1	
CO3	3	2	1		2				1	1	
CO4	3	2	1		2				1	1	
CO5	3	2	1		2				1	1	

Practice:

a) Introduction to Engineering Graphics
Introduction to AutoCAD, Dimensioning, elements of dimensioning, system of dimensioning, and Conventions in Drawing.

b) Construction of Cycloids and Involutes
a) Construction of Cycloid, Epicycloid and Hypocycloid
b) Involute of a pentagon and circle

c) Introduction to Orthographic Projection
a) Principles of Projection
b) Orthographic Projection – Four Quadrants.
c) First angle & Third angle Projection with examples, reference plane, importance of reference lines or Plane.
d) Projections of a point situated in any one of the four quadrants.

d) Projection of straight lines-I:
a) Projections of straight lines parallel to both reference planes.
b) Projections of straight lines perpendicular to one reference plane and parallel to other reference plane
c) Projections of straight line parallel to one plane & inclined to another plane

e) Projection of straight lines-II:

a) Projections of straight line inclined to both reference planes

f) Projection of planes:

a) Regular planes perpendicular to both reference planes, Parallel to one reference plane and inclined to the other reference plane
   Ex: Rectangle, Pentagon, Hexagon and Rhombus

g) Projection of planes

a) Projections of Planes inclined to both reference planes Ex: Rectangle, Pentagon, Hexagon and Rhombus.

h) Projection of solids

a) Axis Perpendicular to H.P and Axis Perpendicular to V.P, Ex: Pentagonal and Hexagonal Prisms, Pyramids, Cylinder and Cone
b) Axis Parallel to H.P and V.P, Pentagonal and Hexagonal Prisms, Pyramids, Cylinder and Cone

i) Projection of solids

a) Projection of Solids with axis inclined to one reference plane and parallel to another plane. Ex: Pentagonal and Hexagonal Prisms, Pyramids, Cylinder and Cone

j) Development of Surfaces

a) Development of Prisms and Cylinder simple cases

k) Development of Surfaces

a) Development of Pyramids and Cone simple cases

l) Conversion of Isometric views to Orthographic views

a) Practice figure - 1
b) Practice figure – 2

Additional Practice:

1. Conversion of Isometric views to Orthographic views

a) Practice figure - 3
b) Practice figure - 4

2. Conversion of Orthographic views to Isometric views

a) Practice figure - 1
b) Practice figure - 2

Text Books:

1. Engineering Drawing, N. D. Bhatt, Charotar Publishing House, 54th edition, 2024, ISBN : 9789385039706.
2. Engineering Drawing and Graphics , Venugopal, New Age Publications, 2nd edition, 2019, ISBN: 9788122415452.

Reference Books:

1. Engineering Drawing, K.L. Narayana and P. Kannaiah, Tata McGraw Hill, 2021, ISBN: 978-9385983177.
2. Computer Aided Engineering Graphics, T. Jeyapoovan,Vikas Publishing house, New Delhi, 1stEdition, 2023, ISBN : 9789356743199.

Web Links:

1. https://nptel.ac.in/courses/112103019/
2. https://academy.autodesk.com/authenticated-home-user
3. https://www.sciencedirect.com/book/9780080108391/engineering-drawing-from-thebeginning

Differential Equations & Vector Calculus
(Common to CE,EEE,ME,ECE,CSE,IT,AIML,CSE(DS),PT&Min.E)

	L	T	P	C
Course Code: 2501MA02	**2**	**1**	**0**	**3**

Course Outcomes:
At the end of the Course, Student will be able to:

CO1: Solve the linear differential equations of first order and apply in various engineering problems.
CO2: Solve the linear differential equations of higher order and and apply in various engineering problems.
CO3: Solve the linear partial differential equations
CO4: Calculate the gradient, divergence and curl.
CO5: Compute work done, flux using vector integration

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2									
CO2	3	2									
CO3	3	2									
CO4	3	2									
CO5	3	2									

UNIT – I

Ordinary Differential Equations of First Order and First Degree:
Solution of first order linear differential equations, exact differential equations and equations reducible to exact differential equations, Orthogonal Trajectories, Modelling of RL- circuit.
Practice(Using any computational tool): Solving the first order initial value problems using **odesolver** and plot the solution curves.

UNIT – II

Linear Differential Equations of Higher Order:
Solution of linear differential equations with constant coefficients, method of variation of parameters, solution of simultaneous linear differential equations.
Equations reducible to Linear differential equations with constant coefficients: Cauchy's homogeneous Linear Equations, Legendere's Linear Equations. Study of oscillations arising in LCR circuit (free oscillations and forced oscillations).

Practice(Using any computational tool): Solving the second order initial value problems using **odesolver** and plot the solution curves

UNIT – III

Partial Differential Equations:
Solution of linear PDE of first order by Lagrange's method, solution of homogeneous linear PDE of higher order with constant coefficients.

UNIT – IV

Vector Differentiation:
Gradient of a scalar field, finding angle between two surfaces, directional derivative. Divergence and solenoidal fields. Curl and irrotational fields, Finding Scalar Potential.
Practice(Using any computational tool): Plotting of surfaces, 3D-plots, plotting vector fields.

UNIT – V

Vector Integration:
Line integrals, work done by a force, conservative force field, surface integral, flux, volume integral. Green's theorem, Stoke's theorem and Gauss divergence theorem.

Students are advised to use any computational / AI Tool like Wolfrum Alpha, Symbolab, Mathway, Desmos, Geogebra etc., for the practice

Text Books:

1	Advanced Engineering Mathematics, E. Kreyszig, John Willey & Sons,10th Ed., 2018. ISBN 978-0470458365
2	Higher Engineering Mathematics, B. S. Grewal, Khanna Publishers, 44'th Edition (2021). ISBN 978-9383214204

Reference Books:

1	Advanced Engineering Mathematics, Dennis G. Zill, Jones & Bartlett Learning, 2018, 6th Edition.ISBN 978-1284105902.
2	Higher Engineering Mathematics, B.V. Ramana, McGraw-Hill Education, 11'th Ed., 2017. ISBN 978-9339216016.

Web Links:

1	https://www.classcentral.com/course/differential-equations-engineers-13258
2	https://archive.nptel.ac.in/courses/111/106/111106100/
3	https://www.khanacademy.org/math/differential-equations
4	https://archive.nptel.ac.in/courses/111/101/111101153/
5	https://archive.nptel.ac.in/courses/111/105/111105122/

Data Structures

(Common to EEE, ECE, CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501CS03	**2**	**0**	**2**	**4**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Utilize structure, union to handle heterogeneous data.
CO2: Illustrate Time and Space complexities for different sorting Algorithms
CO3: Demonstrate various operations on Linked Lists
CO4: Explain different operations on Stack and Queue with applications.
CO5: Demonstrate the importance and various operation on non-linear data structures

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	1	1	2					1	1		
CO2	1	1	2	2				1	1		
CO3	1	2	2	1				1	1		
CO4	1	1	2	1				1	1		
CO5	1	1	2	2				1	1		

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	-
CO2	2	-
CO3	2	-
CO4	2	-
CO5	2	-

UNIT – I

Structures and Unions: Introduction, Nested Structures, Arrays of Structures, Structures and Functions, Self-Referential Structures, Unions, Enumerated Data Type - enum variables, Using Typedef keyword, Bit Fields.
Data Structures: Introduction to Data Structures, Types of Data Structures.

Practice:
1. Write a C program to find the total, average of n students using structures
2. Copy one structure variable to another structure of the same type.
3. Read student name and marks from the command line and display the student details along with the total.

UNIT – II

Introduction to Linear Data Structures: Definition and importance of linear data structures, Abstract data types (ADTs) and their implementation, Overview of time and space complexity analysis for linear data structures.
Sorting Techniques: Quick sort, Merge sort, Radix sort

Practice:
1. Implement Merge sort using arrays.
2. Implement Quick sort using arrays
3. Implement Radix Sort using arrays

UNIT – III

Linked Lists: Singly linked lists: representation and operations, doubly linked lists and its operations and circular linked lists and its operations, Comparing arrays and linked lists, Applications of linked lists.

Practice:
1. Single Linked List: Perform different operations in single linked list.
2. Reversing a single linked list
3. Perform different operations in double linked list
4. Circular Linked List

UNIT – IV

Stacks: Introduction to stacks: properties and operations, implementing stacks using arrays and linked lists, applications of stacks.
Queues: Introduction to queues: properties and operations, Types of Queues, implementing queues using arrays and linked lists, applications of queues.

Practice:
1. Stack and its operations using arrays and Linked List
2. Implement a program to evaluate a postfix expression.
3. Queue and its operations using arrays and Linked List
4. Implement Circular Queue using Arrays

UNIT – V

Non-linear Data Structures:
Trees: Difinition of tree, Tree Terminology, types of trees, Binary tree traversals, Binary Search Tree – Insertion, Deletion.
Graphs: Definition and Terminology – Representation of Graphs-Adjacency Matrix and Linked list, Graph Traversals (BFT & DFT)

Practice:
1. Implement Binary search tree (BST).
2. Implement Binary search Tree (BST) Traversals.
3. Graph Traversal using Breadth First Search (BFS)
4. Graph Traversal using Depth First Search (DFS)

Additional Practice:

1. Using a structure for transporting some boxes through a tunnel
2. Delete duplicate-value nodes from a sorted linked list
3. Convert an infix expression into postfix expression
4. De Queue Implementation

Text Books:

1. Data Structures using C, Rema Theraja, Oxford University Press, 3rd Edition. ISBN: 978-9354979453
2. Data Structures, KV Sambasivarao, S Rama Sree, S.Chand. ISBN: 978-9358704730

Reference Books:

1. The Algorithm Design Manual, Steven S. Skiena, Springer Publication, Second Edition. ISBN: 978-1849967204.
2. Fundamentals of data structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Press, ISBN: 978-0716782506.
3. Data Structures and Algorithms by Maganti Venkatesh, Naresh. ISBN: 9780929306407.
4. Data Structures Using C,Reema Thareja, Oxford University Press, 2nd Edition. ISBN:978-0198099307.

Web Links:

1. https://nptel.ac.in/courses/106102064
2. https://archive.nptel.ac.in/courses/106/105/106105225/
3. https://www.udemy.com/topic/data-structures/
4. https://www.coursera.org/specializations/data-structures-algorithms
5. https://ds1-iiith.vlabs.ac.in/List%20of%20experiments.html

Programming with Python
(Common to CSE, IT, AI&ML and DS)

	L	T	P	C
Course Code: 2501IT42	**2**	**0**	**2**	**4**

Course Outcomes: At the end of the Course, Student will be able to:

CO1: Develop programs using fundamental concepts in python.
CO2: Solve problems using control statements and string methods.
CO3: Develop real time applications using data structures and functions.
CO4: Apply Object Oriented Programming concepts and files.
CO5: Build various applications using GUI and exceptions.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	2	3	-	-	-	-	-	-	-	1
CO2	2	2	3	-	-	-	-	-	-	-	1
CO3	3	-	2	-	-	-	-	-	-	-	1
CO4	2	1	2	-	-	-	-	-	-	-	1
CO5	1	1	2	-	-	-	-	-	-	-	1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	-
CO2	3	-
CO3	2	-
CO4	2	-
CO5	1	-

UNIT – I

Programming:
Programming: Introduction to Programming Concepts with Scratch Introduction: Introduction to Python, Program Development Cycle, Input, Processing, and Output, Displaying Output with the Print Function, Comments, Variables, Reading Input from the Keyboard, Performing Calculations, Operators. Type conversions, Expressions, More about Data Output. Data Types Expression: Strings Assignment, and Comment, Numeric Data Types and Character Sets, Using functions and Modules. Decision Structures and Boolean Logic: if, if-else, if-elif-else Statements, Nested Decision Structures, Comparing Strings, Logical Operators, Boolean Variables. Repetition Structures: Introduction, while loop, for loop, Calculating a Running Total, Input Validation Loops, Nested Loops.

