

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	-
2501MA02	Differential Equations & Vector Calculus	FC	2	1		3	50	50	100	-
2501CS71	Computer Organization & Architecture	FC	2	1		3	50	50	100	DLD
2501PH02	Modern Physics	FC	2		1	3	50	50	100	-
2501ME01	Engineering Graphics	FC	1		2	3	50	50	100	-
2501IT42	Programming with Python	FC	2		2	4	50	50	100	PPSC
2501CS01	Programming for Problem Solving using C	FC	2		2	4	50	50	100	-
2501CS03	Data Structures	FC	2		2	4	50	50	100	PPSC
2501IT01	Business Intelligence Lab	FC			2	2	50	50	100	-
2501MA08	Discrete Mathematics	IC	2	1		3	50	50	100	LAC
2501MA09	Probability & Statistics	IC	2	1		3	50	50	100	LAC
2501AI02	Artificial Intelligence	IC	2		1	3	50	50	100	-
2501CS08	Object Oriented Programming through C++	IC	2		2	4	50	50	100	PPSC
2501IT05	Database Management Systems	IC	2		2	4	50	50	100	PPSC
2501AI03	Data Mining	IC	1		2	3	50	50	100	-
2501IT06	Java Programming	IC	2		2	4	50	50	100	PPSC
2501CS09	Language Processors	IC	2	1		3	50	50	100	PPSC
2501IT07	Agile Software Engineering	IC	2		1	3	50	50	100	PPSC
2501CS13	Operating Systems	IC	2		1	3	50	50	100	-
2501CS07	Computer Networks	IC	2		1	3	50	50	100	-
2501AI05	Machine Learning	IC	2		2	4	50	50	100	DAE
2501CS10	Advanced Data Structures & Algorithm Analysis	AC	2		1	3	50	50	100	DS
2501AI11	Deep Learning	AC	2		2	4	50	50	100	DAE
2501AI20	Natural Language Processing & Prompt Engineering	AC	1		2	3	50	50	100	ML
2501AI21	Reinforcement Learning & Gen AI	AC	1		2	3	50	50	100	ML
2501AI04	Big Data Analytics	AC	2		1	3	50	50	100	DM
Total			46	6	33	85				

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, Eigenvectors : 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):

1. 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):

1. 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):

1. Plotting the region of Integration

Students are advised to use any computational / AI Tool like Wolfram 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>

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

Course Code: 2501MA02	L	T	P	C
	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 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) :

1. 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, Legendre's Linear Equations. Study of oscillations arising in LCR circuit (free oscillations and forced oscillations).

Practice(Using any computational tool) :

1. 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) :

1. 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 Wolfram 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/>

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

Course Code: 2501CS71

L	T	P	C
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 1	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, Zvonko Vranesic, Safwat Zaky, 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>

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

Course Code: 2501PH02	L	T	P	C
	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	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	-	-	-	-	-	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 Fibres

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: 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 Introduction – Intrinsic semiconductors – density of charge carriers – Electrical conductivity –Fermi level - extrinsic semiconductors - P-type & N-type semiconductors-Density of charge carriers (Qualitative) - 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 – Zener 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.

Text Books:

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

Reference Books:

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

Engineering Graphics

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

Course Code: 2501ME01	L	T	P	C
	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:

- 1. Introduction to Engineering Graphics**
 - a. Introduction to AutoCAD, Dimensioning, elements of dimensioning, system of dimensioning, Conventions in Drawing.

- 2. Construction of Cycloids and Involutes**
 - a. Construction of Cycloid, Epicycloid and Hypocycloid
 - b. Involute of a pentagon and circle

- 3. 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

- 4. 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

- 5. Projection of straight lines-II**
 - a. Projections of straight line inclined to both reference planes

6. **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.
7. **Projections of Planes**
 - a. Projections of Planes inclined to both reference planes Ex: Rectangle, Pentagon, Hexagon and Rhombus etc.
8. **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
Ex: Pentagonal and Hexagonal Prisms, Pyramids, Cylinder and Cone
9. **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
10. **Development of Surfaces**
 - a. Development of a Prisms and Cylinder simple cases
11. **Development of Surfaces**
 - a. Development of Pyramids and Cone simple cases
12. **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, 1st Edition, 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>