Practice:
1. Basic Programs.
   a. Convert weight from kilograms to pounds.
   b. Input three numbers; compute total and average.
   c. Use a for loop to print 8, 11, 14, ..., 89.

d. Accept user name and repeat it n times.
e. Print a triangle of height n using *.

2. Decision Structures and Loops.
   a. Generate a random number (1-10), and ask the user to guess it.
   b. Accept two float values; print "Close" if difference ≤ 0.001, else "Not Close".
   c. Check if a string contains any vowels.
   d. Input a large number and insert commas (e.g., 1000000 → 1,000,000).
   e. Insert * in expressions like 3x+4y to get 3*x+4*y.

UNIT – II

Control Statements:
Definite iteration for Loop Formatting Text for output, Selection if and if else Statement Conditional Iteration the While Loop Strings and Text Files: Accessing Character and Substring in Strings, Data Encryption, Strings and Number Systems, String Methods Text Files.

Practice:
1. Conditional and Looping Constructs
   a. Check if the year is a leap year or not.
   b. Simulate a basic calculator using conditionals.
   c. Display electricity bill based on consumption slabs.
   d. Check if a number is prime and print all primes in a range.
   e. Check for Armstrong number, Palindrome.
2. Loop-based Patterns and String Logic
   a. Print a number pyramid.
   b. Accept two strings of same length and interleave their characters.
   c. Count vowels, consonants, digits in a string.
   d. Convert strings: lowercase ↔ uppercase, toggle case.

UNIT – III

List and Dictionaries: List and Dictionaries: Lists, Defining Simple Functions, Dictionaries Design with Function: Functions as Abstraction Mechanisms, Problem Solving with Top-Down Design, Design with Recursive Functions, Case Study Gathering Information from a File System, Managing a Program's Namespace, Higher Order Function. Modules: Modules, Standard Modules, Packages.

Practice:
1. Working with Lists and Strings
   a. Generate 20 random numbers; print min, max, avg, even count.
   b. Generate a list of factors for a given integer.
   c. Generate 100 random 0s and 1s; find longest run of 0s.
   d. Remove duplicates from a list.
   e. Menu-based unit converter (feet ↔ inches, yards, etc.)
2. Matrix and String Processing
   a. Create and print matrices; add, multiply two matrices.
   b. Reverse, concatenate, and find length of strings.
   c. Count digit frequency in a string.
   d. Print each word in a sentence on a new line.
   e. Lower Triangular Matrix.

UNIT – IV

Object Oriented Programming: Concept of class, object and instances, Constructor, class attributes and destructors, Real time use of class in live projects, Inheritance , overlapping and overloading operators, Adding and retrieving dynamic attributes of classes, Programming using Oops support Design with Classes: Objects and Classes, Data modeling Examples, Case Study An ATM, Structuring Classes with Inheritance and Polymorphism File Operations: Reading config files in python, Writing log files in python, Understanding read functions, read(), readline() and readlines(), Understanding write functions, write() and writelines(), Manipulating file pointer using seek, Programming using file operations.

Practice:

1. OOP – Classes and Methods
    a. Create a Product class with dynamic pricing and stock update.
    b. Create a Time class to convert seconds → minutes, hours.
    c. Define Converter class for unit conversions (e.g., inches → feet).
2. File Handling
    a. Read email addresses from file and print as semicolon-separated list.
    b. Read temperatures from a file and convert them to Fahrenheit.
    c. Count character frequencies in a file.
    d. Count how many times each word appears in a file.
3. Additional OOP
    a. Implement a class for pow(x, n).
    b. Define a class to reverse a string word-by-word.

UNIT – V

Errors and Exceptions: Syntax Errors, Exceptions, Handling Exceptions, Raising Exceptions, User- defined Exceptions, Defining Clean-up Actions, Redefined Clean up Actions. Graphical User Interfaces: The Behavior of Terminal Based Programs and GUI - Based, Programs, Coding Simple GUI-Based Programs, Other Useful GUI Resources.

Practice:

1. GUI and Exceptions
    a. Build a file-open dialog and display content in a text box.
    b. Demonstrate try/except/else blocks.
    c. Show file handling using try/finally and with statements.
2. Integrated Problem Solving
    a. Implement the Hangman game using string logic and loops.
    b. Recursively reduce a number to a single-digit sum (e.g., 4683 → 4+6+8+3=21 → 2+1=3).
    c. Validate recursively whether an input is a well-formatted list.

Additional Practice:

1. Hangman **The Goal:** Despite the name, the actual “hangman” part isn’t necessary. The main goal here is to create a sort of “guess the word” game. The user needs to be able to input letter guesses. A limit should also be set on how many guesses they can use. This means you’ll need a way to grab a word to use for guessing. (This can be grabbed from a pre-made list. No need to get too fancy.) You will also need functions to check if the user has actually inputted a single letter, to

check if the inputted letter is in the hidden word (and if it is, how many times it appears), to print letters, and a counter variable to limit guesses.

2. Write a program to find the greatest number that can be formed by using given set of numbers.
3. Write a program to find sum of digits of a number till you get single digit sum.

Example:
Input : 142 (Hint: 1+4+2=7)
Output : 7
Input : 4683 (Hint: 4+6+8+3=21 => 2+1=3)
Output : 3

Text Books:

1 Fundamentals of Python First Programs, Kenneth. A. Lambert, Cengage, 2018
2 Beginning Python: from Novice to Professional, Lie Hetland, Magnus, 2nd Edition, 2005

Reference Books:

1. Introduction to Python Programming, Gowrishankar.S, Veena A, CRC Press.
2. Introduction to Programming Using Python, Y. Daniel Liang, Pearson, 2017.
3. Think Python, Allen Downey, Green Tea Press,2012.
4. Python for Everybody Exploring Data in Python 3, Charles Russell Severance, Sue Blumenberg.

Web Links:

1 https://www.python.org/
2 https://www.coursera.org/courses?query=Python%20programming
3 https://www.learnpython.org/
4 https://onlinecourses.nptel.ac.in/noc21_cs32/preview/

Computer Organization & Architecture
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Semester:				
Course Code: 2501CS71	**2**	**1**	**0**	**3**

Course Outcomes:
At the end of the course, student will be able to:

- **CO1:** Describe the basic structure of a computer system, various number systems and arithmetic operations.
- **CO2:** Explain the Operation of CPUs including RTL, ALU, Instruction Cycle and Buses
- **CO3:** Demonstrate the architecture and functionality of central processing unit
- **CO4:** Illustrate the I/O and memory organization in an efficient way.
- **CO5:** Make use of multi processors and pipelining to improve the efficiency of computer system.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	1	-	-	2	-	-	-	-	-	-
CO2	2	1	-	-	2	-	-	-	-	-	-
CO3	3	1	-	-	2	-	-	-	-	-	-
CO4	2	1	-	-	2	-	-	-	-	-	-
CO5	3	2	-	-	3	-	-	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO / PSO	PSO	PSO 2
CO1	2	
CO2	2	
CO3	2	
CO4	2	
CO5	3	

UNIT – I

Basic Structure of Computers: Basic Organization of Computers, Historical Perspective, Bus Structures. Data Representation: Data types, Complements, Fixed Point Representation. Floating – Point Representation. Other Binary Codes, Error Detection Codes. Computer **Arithmetic:** Addition and Subtraction, Multiplication Algorithms, Division Algorithms.

UNIT – II

Register Transfer Language and Microoperations: Register Transfer language. Register Transfer Bus and Memory Transfers, Arithmetic Micro operations, Logic Micro Operations, Shift Micro Operations, Arithmetic Logic Shift Unit

Basic Computer Organization and Design: Instruction Codes, Computer Register, Computer Instructions, Instruction Cycle, Memory – Reference Instructions. Input – Output and Interrupt, Complete Computer Description.

UNIT – III

Central Processing Unit: General Register Organization, STACK Organization. Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer.
Microprogrammed Control: Control Memory, Address Sequencing, Micro Program example, Design of Control Unit.

UNIT – IV

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary Memory, Associative Memory, Cache Memory, Virtual Memory.
Input-Output Organization: Peripheral Devices, Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupts, Direct Memory Access.

UNIT – V

Multi Processors: Introduction, Characteristics of Multiprocessors, Interconnection Structures, Inter Processor Arbitration.
Pipeline: Parallel Processing, Pipelining, Instruction Pipeline, RISC Pipeline, Array Processor.

Text Books:

1. Computer System Architecture, M. Morris Mano, Third Edition, Pearson, 2008.
2. Computer Organization, Carl Hamacher, ZvonkoVranesic, SafwatZaky, 5/e, McGraw Hill, 2002.

Reference Books:

1. Computer Organization and Architecture, William Stallings, 6/e, Pearson, 2006.
2. Structured Computer Organization, Andrew S. Tanenbaum, 4/e, Pearson, 2005.
3. Fundamentals of Computer Organization and Design, Sivarama P. Dandamudi, Springer, 2006

Web Links:

1. https://nptel.ac.in/courses/106/105/106105163/
2. https://nptel.ac.in/courses/106/106/106106092/
3. https://www.udemy.com/course/computer-architecture-computer-organization- course/
4. http://www.cuc.ucc.ie/CS1101/David%20Tarnoff.pdf

Discrete Mathematics
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501MA08	**2**	**1**	**0**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Apply the principles of mathematical logic to statement calculus and predicate calculus.
CO2: Compute transitive closure, equivalence classes of binary relations.
CO3: Apply basic counting techniques to solve the combinatorial problems.
CO4: Solve recurrence relations by various methods.
CO5: Apply the concepts of Graph theory to find Euler paths, Hamiltonian paths and demonstrate different traversal methods for trees.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2									
CO2	3	2									
CO3	2	3									
CO4	3	2									
CO5	3	2									

UNIT – I

Logic: Propositional Logic: Statements and logical connectives, Well-Formed Formulas, Truth Tables, Tautologies, Equivalence of Formulas, Duality Law, Tautological Implications, Normal Forms, Theory of Inference for Statement Calculus. Predicate Logic: Statement Functions, Variables, and Quantifiers; Inference Theory for Predicate Calculus.
Practice(Using any computational tool): Construct the truth table calculator taking two inputs, P and Q, and calculate AND, OR and NOT.

UNIT – II

Relations: Properties of Binary Relations, Relation Matrix, Digraph, Operations on Relations, Partition and Covering, Transitive, Closure, Warshall Algorithm, Equivalence Relation, R-Equivalence Class, partially ordered sets, Hasse Diagrams.

UNIT – III

Combinatorics: Basis of Counting, Permutations, Permutations with Repetitions, Circular and Restricted Permutations, Combinations, Restricted Combinations, Binomial and Multinomial Coefficients and Theorems.

UNIT – IV

Recurrence Relations: Formation of Recurrence Relations, Solving Recurrence Relations by Substitution and Generating Functions, Method of Characteristic Roots.
Practice(Using any computational tool: Compute the nth Fibonacci number.

UNIT – V

Graph Theory: Basic Concepts of Graphs, Matrix Representation of Graphs: Adjacency Matrix, Incidence Matrix, Isomorphic Graphs, Paths and Circuits, Euler and Hamilton Graphs, Planar Graphs, and Euler's Formula. **Trees:** Tree Properties, Spanning Trees, BFS Algorithm, DFS Algorithm, Minimal Spanning Trees, and Kruskal's Algorithm, Graph coloring, chromatic number.
Practice(Using any computational tool): BFS and DFS algorithms.

Text Books:

1. Discrete Mathematics and its Applications with Combinatorics and Graph Theory, Kenneth H. Rosen, McGraw Hill Education (India) Private Limited, ISBN : 9780070681880**.**
2. Discrete Mathematical Structures with Applications to Computer Science, J. P. Tremblay and R. Manohar, Tata Mc Graw Hill, ISBN 978-0074631133

Reference Books:

1. Discrete Mathematical Structures, Bernand Kolman, Robert C. Busby and Sharon Cutler Ross, PHI. ISBN 978-9332549593.
2. Mathematical Foundations of Computer Science, S. Santha, E. V. Prasad, Cengage Publishers, ISBN 978-8131515938.

Web Links:

1. http://nptel.ac.in/courses/106106094/
2. http://mathworld.wolfram.com/classroom/classes/DiscreteMathematics.html
3. http://mathworld.wolfram.com/topics/GeneralLogic.html
4. https://ggc-discrete-math.github.io/

Object Oriented Programming through C++
(Common to CSE, IT, AIML & CSE(DS))

	L	T	P	C
Course Code: 2501CS08	**2**	**0**	**2**	**4**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Make use of C and C++ programming constructs to demonstrate OOP concepts.
CO2: Develop applications using constructor and destructor.
CO3: Apply C++ features for problem solving.
CO4: Apply inheritance concepts and file I/O to solve a given problem.
CO5: Design C++ classes using templates and STL.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	1	2					1	1		
CO2	1	2	2					1	1		
CO3	1	1	1					1	1		
CO4	1	1	2					1	1		
CO5	2	1	2					1	1		

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	
CO2	2	
CO3	2	
CO4	2	
CO5	2	

UNIT – I

Introduction to C++: Differences between C and C++, Disadvantages of Conventional Programming, Evolution of C++, Key Concepts of Object Oriented Programming, Structure of C++ program , Advantages of OOP, scope resolution Operator, Namespace.

Practice:

1. Find the roots of a quadratic equation.
2. Find factorial of a given number using recursion
3. Implement scope resolution and namespaces.
4. Illustrate the use of default arguments and access specifiers

UNIT – II

Classes and Objects: Declaring Objects- Access Specifiers and their Scope,Defining Member Function, Rules for Inline Functions, static Member Variable, static Member Function, Friend functions, overloading Member Function, Array of Objects, Object as Parameter, Object as return type, Nestedclass, Introduction to Constructors and Destructor,Characteristics of Constructor and Destructor, Application with Constructor, Constructor with Arguments, Copy Constructor, Anonymous Objects.

Practice:

1. Program to illustrate inline functions and function overloading.
2. Program to illustrate friend function
3. Program to illustrate the use of Constructors and Destructors.
4. Program illustrating Constructor overloading.
5. Program illustrating Copy Constructor.

UNIT – III

Operator Overloading: Introduction, The Keyword Operator,Rules for Overloading Operators,Overloading Unary Operator,Overloading Binary Operator, Overloading using friend function.
Inheritance: Introduction, Reusability, Types of Inheritance, Virtual Base Classes, Object as a Class Member, Abstract Classes, Advantages and Disadvantages of Inheritance. Pointer to Class- Pointer Object, this Pointer, Pointer to Derived Classes and Base Class.
Binding Polymorphisms and Virtual Functions: Introduction, Binding in C++,Virtual Functions, Rules for Virtual Function, Abstract classes, Virtual Destructor

Practice:

1. Overload Unary, and Binary Operators using member function.
2. Overload Unary, and Binary Operators using friend function.
3. Incorporate various forms of Inheritance i. Single Inheritance ii. Multiple Inheritances iii. Multi-level inheritance iv. Hierarchical Inheritance v. Hybrid inheritance
4. Order of execution of constructors and destructors in inheritance.
5. Illustrate the use of object as a class member,pointer to a class,this pointer and Virtual Base Class.
6. Illustrate virtual functions.
7. Implement pure virtual function and calculate the area of different shapes by using abstract class.

UNIT – IV

Generic Programming with Templates: Introduction, Need for Templates,Definition of class Templates, Normal Function Templates, Overloading of Template Function, Bubble Sort Using Function Templates, Difference between Templates and Macros, Linked list with templates

Practice:

1. Illustrate the use of function template.
2. Implement template class.

3. Implement class templates with multiple parameters.

UNIT – V

Exception Handling: Introduction, Principles of Exception Handling, The Keywords try throw and catch, Multiple Catch Statements.
Files: Stream I/O ,Reading and writing to Files.
Overview of Standard Template Library: Introduction, STL Programming Model, Containers, Sequence Containers, Associative Containers, Algorithms, Iterators, Vectors, Lists, Maps

Practice:

1. Implement Exceptions handling.
2. Illustrate the use of multiple catch statements.
3. Implement List,Vector and its Operations.
4. Implement Deque and its Operations.
5. Implement Map and its Operations.

Additional Practice:

1. Develop a C++ program for flight booking system
2. Develop Qt application containing slider and spin box in which a slider responds to changes in the spin box.
3. Develop a Qt application for creating a text pad.
4. Develop a C++ program with maximum of 20 characters, that your user will be guessed and will show only asterisks (*) on the screen. The user will input or enter one character at a time. And for every correct character, the asterisk will be replaced by that character until all the characters or the mystery word/s will reveal. Your program will accept a maximum three (3) errors or mistakes in entering/inputting character otherwise the mystery word/s will be viewed. Sample Output:

Output: ******* Enter your character: e
Output: ***e**e
Enter your character: a Output: sorry! the character is not existing. you still have 2 chances
Enter your character: s Output: s**e**e
Enter your character: c Output: sc*e*ce
Enter your character: i Output: scie*ce
Enter your character: n Output: science

Text Books:

1 The Complete Reference C++, Herbert Schildt, Fourth Edition, TMH ISBN-13:978-0070532465, and the ISBN-10 is 007053246X.
2 The C++ Programming Language, Bjarne Stroustrup, Fourth Edition, Pearson ISBN 978-0321563842

Reference Books:

1 A First Book Of C++, Gary Bronson, Fourth Edition, Cengage Learning ISBN-10. 8131532712 and ISBN-13. 978-8131532713
2 C++ Primer Plus By Stephen Prata, Sixth Edition, Pearson ISBN: 978-0321776402

3 C++ Programming:From Problem Analysis To Program Design, Ds Malik,Eighth Edition, Cengage Learning ISBN: 9781337102087

Web Links:

1 http://nptel.ac.in/courses/106105151/
2 http://www.geeksforgeeks.org/c-plus-plus/
3 https://infyspringboard.onwingspan.com/web/en/viewer/html/lex_auth_013501572848615424101 01
4 https://www.udemy.com/course/beginning-c-plus-plus-programming/

Theory of Computation
(Common to CSE&IT)

	L	T	P	C
Course Code: 2501CS11	**2**	**1**	**0**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Apply the properties of languages and automata to solve a given problem.
CO2: Construct an automata for a given language.
CO3: Design grammars for a given language.
CO4: Apply interconversion on automata, grammar and regular expressions.
CO5: Analyze decidability and undecidability problems.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3		1								
CO2	1	2	1								
CO3	1	2	1								
CO4	3										
CO5	2	2		2							

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	
CO2	2	
CO3	2	
CO4	2	
CO5	2	

UNIT – I

Finite Automata: Need of Automata theory, Central Concepts of Automata Theory, Automation, Finite Automation, Transition Systems, Acceptance of a String, DFA, Design of DFAs, NFA, Design of NFA, Equivalence of DFA and NFA, Conversion of NFA into DFA, Finite Automata with Є-Transitions, Minimization of Finite Automata, Finite Automata with output-Mealy and Moore Machines, Applications and Limitation of Finite Automata.

UNIT – II

Regular Expressions, Regular Sets, Identity Rules, Equivalence of two RE, Manipulations of REs, Finite Automata and Regular Expressions, Inter Conversion, Equivalence between FA and RE, Pumping Lemma of Regular Sets, Closure Properties of Regular Sets, Grammars,

Classification of Grammars, Chomsky Hierarchy Theorem, Right and Left Linear Regular Grammars, Equivalence between RG and FA, Inter Conversion.

UNIT – III

Formal Languages, Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars-Elimination of Useless Symbols, Є-Productions and Unit Productions, Normal Forms-Chomsky Normal form and Greibach Normal Form, Pumping Lemma, Closure Properties, Applications of Context Free Grammars.

UNIT – IV

Pushdown Automata, Definition, Model, Graphical Notation, Instantaneous Description, Language Acceptance of Pushdown Automata, Design of Pushdown Automata, Deterministic and Non – Deterministic Pushdown Automata, Equivalence of Pushdown Automata and Context Free Grammars, Conversion, Two Stack Pushdown Automata, Application of Pushdown Automata.

UNIT – V

Turing Machine: Definition, Model, Representation of TMs-Instantaneous Descriptions, Transition Tables and Transition Diagrams, Language of a TM, Design of TMs, Types of TMs, Church's Thesis, Universal and Restricted TM, Decidable and Un-decidable Problems, Halting Problem of TMs, Post's Correspondence Problem, Modified PCP, Classes of P and NP, NP-Hard and NP-Complete Problems.

Text Books:

1. Introduction to Automata Theory, Languages and Computation, J. E. Hopcroft, R. Motwani and J. D. Ullman, 3rd Edition, Pearson, 2008 ISBN: 978-8131720479
2. Theory of Computer Science-Automata, Languages and Computation, K. L. P. Mishra and N. Chandrasekharan, 3rd Edition, PHI, 2007 ISBN: 978-8120329683

Reference Books:

1. Elements of Theory of Computation, Lewis H.P. & Papadimitriou C.H., Pearson /PHI ISBN: 978-0132624787
2. Theory of Automata, Languages and Computation, Rajendra Kumar, McGraw Hill, 2014 ISBN: 978-0070702042

Web Links:

1. http://nptel.ac.in/courses/111103016
2. https://archive.nptel.ac.in/courses/106/104/106104148/
3. https://www.iitg.ac.in/dgoswami/Flat-Notes.pdf
4. https://www.ics.uci.edu/~goodrich/teach/cs162/notes/
5. https://www.geeksforgeeks.org/introduction-of-finite-automata/

Database Management Systems
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501IT05	**2**	**0**	**2**	**4**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Describe the Fundamental concepts of DBMS.
CO2: Interpret relational database using SQL.
CO3: Make use of normalization techniques for database design.
CO4: Illustrate the mechanisms of transaction management.
CO5: Optimize database performance with advanced indexing, query optimization, and robust backup and recovery strategies.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	3	3		1			1			
CO2	3	3	3	2	2	1		1			
CO3	3	3	3	3	3	2					1
CO4	3	3	3	1	2	1					
CO5	3	3	3	3	3	2					2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	3	3
CO2	3	2
CO3	3	2
CO4	3	1
CO5	3	2

UNIT – I

Introduction to DBMS: History and Architecture, Data Independence, Data Models, Levels of abstraction, structure of DBMS.
Data Models : Concepts of Schema, Instance and data independence,Three tier schema architecture for data independence, Database system structure, environment, Centralized and Client Server architecture for the database**.**

Practice
1. Familiarization with installation of any DBMS.
2. Implementing a University Database System.

UNIT – II

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance
BASIC SQL: Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update), basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions(Date and Time, Numeric, String conversion).

Practice
1. Querying and modifying the database using Data Manipulation Language commands - select, insert, update, delete.
2. Implementation of Aggregate Functions – sum, avg, min, max, count. Use group-by and having clause.

UNIT – III

Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams.
SQL: Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view(updatable and nonupdatable), relational set operations.

Practice
1. Perform join operations – natural join, equi-join, outer join, left outer join, right outer join, inner join and assess the impact of query plans on the performance of join heavy queries.
2. Perform set operations - union, intersection, set difference.
3. Implementation of corelated sub-queries and nested queries
4. Creating and querying views and materialized views.

UNIT – IV

Normalization: Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency (1NF, 2NF and 3 NF), concept of surrogate key, Boyce-codd normal form(BCNF), Lossless join and dependency preserving decomposition, Fourth normal form(4NF), Fifth Normal Form (5NF).
Transaction Concept: Transaction State, Implementation of Atomicity and Durability, Concurrent Executions, Serializability, Recoverability, Storage, Recovery and Atomicity, Recovery algorithm.

Practice
1. Implement SQL queries on a normalized database schema based on the provided schema.
For this exampe: use the schema for a university database, which includes:
- Students (StudentID, StudentName, Major)
- Courses (CourseID, CourseName, Credits)
- Enrollments (StudentID, CourseID, EnrollmentDate)

- Instructors (InstructorID, InstructorName, Phone)
- Course_Instructors (CourseID, InstructorID)

2. (A) Implementation of Data Control Language commands – grant and revoke.
   (B) Implementation of Transaction Control Language commands - commit, save point, and rollback.

UNIT – V

Indexing Techniques: Introduction**,** B+ Trees: Search, Insert, delete algorithms, File Organization and Indexing, Cluster Indexes, Primary and Secondary Indexes, Index data Structures, Hash Based Indexing: Tree base Indexing, Comparison of File Organizations.
Database Tuning: Introduction, Database Performance Tuning, Query Optimization: Introduction, Query Optimization algorithms, Backup and Recovery.

Practice

1. Create a Primary, Secondary Index on a Column?
2. Retrieve Data Using an Index?
3. Insert Data and Update Indexes?
4. Delete Data and Impact on Indexes?

Note: The student must Complete & Submit a Database Fundamentals Certificate Course offered by Oracle Academy at the end of the Practice Session.

Text Books:

1. Database System Concepts ,Abraham Silberschatz, Henry Korth, and S. Sudarshan, McGraw-Hill , 8th Edition., ISBN-13**:** 978-1260230508.
2. Database Management Systems, Raghurama Krishnan, Johannes Gehrke, Mc Graw-Hill, 4th Edition, ISBN: 978-1260091195.

Reference Books:

1. Fundamentals of Database Systems ,R Elmasri, S Navathe, Addison-Wesley.6th edition, ISBN: 978-0136086208..
2. Database Management Systems ,Ramakrishna Gehrke,Mcgraw Hill Publication,3rd Edition, ISBN: 978-0072465631 .
3. Introduction to Database Systems, C J Date, Pearson, 9th Edition, ISBN: 978-0133970777.

Web Links:

1. https://academy.oracle.com/pages/coursedescription/Oracle%20Academy%20Database%20Foundations%20Course%20Description.pdf
2. https://www.w3schools.com/sql/
3. https://onlinecourses.nptel.ac.in/noc22_cs91/

Engineering Workshop

(Common to CE,EEE,ME,ECE,CSE,IT,AIML,CSE(DS),PT&Min.E)

	L	T	P	C
Course Code: 2501ME03	**0**	**0**	**1**	**1**

Course Outcomes:

At the end of the course, student will be able to:

CO1: Prepare various wooden joints.
CO2: Demonstrate various sheet metal models.
CO3: Develop the basic knowledge of house wiring.
CO4: Develop the basic knowledge of plumbing.
CO5: Practice various welded joints.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	1	1				1	1	2	1		1
CO2	1	1				1	1	2	1		1
CO3	2	2				1	1	2	1		1
CO4	1	2				1	1	2	1		1
CO5	2	2				1	1	2	1		1

Practice:

1. To make a T-Lap joint from the given wooden workpieces.
2. To make a dovetail lap joint from the given wooden workpieces.
3. To make a taper tray using the given sheet metal.
4. To make a funnel using the given sheet metal.
5. To make a square tin using the given sheet metal.
6. To connect three bulbs by using series and parallel connections.
7. To give connection to a bulb by using staircase wiring.
8. To prepare wiring for a tube light with switch control.
9. To prepare a PVC pipe joint by using the given circuit1.
10. To prepare a PVC pipe joint by using the given circuit2.
11. To make a butt joint using the given M.S pieces by arc welding.
12. To make a lap joint using given M.S pieces by arc welding

Additional Practice:

1. To make a cross-lap joint from the given wooden workpieces.
2. To make an open scoop using the given sheet metal.
3. To make a T-joint using given M.S pieces by arc welding.

Text Books:

1. Work shop Manual , P.Kannaiah & K.L.Narayana/ SciTech Publishers, 2nd edition, ISBN: 978-8183711302.
2. Elements of Workshop Technology, Vol I by S.K. Hajra Choudhury, S.K. Hajra Choudhury & Nirjhar Roy, Media Promoters and Publishers Pvt. Limited, 14th edition, ISBN: 8185099146.

Reference Books:

1. Workshop Technology, Part 1, W.A.J. Chapman, 5th edition, ISBN 9780415503020.
2. Engineering Practices Lab Manual, T. Jeyapoovan & M. Saravanapandian, Vikas Publishing House Pvt. Limited, 4th edition, ISBN: .8125929037
3. Engineering Practices Laboratory Manual, Ramesh Babu.V., VRB Publishers Private Limited, Chennai, Revised edition, 2019-20, ISBN: 978-81-935993-8.

Web Links:

1. https://bharatskills.gov.in
2. https://www.gopracticals.com/basic-engineering/workshop/

Agile Software Engineering
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501IT07	**2**		**1**	**3**

Course Outcomes:
At the end of the Course, Student will be able to:

CO1: Apply software engineering practices and evaluate different software development models to design and manage the life cycle of a Student Management System (SMS).
CO2: Analyze the principles of Agile methodologies and compare them with traditional models by transitioning a retail software company from Waterfall to Agile.
CO3: Evaluate the impact of Agile processes on requirements engineering and manage unstable requirements for a social media platform project.
CO4: Design and implement a comprehensive testing strategy using TDD and test automation tools in an Agile project.
CO5: Apply Agile design principles and execute refactoring techniques to develop maintainable and scalable software systems.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	1			2						
CO2	3	2	1	1				2	2	2	1
CO3	2	1	1							2	1
CO4	3	2	2	1				2	2	2	1
CO5	1	2									

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	1	
CO2	2	
CO3	1	1
CO4	2	2
CO5	1	1

UNIT – I

Introduction to Software Engineering: Define Software Engineering, The Software Process, Software Engineering Practice, Software Myths
Software Life Cycle models and Processes: Software Process, Process Classification, Phased Development Life Cycle, Software Development Process Models – The Waterfall, Spiral and Evolutionary Models, Software Project Management Process, Project Planning and Estimation, Gathering Requirements, Software Design, Software Testing, Software Quality and Reliability.
Case Study: Developing a Student Management System (SMS)

UNIT – II

Fundamentals of Agile Methodology: Theories for Agile Management, Agile Software Development, Traditional Model Vs. Agile Model, Classification of Agile Methods, Agile Manifesto and Principles, Agile Project Management, Agile Team Interactions, Ethics in Agile Teams, Agility in Design, Testing, Agile Documentations, Agile Drivers, Capabilities and Values.

Case Study: Implementing Agile Methodology in a Software Development Project

UNIT – III

Agile Processes: Lean production - SCRUM, Crystal, Feature Driven Development, Adaptive Software Development, and Extreme Programming: Method overview – lifecycle – work products, roles and practices.

Agility And Requirements Engineering: Impact of Agile Processes in RE, Current Agile Practices, Overview of RE Using Agile, Managing Unstable Requirements, Requirements Elicitation

Case Study: Agile Processes and Requirements Engineering in a Healthcare Software Project

UNIT – IV

Agile Testing: The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), xUnit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester.

Case Study: Implementing Test-Driven Development (TDD) for a Healthcare Application

UNIT – V

Agile Software Design and Development: Agile design practices, Role of design Principles including Single Responsibility Principle, Open Closed Principle, Liskov Substitution Principle, Interface Segregation Principles, Dependency Inversion Principle in Agile Design, Need and significance of Refactoring, Refactoring Techniques, Continuous Integration, Automated build tools, Version control.

Case Study: Refactoring an E-Commerce Backend Using Agile Design Principles.

Capstone Project:
Students shall choose a real-time software application (e.g., e-commerce backend, SMS, food delivery app) and apply Agile design principles, perform refactoring, implement TDD, and use version control and CI tools. This project should be implemented as a mini project and submitted at the end of the course.

Text Books:

1 Software Engineering – Concepts and Practices: Ugrasen Suman, Cengage Learning, ISBN: 9789390555475.
2 Agile Software Development, Current Research and Future Directions‖, Springer-Verlag Berlin Heidelberg, Dingsoyr, Torgeir, Dyba, Tore, Moe, Nils Brede (Eds.), ISBN-10. 3642125743.

Reference Books:

1 Software Engineering, Lan Sommerville, Ninth Edition, Pearson
2 Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results‖, Prentice Hall, David J. Anderson; Eli Schragenheim,

Web Links:

1 https://www.tutorialspoint.com/software_engineering/
2 https://www.sciencedirect.com/science/article/pii/S0950584916303329

Probability & Statistics
(Common to CSE, IT AIML & CSE (DS))

	L	T	P	C
Course Code: 2501MA09	**2**	**1**	**0**	**3**

Course Outcomes:
At the end of the course, student will be able to:

- **CO1:** Apply Baye's theorem
- **CO2:** Apply various Probability distributions for both discrete and continuous random variables
- **CO3:** Compute mean and variance of sample means with replacement and without replacement and maximum errors.
- **CO4:** Apply various tests to test the hypothesis concerning mean, Proportion,variance and attributes.
- **CO5:** Apply the concepts of correlation and regression to the given statistical data.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	3									
CO2	3	2									
CO3	2	3									
CO4	3	2									
CO5	3	2									

UNIT – I

Probability and Random Variables:
Probability – Conditional probability and Baye's Theorem-Random variables – Discrete and Continuous random variables-Mathematical Expectation, Variance and Moment generating function.
Practice(Using any computational tool):
Calculation of Mean and Variance.

UNIT – II

Probability Distributions:
Distribution function –Binomial, Poisson, Normal and Exponential distributions.
Practice(Using any computational tool):
Discrete and continuous probability distributions (Binomial, Poisson, normal and exponential distributions)

UNIT – III

Data Analytics and Sampling Theory:

Data science –Collection of data – primary and secondary data – Types of variables: dependent and independent Categorical and Continuous variables – Data visualization – Measures of Central tendency(Mean,mode and median) – Measures of Variability (spread or variance).
Sampling Theory: Introduction – Population and samples – Sampling distribution of Means and Variance – Central limit theorem (without proof)-Point and Interval estimations – Maximum error of estimate.

Practice(Using any computational tool):
1. Finding measures of variability and central tendency (mean, median, mode, arithmetic mean, geometric mean, Variance)
2. Data visualization (Construction of bar charts, pie charts).
3. Construction of confidence intervals

UNIT – IV

Tests of Hypothesis:
Introduction – Hypothesis – Null and Alternative Hypothesis – Type I and Type II errors – Level of significance – One tail and two-tail tests – Tests concerning one mean and two means (Large and Small samples) – Tests on proportions, Chi-Square and F test.

Practice(Using any computational tool):
1. Z-test, t-test and Chi-square test

UNIT – V

Correlation and Regression:
Method of least squares – Straight line - nonlinear curves– parabola -Exponential – Power curves-Correlation – Karl Pearson's correlation coefficient – rank correlation – regression-– regression coefficients and properties (without proof) –regression lines

Practice(Using any computational tool):
1. Finding correlation coefficient and regression lines

*Students are advised to use any Programming language/computational tool/AI tool like Python,Excel, PowerBI ,Wolfrum Alpha, Symbolab, Mathway, etc., for the practice

Text Books:

1. Miller and Freund's, Probability and Statistics for Engineers, 9/e, Pearson, 2020. ISBN 978-9353945237
2. Fundamentals of Mathematical Statistics, S.C. Gupta and V.K.Kapoor, 12th edition, Sultan Chand & Sons Publications, 2020, ISBN: 978-9351611738

Reference Books:

1. Probability, Statistics and Random processes, T.Veerrajan, Tata McGraw-Hill, ISBN: 978-0070669253
2. Probability and statistics by T.K.V. Iyengar, S. Chand publishers,2022, ISBN: 9789355010643
3. Modern Statistics: Intuition, Math, Python, R by Mike X Cohen, ISBN: 9798867723736

Web Links:

1. https://onlinecourses.nptel.ac.in/noc21_ma74/preview
2. http://mathworld.wolfram.com/topics/ProbabilityandStatistics.html
3. https://www.khanacademy.org/math/statistics-probability
4. https://www.geeksforgeeks.org/statistics-with-python/
5. https://www.coursera.org/learn/inferential-statistical-analysis-python?specialization=statistics-with- python

Java Programming
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501IT06	**2**	**0**	**2**	**4**

Course Outcomes:
At the end of the course, student will be able to:

- **CO1** Apply Java features for problem solving.
- **CO2** Build applications using principles of OOPs, interfaces and Packages.
- **CO3** Develop programs using Exception Handling to handle run-time errors.
- **CO4** Develop applications using multithreading for inter thread communication.
- **CO5** Build JDBC applications for performing CRUD operations using MySQL.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	2	1	1		1				1	2	1
CO2	2	1	3		1				1	2	1
CO3	2	3	2		1				1	2	1
CO4	2	2	2		1				1	2	1
CO5	2	2	3		1				1	2	1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	1	
CO2	1	
CO3	1	
CO4	2	
CO5	3	

UNIT – I

Introduction to Java: History of Java, Features of Java Language, JVM Architecture, Java source file structure. Building Blocks of Java, Control Statements, Operators, Type Conversion, Command Line Arguments, Arrays.

Practice:
1. **Control Statements**
   a. Select all the prime numbers within the range of 1 to 100.
   b. Find the sum of all even terms in the Fibonacci sequence up to the given range N.
   c. Check whether a given number is Armstrong or not.

2. Arrays
   a. Sort an array of integers in ascending order.
   b. Find the maximum and minimum element in an array.

c. Remove duplicate elements from an array.

UNIT – II

Concepts of Class: Class, Object, Method, Constructor, Overloading- Method and Constructor Keywords: this, static and final, String Class and its Methods, StringTokenizer, StringBuffer and StringBuilder.

Practice:

1. Class Mechanism

a. Display the details of a person. Personal details should be given in one method and the qualification details in another method.
b. Implement constructor and constructor overloading.
c. Implement method overloading.

2.Strings

a. Check if a given string is a pangram (contains every letter of the alphabet at least once).
b. Find the most frequently occurring character in a string.
c. Find all permutations of a given string.
d. Check if a given string is a anagram (Ex: CAT and ACT).

UNIT – III

Inheritance: Inheritance, Types of Inheritance, super and super( ), Method Overriding, Using final with inheritance, abstract with Class and Method, Dynamic Method Dispatch.
Interfaces: Introduction to interface, Multiple inheritance.

Practice:

1. Inheritance & interface

a. Implement multi level Inheritance.
b. Implement multiple Inheritance.
c. Find the areas of different shapes using abstract classes.

UNIT – IV

Packages: Introduction, Importing Packages and Classes, path and classpath, Access Specifiers, java.util: Collection Framework (related interfaces and classes), java.io: FileReader, FileWriter, FileInputStream, FileOutputStream, BufferReader and BufferWriter, Wrapper Classes.

Practice:

1. Packages & java.io

a. Import and use user defined package.
b. Illustrate the use of protected members in a package.
c. Copy Even numbers into Even.txt file and Odd Numbers into Odd.txt file.

2. Collection Framework

a. ArrayList and LinkedList
b. Iterator and Iterable
c. Comparator and Comparable
d. HashMap and TreeMap
e. HashSet and TreeSet
f. HashTable

UNIT – V

Exception Handling: Introduction, Keywords: try, catch, throw, throws and finally, Multiple Catch Clauses, Checked and Unchecked Exceptions, User Defined Exceptions.

Multithreading: Introduction, Thread Life Cycle, Creation, Priorities, Synchronization, ITC.

Java Database Connectivity: Introduction, JDBC Architecture, Types of Drivers, JDBC Environment Setup, JDBC Steps, CRUD Operations.

Practice:

1. Exception Handling

a. Illustrate exception handling mechanism using multiple catch clauses.
b. Make use of Built-in and user-defined Exceptions in handling a run time exception.

2. Multithreading

a. Creates threads by extending Thread class. First thread display "Good Morning "every 1 sec, the second thread displays "Hello "every 2 seconds and the third display "Welcome" every 3 seconds, (Repeat the same by implementing Runnable).
b. Solve Producer-Consumer problem using synchronization.

3. JDBC

a. CRUD operations.

Additional Practice:

1. Student Management
2. Inventory Management
3. Phone book Management

Capstone Project:

Develop a console-based quiz application in Java where questions are fetched from a database. The app should handle user login, score calculation, timer using threads, exception handling, and store results in both files and database.

*** Note: The student must Complete & Submit a Java Programming Certificate Course offered by Oracle Academy at the end of the Practice Session.**

Text Books:

1. The Complete Reference Java, Herbert Schildt, TMH ,8th Edition, ISBN 978-1260440232.
2. Java one step ahead, Anita seth, B.L.Juneja, Oxford, First Edition. ISBN: 9780199459643.

Reference Books:

1. Introduction to java programming, by Y Daniel Liang, Seventh Edition, Pearson, ISBN: 978-8131729588
2. Core Java: An Integrated Approach, R.Nageswara Rao, Dream tech press, ISBN: 978-9351199250

Web Links:

1 https://nptel.ac.in/courses/106/105/106105191/
2 http://java.sun.com/docs/books/tutorial/

Operating Systems
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501CS13	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Describe basic concepts of Operating Systems and its structure.
CO2: Analyse various issues related to inter process communication like process scheduling, resource management and deadlocks.
CO3: Interpret the issues and challenges of memory management.
CO4: Illustrate concepts of Disk management and file system implementation.
CO5: Explain issues related to protection and security mechanisms.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3								1		
CO2	1	3		1					2		1
CO3	1	3		1					2		1
CO4	1	3		1					1		1
CO5	2	2							1		

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	**1**	
CO2	**2**	
CO3	**2**	
CO4	**2**	
CO5	**2**	

UNIT – I

Operating Systems Overview: Operating system functions, Operating system structure, Operating systems operations, Computing environments, Open-Source Operating Systems. System Structures: Operating System Services, User and Operating-System Interface, systems calls, Types of System Calls, system programs, operating system structure, System Boot.

Case Study: The above topics are discussed as case study in Windows/ Unix OS
Practice:
1. Basic Linux commands

2. Implementation of System calls
3. Multiprogramming-Memory management-Implementation of fork (), wait (), exec() and exit (), System calls

UNIT – II

Process Concept: Process scheduling, Operations on processes, Inter-process communication.
Process Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling, Thread scheduling.
Multithreaded Programming: Multithreading models, Thread libraries, Threading issues

Case Study: The above topics are discussed as case study in Windows/ Unix OS

Practice:
1. Simulate the following CPU scheduling algorithms:FCFS and SJF
2. Simulate the following CPU scheduling algorithms: Priority and Round Robin

UNIT – III

Inter-process Communication: Race conditions, Critical Regions, Mutual exclusion with busy waiting, Sleep and wakeup, Semaphores, Mutexes, Monitors, Message passing, Classical IPC Problems - Dining philosophers problem, Readers and writers problem.
Deadlocks: Resources, Conditions for resource deadlocks, Ostrich algorithm, Deadlock detection and recovery, Deadlock avoidance, Deadlock prevention.
Case Study: The above topics are discussed as case study in Windows/ Unix OS

Practice:
1. Simulate Bankers Algorithm for Dead Lock Avoidance
2. Write a C program that illustrates two processes communicating using shared memory.
3. Write C program to create a thread using pthreads library and let it run its function.
4. Write a C program to illustrate concurrent execution of threads using pthreads library

UNIT – IV

Memory-Management Strategies: Introduction, Swapping, Contiguous memory allocation, Paging, Segmentation.
Virtual Memory Management: Introduction, Demand paging, Copy on-write, Page replacement, Frame allocation, Thrashing, Memory-mapped files, Kernel memory allocation.
File Systems: Files, Directories, File system implementation, management and optimization.
Case Study: The above topics are discussed as case study in Windows/ Unix OS

Practice:
1. Simulate the Multiprogramming with a fixed number of tasks (MFT)
2. Simulate the Multiprogramming with a variable number of tasks (MVT)
3. Simulate the FIFO page replacement algorithm
4. Simulate the LRU page replacement algorithm

5. Simulate the following File allocation strategies: Sequenced, Indexed and Linked

UNIT – V

Secondary-Storage Structure: Overview of disk structure, and attachment, Disk scheduling, RAID structure, Stable storage implementation.
System Protection: Goals of protection, Principles and domain of protection, Access matrix, Access control, Revocation of access rights.
System Security: Introduction, Program threats, System and network threats.

Practice:

1. The above topics are discussed as case study in Windows/ Unix OS

Text Books:

1	Operating System Concepts, Abraham Silberschatz, Peter B Galvin and Greg Gagne, John Wiley and Sons Inc., 12th Edition. ISBN: 978-0470128725
2	William Stallings, Operating Systems: Internals and Design Principles. Prentice-Hall, 7th Edition.ISBN: 978-9332518803

Reference Books:

1	Modern Operating Systems, Tanenbaum A S, Pearson Education, 3rd Edition.ISBN: 978-0136006633
2	Operating Systems A Concept Based Approach, Dhamdhere D M, Tata McGraw-Hill, 3rd Edition.ISBN: 978-1259005589
3	Operating Systems, Nutt G,Pearson Educatio, 3rd Edition,. ISBN: 978-8131723593

Web Links:

1	https://archive.nptel.ac.in/courses/106/106/106106144/
2	https://archive.nptel.ac.in/courses/106/105/106105214/
3	https://www.coursera.org/learn/iot/lecture/MrgxS/lecture-3-1-operating-systems
4	https://in.udacity.com/auth?next=/course/introduction-to-operating-systems--ud923

Artificial Intelligence
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501AI02	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Illustrate the historical development of AI, key milestones and contributors.
CO2: Apply the basic principles of AI in problem solving
CO3: Apply Local Search Algorithms and Optimization Techniques.
CO4: Choose the appropriate representation of Knowledge.
CO5: Summarise the expert systems and real time applications

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3			1				1	1		1
CO2	3	2		2				1	1		1
CO3	3	2	1	2	1			1	1		1
CO4	2		1					1	1		
CO5	2	2	1	1	1			1	1		1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	1	
CO2	2	1
CO3	1	1
CO4	1	
CO5	1	1

UNIT – I

Introduction:
What Is AI? The Foundations of Artificial Intelligence, The History of Artificial Intelligence, The State of the Art, Agents and Environments, Good Behaviour: The Concept of Rationality, The Nature of Environments, The Structure of Agents.

Practice :
1. Introduction to LISP : To demonstrate simple recursive functions and list manipulation in LISP
2. Introduction to Prolog : Provide a brief introduction to Prolog syntax, including facts, rules, and queries.

UNIT – II

Problem formulation:
Problem Definition Production systems, Control strategies, Search strategies. Problem characteristics, ProblemSolving Agents, Example Problems, Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies

Practice :
1. Implementation of DFS for water jug problem using LISP/PROLOG
2. Implementation of BFS for tictactoe problem using LISP/PROLOG

UNIT – III

Problem solving methods
Problem graphs, Matching, Indexing and Heuristic functions Local Search Algorithms and Optimization Problems, Searching with Nondeterministic Actions, , Constraints satisfaction – Measure of performance and analysis of search algorithms. Game playing.

Practice :
1. Implementation of TSP using heuristic approach using LISP/Prolog
2. Implementation of Hillclimbing to solve 8 Puzzle Problem

UNIT – IV

Knowledge representation
Knowledge representation using Predicate logic, Introduction to predicate calculus, Resolution, Use of predicate calculus, Knowledge representation using other logicStructured representation of knowledge Basic plan generation systems – Strips Advanced plan generation systems – K strips Strategic explanations Why, why not and how explanations.

Practice :
1. Design a semantic network to represent knowledge about a specific domain, such as animals or transportation systems. Experiment with different node types, relationships, and attributes to encode information effectively. Implement a Python / LISP code for this program.
2. Create a framebased representation for a simple domain like a restaurant. Define frame templates for entities like menu items, customers, and orders, along with slots for attributes such as price, ingredients, and customer preferences.

UNIT – V

Expert Learning
Expert systems – Architecture of expert systems, Roles of expert systems – Knowledge Acquisition – Meta knowledge, Heuristics. Typical expert systems – MYCIN, DART, XOON, Expert systems shells.

Practice :
1. Using PYTHON / LISP , develop an expert system to help users plan travel itineraries, including transportation, accommodation, and activities. Utilize knowledge about travel destinations, transportation options, and user preferences to generate personalized travel plans.

2. Using PYTHON / LISP , create an expert system to assist students in selecting courses, majors, or career paths based on their interests, skills, and academic performance. Use knowledge about educational programs, career prospects, and academic requirements to make personalized recommendations.

Text Books:

1 Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig , Pearson, 3rd Edition, ISBN: 978-9332543515
2 Artificial Intelligence: Structures and Strategies for Complex Problem Solving by George F. Luger, 2nd Edition, ISBN: 978-0805347807

Reference Books:

1 Artificial Intelligence, Saroj Kaushik, Cengage Learning India, ISBN: 978-8131510995
2 Artificial Intelligence, Elaine Rich and Kevin Knight, Tata McGraw Hill, ISBN: 978-0-07-052263-3
3 Artificial Intelligence: Foundations for Computational Agents, David Poole and Alan Mackworth, Cambridge University Press, ISBN: 978-1107195394

Web Links:

1 https://nptel.ac.in/courses/106105077
2 https://nptel.ac.in/courses/106106126
3 https://aima.cs.berkeley.edu

Computer Networks
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501CS07	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Describe network topologies, reference models and media for data transmission
CO2: Analyze error and flow control issues in data link layer
CO3: Classify MAC protocols and channelization techniques
CO4: Apply routing algorithms and congestion control techniques for effective data transmission
CO5: Analyze protocols Transport and Application Layers

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	1									
CO2		2									
CO3		2							1	1	
CO4	1	2	2						2		1
CO5		1			1						1

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PO	PSO1	PSO2
CO1	1	
CO2	2	
CO3	2	
CO4	2	
CO5	2	

UNIT – I

Network Overview: interfaces, protocols and services, connection-oriented and connectionless services, OSI & TCP/IP Reference Models. Local Area Networks: Topologies - star, bus, ring, media access control - deterministic and probabilistic, IEEE 802.x. wireless networking.

Practice:
1. Study of Network devices in detail and connect the computers in Local Area Network

UNIT – II

Data Link Protocols: Framing and Data Transparency, Error Detection & Correction, Flow control.

Practice:
1. Write a Program to implement the data link layer farming methods such as i) Character stuffing ii) bit stuffing.

UNIT – III

Media Access Control: Random Access: ALOHA, Carrier sense multiple access (CSMA), CSMA with Collision Detection, CSMA with Collision Avoidance, Controlled Access: Reservation, Polling, Token Passing, Channelization: frequency division multiple Access (FDMA), time division multiple access (TDMA), code division multiple access (CDMA).

Practice:
1. Write a Program to implement Sliding window protocols

UNIT – IV

Routing Algorithms: The Optimality principle, Shortest path, Flooding, Distance vector, Link state, Hierarchical, Congestion Control algorithms: General principles of congestion control, Congestion prevention polices, Approaches to Congestion Control, Traffic Aware Routing, Admission Control, Traffic Throttling, Load Shedding, Traffic Control Algorithm: Leaky bucket & Token bucket.

Practice:
1. Programs to implements routing protocols like shortest path.

UNIT – V

IP Protocols: IP Addressing, IP & ICMP.
Transmission Control Protocol: UDP & TCP.
Application Layer: World Wide Web, HTTP, Electronic mail, Architecture, web based mail, email security, TELENET, local versus remote Logging, Domain Name System: Name Space, DNS, SNMP.

Practice:
1. NS2 Simulator i. NS2 Simulator-Introduction ii. Simulate to Find the Number of Packets Dropped iii. Simulate to Find the Number of Packets Dropped by TCP/UDP

Capstone Project:

Simulate and implement data link layer protocols and network routing strategies using NS2 (Network Simulator 2). It includes configuring network topologies, setting up routing protocols like AODV, DSDV, and OLSR, and analyzing their performance through metrics such as throughput, delay, and packet delivery ratio. Additionally, explore the simulation of data link layer functionalities, including MAC addressing and error detection. You may use NS2's capabilities, that insights into the behaviour of network protocols, aiding in the design and optimization of efficient communication systems.

Text Books:

1	Computer Networks — Andrew S Tanenbaum and David J Wetherall, Pearson Education, 5th Edition. ISBN: 978-0132126953
2	Data Communications and Networking – Behrouz A.Forouzan, McGraw Hill Education, 5th Edition ISBN: 978-1259064753

Reference Books:

1	Data Communications and Networks- Achut S Godbole, AtulKahate, McGraw-Hill Education (India) Pvt Limited, 2nd Edition. ISBN: 978-0-07-123110-7
2	Computer Networks, Mayank Dave, CENGAGE India, 6th Edition ISBN: 978-8131509869

Web Links:

1	https://nptel.ac.in/courses/106105081
2	https://www.coursera.org/learn/fundamentals-network-communications

Data Mining
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501AI03	**1**	**0**	**2**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Illustrate data warehousing architectures and mining concepts for knowledge discovery.
CO2: Apply preprocessing techniques to prepare data for mining algorithms.
CO3: Build classification model using Decision tree induction.
CO4: Analyze association rule generation using Apriori and FP growth algorithms.
CO5: Apply Clustering algorithms on given data to find similarity between classes.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	2	2		1			1	1		2
CO2	2	2	2		2			1	1		2
CO3	2	2	3		1			1	1		2
CO4	2	2	3		1			1	1		2
CO5	2	2	3		1			1	1		2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1		1
CO2		2
CO3		2
CO4		2
CO5		2

UNIT – I

Data Warehousing and Online Analytical Processing:
Data Warehouse: Basic Concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Data Generalization by Attribute-Oriented Induction.
Data Mining: What Is Data Mining, Why Data Mining, Data Mining Functionalities, Technologies Are Used, Major Issues in Data Mining, Architecture of Data Mining Systems Classification of Data Mining Systems.

UNIT – II

Know Your Data : Data Objects and Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity.
Data Preprocessing : An Overview, Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization

Practice:
1. Apply Statistical Descriptions and Measures of Similarity and Dissimilarity on given data
2. Apply PreProcessing for training data set related to Weather data

UNIT – III

Classification and Prediction: Classification: Basic Concepts, General Approach to Solving a Classification Problem, Decision Tree Induction, Model Overfitting, Evaluating the Performance of a Classifier, Issues Regarding Classification and Prediction

Practice:
1. Demonstrate classification rule process on given data set using id3 algorithm.
2. Demonstrate classification rule process on Student's Academic Performance Data using j48 algorithm.

UNIT – IV

Mining Frequent Patterns, Associations and Correlations: Basic Concepts and Methods, Frequent Itemset Mining Methods, Which Patterns Are Interesting?—Pattern Evaluation Methods, Association Mining to Correlation Analysis.

Practice:
1. Demonstrate Association rule generation on credit card data using apriori algorithm.
2. Demonstrate Association rule process onc ontactlenses.arff using FPGrowth Algorithm

UNIT – V

Cluster Analysis: Types of Data in Cluster Analysis – A Categorization of Major Clustering Methods – Partitioning Methods – Hierarchical methods – Evaluation of Clustering, Mining Spatial, Multimedia, Text and Web Data: Multidimensional Analysis and Descriptive Mining of Complex Data Objects – Spatial Data Mining – Multimedia Data Mining – Text Mining – Mining the World Wide Web.

Practice:
1. Demonstrate clustering process on given dataset using simple k-means
2. Apply Hierarchical clustering process on dataset

Additional Practice:
1. Write a program of cluster analysis using DB SCAN algorithm Python programming language
2. Demonstrate Web/Text Mining using WEKA/KNIME Tool

Capstone Project: Implement a data mining application involving data preprocessing and various data mining tasks can be achieved by focusing on a specific domain, such as retail, sales and customers, and then applying different techniques to analyse the data. This project would involve data cleaning, transformation, and various data mining tasks like clustering and association rule mining of your own data set, potentially leading to valuable insights into various area of data mining Applications.

Text Books:

1	Data Mining Concepts and Techniques, Jiawei Han, Micheline Kamber and Jian Pei Third Edition, Elsevier, ISBN: 978-9380931913
2	Introduction to Data Mining, PangNing Tan, Michael Steinbach and Vipin Kumar Pearson Education, ISBN: 978-1-292-02615-2

Reference Books:

1	Data Mining: Introductory and Advanced Topics, Margaret H. Dunham, 1st Edition Pearson Edn, ISBN: 978-8177587852
2	Data Mining Techniques and Applications An Introduction, Hongbo Du, 1st Edition, Cengage India, ISBN: 978-8131519554
3	Data Warehousing, Data Mining and OLAP, Alex Berson,.Stephen J. Smith, McGrawHill series, ISBN: 978-0070062726

Web Links:

1	https://nptel.ac.in/courses/106105174/
2	https://www.kdnuggets.com/
3	https://www.import.io/post/datascientistsvsdataanalystswhythedistinctionmatters/

Advanced Data Structures & Algorithm Analysis
(Common to AIML, CSE, CSE-DS & IT)

	L	T	P	C
Course Code: 2501CS10	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the Course, Student will be able to:

CO1: Analyze the algorithms efficiency and hashing techniques for searching.
CO2: Illustrate the concepts of search trees and operations of Priority Queues.
CO3: Analyze the algorithms of Divide & Conquer and Greedy paradigms.
CO4: Analyze the algorithms of Dynamic Programming paradigm.
CO5: Design the algorithms of Backtracking and Brach & Bound paradigms.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	-	-	-	-	-	-	-	-
CO2	2	2	1	-	-	-	-	-	-	-	-	-
CO3	1	2	1	2	-	-	-	-	-	-	-	-
CO4	2	2	2	1	-	-	-	-	-	-	-	-
CO5	2	1	1	2	-	-	-	-	-	-	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	-
CO2	2	-
CO3	2	-
CO4	2	-
CO5	2	-

UNIT – I

Introduction to Algorithm Analysis, Space and Time Complexity analysis, Asymptotic Notations.

Hashing: Introduction to Static Hashing, Hash Tables, Hash Functions, Different Hash Functions, Collision Resolution Techniques, Dynamic Hashing.

Practice:

1. Write a program to implement collision resolution techniques of Hash data structure.

UNIT – II

AVL Trees – Creation, Insertion, Deletion operations and Applications.
B-Trees – Creation, Insertion, Deletion operations and Applications
Priority Queues (Heaps): Introduction, Binary Heaps-Model and Simple Implementation, Basic Heap Operations, Other Heap Operations, Applications of Priority Queues

Practice:

1. Write a program to implement AVL tree operations
2. Write a program to implement Max heap and min heap operations

UNIT – III

Divide and Conquer: The General Method, Binary Search, Merge Sort, Quick Sort.

Greedy Method: The General Method, Job Sequencing with deadlines, Knapsack Problem, Minimum cost spanning trees, Single Source Shortest Paths.

Practice:

1. Write a program to sort the given list of elements using Merge sort technique using divide and conquer approach.
2. Write a program to implement Single Source Shortest Paths using greedy approach.

UNIT – IV

Dynamic Programming: The General Method, Single Source Shortest Paths– General Weights (Bellman Ford Algorithm), All-pairs shortest paths, Optimal Binary Search Trees, 0/1 Knapsack, Travelling Salesperson problem

Practice:

1. Write a program to implement 0/1 Knapsack problem using Dynamic Programming approach.
2. Write a program to implement all pairs shortest path problem using Dynamic Programming approach.

UNIT – V

Backtracking: The General Method, N-Queens Problem, Sum of Subsets problem, Graph Coloring, Hamiltonian cycles.

Branch and Bound: The General Method, 0/1 Knapsack Problem, Travelling Salesperson problem.

Practice:

1. Write a program to implement N-Queens problem using backtracking algorithm.
2. Write a program to implement Travelling Salesperson problem using branch and bound approach.

Text Books:

1. Fundamentals of Data Structures in C++, Horowitz, Ellis; Sahni, Sartaj; Mehta, Dinesh, 2nd Edition Universities Press.
2. Computer Algorithms in C++, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran,2nd Edition University Press.

Reference Books:

1. Data Structures and program design in C, Robert Kruse, 2nd Edition Pearson Education Asia.
2. An introduction to Data Structures with applications, Trembley& Sorenson, McGraw Hill
3. The Art of Computer Programming, Vol.1: Fundamental Algorithms, Donald E Knuth, Addison - Wesley.
4. Data Structures using C & C++: Langsam, Augenstein &Tanenbaum, Pearson

Web Links:

1. https://archive.nptel.ac.in/courses/106/102/106102064/
2. https://archive.nptel.ac.in/courses/106/106/106106131/
3. http://peterindia.net/Algorithms.html

Compiler Design
(Common to CSE&IT)

	L	T	P	C
Course Code: 2501IT04	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Analyze the phases in compilation process.
CO2: Apply different parsing algorithms to develop parsers for a given grammar
CO3: Construct various forms of intermediate code.
CO4: Compare the storage allocation strategies.
CO5: Apply techniques to generate optimized and target code.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	3							1	1	
CO2	2	3	1		1				1	1	
CO3	3	2							1	1	
CO4	2	2							1	1	
CO5	2	2	1						1	1	

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	
CO2	1	
CO3	1	
CO4	1	
CO5	1	

UNIT – I

Overview of Language Processing: Language Processors, Structure of a Compiler.
Lexical Analysis: The Role of the Lexical Analyzer - Lexical Analysis Vs. Parsing, Tokens, Patterns and Lexemes, Lexical Errors, Regular Expressions, Regular Definitions, Recognition of Tokens, The Lexical-Analyzer Generator Lex.

Practice:
1 a) Write a lex program which removes white spaces from its input file.
b) Write a program to implement the lexical analyzer using JLex, flex, lex or other lexical analyzer generating tools.

UNIT – II

Syntax Analysis:
The Role of the Parser, Context Free Grammars- Derivations, Parse Trees, Ambiguity, Left Recursion, Left Factoring, Classification of Parsing Techniques.

Top-Down Parsing:
Brute-Force Parsing, Recursive-Descent Parsing, First and Follow, LL(1) Grammars, Nonrecursive Predictive Parsing, Error Recovery in Predictive Parsing.

Practice:
1.Write a C program to implement Recursive Descent Parser for an expression.

UNIT – III

Bottom-Up Parsing:
Reductions, Handle Pruning, Shift-Reduce Parsing, Why LR Parsers, The LR-Parsing Algorithm, Constructing SLR-Parsing Tables, More Powerful LR Parses - Construction of CLR(1), LALR Parsing Tables, Using Ambiguous Grammars.
Practice:
1.Write a C program to implement LALR bottom up parser for the given language.

UNIT – IV

Semantic Analysis:
Syntax-Directed Definitions, Evaluation Orders for SDD's, Syntax Directed Translation. **Intermediate Code Generation:** Three Address Code, Quadruples, Triples, Abstract Syntax Trees, Type Checker - Type Expressions, Type Conversion.
Runtime Environments:
Activation Record, Storage Organization, Storage Allocation Strategies – Static Allocation, Stack Allocation and Heap Allocation.

Practice:
1.Write a C program to implement intermediate code generation for simple expression.

UNIT – V

Code Optimization: Basic Blocks & Flow Graphs, Machine-Independent Optimizations – The Principal Sources of Optimization: Common Subexpression Elimination, Copy Propagation, Dead Code Elimination, Code Motion, Induction Variables and Reduction in Strength, DAG Representation, Machine-dependent Optimization: Peephole Optimization.
Code Generation: Issues in the Design of a Code Generator, Object Code Forms, A Simple Code Generator, Register Allocation and Assignment.

Practice:
1 a) Write a C program to perform Loop Unrolling.
b) Write a C program for constant propagation

TextBooks:

1 Compilers: Principles, Techniques and Tools, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, Second Edition, Pearson,2007. ISBN: 978-1-29202-434-9
2 Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press ISBN: 978-0521607650

Reference Books:

1 Compiler Construction-Principles and Practice, Kenneth C Louden, Cengage Learning. ISBN: 978-0534939724
2 The Compiler Design Handbook, Y.N. Srikant and Priti Shankar, 2nd ed., CRC Press ISBN: 978-1420043822.

Web Links:

1 https://nptel.ac.in/courses/106/108/106108113/
2 https://nptel.ac.in/courses/106/104/106104123/

Machine Learning
(Common to CSE, IT, AIML & CSE(DS))

	L	T	P	C
Course Code: 2501AI05	**2**	**0**	**2**	**4**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Outline the Concepts of Machine Learning and Statistical Learning
CO2: Build Regression and Classification models for given data.
CO3: Apply Instance based Learning techniques and SVM techniques linear and nonlinear data.
CO4: Apply clustering techniques on high dimensional data to group the similar entities.
CO5: Make use of ensemble learning techniques to improve the performance of a model.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	2			2	2		1	1		
CO2	2	2	3		2	2		1	1		
CO3	3	2	3		2			1	1		2
CO4	3	2	3		2			1	1		
CO5	2	2	2		2			1	1		2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1		1
CO2		2
CO3		2
CO4		2
CO5		2

UNIT – I

Introduction:
Machine Learning, Types of Machine Learning Systems, Main Challenges of Machi Learning. Statistical Learning: Introduction, Supervised and Unsupervised Learning, T Learning Problem, Feasibility of Learning.

Practice :
1. Installation of Jupitar/Spider notebook and working on basic commands.
2. Loading and apply key preprocessing techniques on the dataset and also analyse t dataset.

UNIT – II

Supervised Learning(Regression/Classification):Basic Methods:
Concept Learning: General-to-Specific Hypotheses Ordering, Find-S and Candidate Elimination Algorithm, Version Space, and Inductive Bias.
Bayesian Learning: Probability Overview, MLE and MAP Estimates, Gaussian Naive Bayes Classifier, Bayesian Networks.
Instance-based Learning: k-Nearest Neighbour (kNN) Classifier, Voronoi Diagram and Distance-Weighted kNN, Distance Metrics and Curse of Dimensionality.

Practice :
1. Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2. For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
3. Exercises to solve the real-world problems using Binary Classifier
4. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct andwrong predictions.

UNIT – III

Linear Models and Regression: Linear Classification, Linear Regression, Non-linear Transformation, Logistic Regression.
Support Vector Machines: Decision Boundary and Support Vector: Optimization and Primal-Dual Problem, Soft Margin and Non-linear Decision Boundary, Kernel Functions and Radial Basis Functions introduction.

Practice :
1. Exercises to solve the real-world problems using Linear Regression
2. Exercises to solve the real-world problems using Logistic Regression
3. Consider Patient Dataset. Apply linear classification technique(SVM) to identify the rate of heartpatients.

UNIT – IV

Classifier / Hypothesis Evaluation: Accuracy, Precision, Recall and F-Measures, Scores, Sampling, Bootstrapping and ROC, Hypotheses Testing and Cross-validation.
Unsupervised Learning:
Clustering : Partitional Clustering and Hierarchical Clustering, Cluster Types, Attributes and Salient Features,Hierarchical and Density-based Clustering Algorithms, Inter and Intra Clustering Similarity, Cohesion and Separation. MST and DBSCAN Clustering Algorithms.

Practice :
1. Develop a program for Bias, Variance, Remove duplicates , Cross Validation
2. Write a program to implement One-hot Encoding.
3. Write a program to implement Categorical Encoding.

UNIT – V

Ensemble Learning :
Bagging and Boosting, Adaboost and Random Forest ,
Computational Learning Theory : Error and Noise Formalisms, Training vs. Testing, Theory of Generalization,PAC Learnability and VC Dimensions , Overfitting, Regularization and Validation.

Practice :
1. Write a program to demonstrate the working of Random Forest classifier. Use appropriate dataset for Random Forest Classifier.
2. Use a sentiment analysis dataset from Twitter or other social media platforms, available on platforms like Kaggle and apply bagging and boosting techniques for prediction.

Additional Practice:
1. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model toperform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy,precision, and recall for your data set.
2. Apply EM algorithm to cluster a Heart Disease Data Set. Use the same data set for clustering using kMeans algorithm. Compare the results of these two algorithms and comment on the quality of clustering.You can add Java/Python ML library classes/API in the program.

Capstone Project:
Student Performance Prediction System

Objective: Predict a student's performance (pass/fail or score range) based on input features like study time, attendance, past grades, family background, etc.
Domain: Education
Techniques & Concepts Used:
- Supervised learning (Classification – Logistic Regression, SVM)
- Data preprocessing (handling nulls, encoding categorical variables)
- Evaluation metrics (accuracy, precision, recall)

Text Books:

1 Hands On Machine Learning with ScikitLearn, Keras, and TensorFlow, 2nd Edition, O'Reilly Publications, 2019, ISBN: 978-1492032649
2 Data Science and Machine Learning Mathematical and Statistical Methods,Dirk P. Kroese, Zdravko I. Botev, Thomas Taimre, Radislav Vaisman, 2019, ISBN: 978-1138492530

Reference Books:

1 Machine Learning Probabilistic Approach, Kevin P. Murphy, MIT Press, ISBN: 9780262018029
2 Machine Learning Tools, Dr Muddafa Maruli Krishna, Dr A Vanathi, Final Thawaksr, Dr Lokesh P. Gangani, publisher: Book Rivers, ISBN: 978-93-5515-730-0

Web Links:

1 https://www.deeplearning.ai/machinelearningyearning/
2 https://www.cse.huji.ac.il/~shais/UnderstandingMachineLearning/index.html

Big Data Analytics
(Common to CSE, IT, AIML & CSE(DS))

	L	T	P	C
Course Code: 2501AI04	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Illustrate big data challenges in different domains
CO2: Use various techniques for mining data stream.
CO3: Demonstrate building blocks of Hadoop
CO4: Make use of Pig and Hive to structure big Data
CO5: Apply Spark tool to analyze big data.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	3	2		1	2			1	2		1
CO2	2	3		1	2			1	2		1
CO3	2	1		3	2			1	2		2
CO4	2	3		2	2			1	2		2
CO5	2	2		1	3			1	2		2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1		2
CO2		2
CO3		2
CO4		2
CO5		2

UNIT – I

Introduction to big data: Introduction to Big Data Platform, Challenges of Conventional Systems, Intelligent data analysis, Nature of Data, Analytic Processes and Tools, Analysis vs Reporting.
Practice:
1. Hadoop File System Operations: Implement the following file management tasks in Hadoop: a.) Adding files and directories b.) Retrieving files c.) Deleting files

UNIT – II

Stream Processing:
Mining data streams: Introduction to Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting

Distinct Elements in a Stream, Estimating Moments, Counting Oneness in a Window, Decaying Window, Real time Analytics Platform (RTAP) Applications.

Practice:

1. Loading DataSet in to HDFS for Spark Analysis.

UNIT – III

Introduction to Hadoop: History of Hadoop, the Hadoop Distributed File System, Components of Hadoop Analyzing the Data with Hadoop, Scaling Out, Hadoop Streaming, Design of HDFS, Java interfaces to HDFS Basics, Developing a Map Reduce Application, How Map Reduce Works, Anatomy of a Map Reduce Job run, Failures, Job Scheduling, Shuffle and Sort, Task execution, Map Reduce Types and Formats, Map Reduce Features Hadoop environment.

Practice:

1. Map Reduce Word Count Hadoop Run a basic Word Count Map Reduce program to understand Map Reduce Paradigm.

UNIT – IV

Pig: Hadoop Programming Made Easier: Admiring the Pig Architecture, Going with the Pig Latin Application Flow, Working through the ABCs of Pig Latin, Checking out the Pig Script Interfaces, Scripting with Pig Latin. Working with Hive Data Types, Creating and Managing Databases and Tables, Seeing How the Hive Data Manipulation Language Works with examples, Querying and Analyzing Data.

Practice:

1. Hive Queries a) Install Hive Framework b) Implement Hive to create, alter, and drop databases, tables
2. Hive Queries Implement hive queries and joins to perform display and retrieve the data

UNIT – V

Spark: Installing Spark, Spark applications, Jobs, stages and Tasks, Resilient Distributed data sets, Shared Variables, Anatomy of a Spark job run.

Practice:

1. Demonstrate Spark SQL on Hive
   a) Create a SQLContext object and load the Parquet file into DataFrame
   b) Load the Dataframe into Hive table.
   c) Verify the created Hive table in Hive environment
   d) Execute Spark SQL query

Text Books:

1. Hadoop: The Definitive Guide, Tom White, O'reilly, 4th Edition, ISBN: 978-1491901632
2. Hadoop for Dummies, Dirk deRoos, Paul C.Zikopoulos, Roman B.Melnyk, Bruce Brown, Rafael Coss, John Wiley & Sons, ISBN: 978-1118727128

Reference Books:

1	Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, Bill Franks, John Wiley& sons, ISBN: 978-1118224912
2	Harness the Power of Big Data:The IBM Big Data Platform, Paul Zikopoulos, DirkdeRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, Tata McGraw Hill Publications, ISBN: 9780071808187
3	Big Data Science & Analytics: A Hands On Approach , Arshdeep Bahga and Vijay Madisetti, VPT, ISBN: 978-0996025577
4	Learning Spark: Lightning Fast Big Data Analysis Paperback, Holden Karau, ISBN: 978-1449358624

Web Links:

1	Hadoop: http://hadoop.apache.org/
2	Hive: https://cwiki.apache.org/confluence/display/Hive/Home
3	https://nptel.ac.in/courses/106106142
4	https://www.cloudera.com/servicesandsupport/tutorials.html
5	https://www.databricks.com/spark/gettingstartedwithapachespark

Cryptography & Network Security
(Common to CSE, IT, AIML & CSE (DS))

	L	T	P	C
Course Code: 2501IT03	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

- **CO1:** Explain Security Threats and Mathematical Foundations of Cryptography
- **CO2:** Classify the basic principles of symmetric key algorithms.
- **CO3:** Revise the basic principles of public key algorithms and Asymmetric key algorithms
- **CO4:** Design applications of hash algorithms, digital signatures, and key management techniques
- **CO5:** Determine the knowledge of Network layer security Protocols.

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1	3	2		1						2	1
CO2	2	3			2					1	1
CO3	1	3			2					1	1
CO4	2	2	3	2						2	2
CO5	3	2								2	2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	2	
CO2	2	
CO3	2	
CO4	2	
CO5	1	

UNIT – I

Basic Principles: Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography (Integer Arithmetic, Modular Arithmetic, Matrices, Linear Congruence).

Practice:
1. Write a program to obtain GCD of two numbers using Euclidean Algorithm

2. Write a c program to implement elliptical curve cryptography algorithm

UNIT – II

Symmetric Encryption: Traditional Symmetric Key Ciphers (Introduction, Substitution ciphers, Transposition ciphers, Stream and block ciphers), Data Encryption Standard, Advanced Encryption Standard.

Practice:

1. Write a C program to implement Mono alphabetic substitution cipher
2. Write a C program to implement Vernam Cipher

UNIT – III

Asymmetric Encryption: Mathematics of Asymmetric Key Cryptography (Euler's Phi Function, Fermat's Little Theorem, Euler's Theorem, Chinese Reminder Theorem), Asymmetric Key Cryptography(Introduction, RSA Cryptography, Diffie Hellman key exchange, Rabin Cryptosystem, Elgamal Cryptosystem)

Practice:

1. Write a C program to implement Rabin Cryptosystem

UNIT – IV

Digital Signature Schemes & Key Management: Message Integrity and Message Authentication, Cryptographic Hash Functions (SHA-512), Digital Signatures(Process, Services, Attacks), Key Management (Kerberos, X.509).

Practice:

1. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

UNIT – V

Network Security: PGP and S/MIME, SSL and TLS, IPSec (Modes, Security Protocols),IPC.

Practice:

1. Implement the following forms of IPC

a. Pipes

b. FIFO

Capstone Project:

1. Design and implement a Secure Messaging Application that ensures confidentiality and integrity of messages using Hybrid Cryptography (combination of symmetric and asymmetric encryption).Use RSA or ElGamal for secure key exchange and AES for encrypting the actual message content.Implement SHA-512 to generate message digests for integrity verification.The application should support basic message exchange between two users with proper key management.Demonstrate the working of encryption, decryption, key generation, and integrity verification through a user-friendly interface or command-line tool.

Text Books:

1. Cryptography and Network Security: Principles and Practice, William Stallings, Pearson, 9th Edition , ISBN-13: 9781292742366, 2023.
2. Introduction to Modern Cryptography: Principles and Protocols, Jonathan Katz, Yehuda Lindell, CRC Press, 4th Edition , ISBN 13: 9781584885511, 2023.

Reference Books:

1 Understanding Cryptography: A Textbook for Students and Practitioners, Christof Paar, Jan Pelzl, Springer, 2nd Edition, 2020**,** ISBN: 978-3642041006.

2 Network Security Essentials: Applications and Standards, William Stallings, Pearson, 7th Edition, 2023, ISBN: 978-8131761755

Web Links:

1 https://nptel.ac.in/courses/106105031

2 https://users.cs.northwestern.edu/~ychen/classes/cs395-w05/lectures.html

Software Architecture
(Common to CSE&IT)

	L	T	P	C
Course Code: 2501CS15	**2**	**0**	**1**	**3**

Course Outcomes:
At the end of the course, student will be able to:

CO1: Explain architectural styles of software
CO2: Illustrate software architectural models
CO3: Analyze quality attributes of software architecture
CO4: Explain architectural and design process.
CO5: Explain user interface architecture design for real time applications

Mapping of Course Outcomes with Program Outcomes:

CO/PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11
CO1	2	1		3	2						2
CO2	2	2	3	2	2						1
CO3	2	1	1	1	3						1
CO4	2	3		2	2						2
CO5	2	3	1	1	2						2

Mapping of Course Outcomes with Program Specific Outcomes:

CO/PSO	PSO1	PSO2
CO1	1	
CO2	1	
CO3	1	
CO4	1	
CO5	1	

UNIT – I

Introduction- Software Architecture- Definition Prospects- State of Art-Architectural Styles-Pipes and Filters-Layered Systems-Repositories-Process Control, Other familiar Architecture-Heterogeneous Architectures

Practice:
1. Study case tools such as rational rose or equivalent tools

UNIT – II

The Architecture Business Cycle, what is Software Architecture, Architectural patterns, reference models, reference architectures, architectural structures and views.

Practice:

1. Requirements implementation of requirements engineering activities such as elicitation, validation management using case tools

UNIT – III

Creating and Architecture Quality Attributes, achieving qualities, Architectural styles and patterns, designing the Architecture, Documenting software architectures, Reconstructing Software Architecture.

Practice:

1. Study and usage of software project management tools such cost estimates and scheduling

UNIT – IV

Architecture Evaluation, Architecture design decision making, Building systems from off the shelfcomponents, Software architecture in future.

Practice:

1. Exposure towards test plan generators, test case generators, test coverage and software metrics

UNIT – V

Architecture design- Introduction Shared Information System, Architecture Structures for Shared Information Systems. Architecture design guidance-User Interface Architecture. Case study: Design of User Interface Design.

Practice:

1. Case study: Design of User Interface Design

Text Books:

1	Software Architecture in **Practice**, second edition, Len Bass, Paul Clements & RickKazman, Pearson Education. ISBN :9788177589962
2	Designing Software Architectures: A Practical Approach" by Humberto Cervantes and Rick Kazman ISBN: 978-0134390789

Reference Books:

1	Beyond Software architecture, Luke Hohmann, Addison wesley. ISBN:9780201775945
2	Software architecture, David M. Dikel, David Kane and James R. Wilson, Prentice Hall PTR. ISBN: 9780130290328

Web Links:

1	https://nptel.ac.in/courses/106105224
2	https://www.geeksforgeeks.org/fundamentals-of-software-architecture/
3	https://core.ac.uk/download/pdf/36753454.pdf