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

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

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

- CO1:** Develop Python programs using basic programming constructs, control statements, loops.
- CO2:** Implement data processing solutions using Python’s string methods and core data structures.
- CO3:** Develop real time applications using functions and packages.
- 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	-	2	-	-	1	-	-	1
CO2	2	2	3	-	2	-	-	1	-	-	1
CO3	3	-	2	-	2	-	-	2	1	1	1
CO4	2	1	2	-	2	-	-	1	1	1	1
CO5	1	1	2	-	2	-	-	2	1	1	1

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Programming:

Introduction to Programming, Introduction to Python, Program Development Cycle, Input–Processing–Output, Print Function, Comments, Variables, Keyboard Input, Basic Calculations, Operators, Type Conversions, Expressions, Data Types, Strings, Using Functions, Using Modules. Decision Making: if, if-else, if-elif-else, Nested Decisions, Logical Operators, Boolean Expressions. Loops: while, for, Running Totals, Input Validation, Nested Loops.

Practice:

1. Basic Programs.
 - a. Adding two numbers
<https://www.codechef.com/problems/FLOW001>
 - b. Perfect Square root or not

<https://maya.technicalhub.io/owl-program-details/668d0b25aa293a75dcf2dda4>

- c. Perfect Number

<https://maya.technicalhub.io/owl-program-details/668b717fd24b0e6bee89ce97>

- d. Even Number or Odd number

<https://www.hackerrank.com/challenges/py-if-else/problem?isFullScreen=true>

2. Decision Structures and Loops.

- a. Python Loops – Printing Squares of Numbers

<https://www.hackerrank.com/challenges/python-loops/problem?isFullScreen=true>

- b. Check for Armstrong number, Palindrome.

<https://www.hackerrank.com/contests/iiitdm-contest1/challenges/armstrong-number-2?utm>

- c. Spy Number

<https://maya.technicalhub.io/owl-program-details/6688e610e467bfe336f89d72>

- d. Compound Interest

<https://maya.technicalhub.io/owl-program-details/6694bce1d5d1d7533859f1d8>

- e. Count vowels, consonants, digits in a string.

<https://www.codechef.com/problems/CV>

- f. Print a number pyramid.

<https://www.codechef.com/problems/PATT01>

UNIT – II

Python Data Types:

Working with Strings: Basic String Operations, String Formatting, String Methods. Lists and Tuples: Sequence Operations, Lists, Tuples, Tuple Function. Dictionaries: Creating and Using Dictionaries, Basic Dictionary Operations, Dictionary Methods. Sets: Creating Sets, Basic Set Operations, Set Methods, Simple Set Comprehensions.

Practice:

- a. String Matching in an Array

<https://leetcode.com/problems/string-matching-in-an-array/>

- b. Product-of-Three-Numbers

<https://leetcode.com/problems/maximum-product-of-three-numbers>

- c. Write a program to check if the substring is present in a given string or not.

<https://www.hackerrank.com/challenges/find-a-string/problem?utm>

- d. Reverse-vowels-of-a-String

<https://leetcode.com/problems/reverse-vowels-of-a-string>

Lists, Tuples, Dictionary

- a. Generate 20 random numbers; print min, max, avg, even count.

<https://www.codechef.com/problems/FLOW002>

b. **Longest Consecutive Sequence**

<https://leetcode.com/problems/longest-consecutive-sequence/>

c. **Subarray Sum Equals K**

<https://leetcode.com/problems/subarray-sum-equals-k/>

UNIT – III

Functions and Packages: Function Definition, Parameters, Local vs Global Variables, Default Arguments, Keyword Arguments, Return Statement, Lambda Functions. Modules: import, from–import, Creating Simple Modules, Standard Modules, Packages. Basic Problem Solving: Top-Down Design, Simple Recursion Concepts.

Practice:

1. Working with Functions

- a. Write a Python program to define a function that takes parameters and returns the sum of two numbers.

<https://www.geeksforgeeks.org/python-program-to-find-factorial-of-number-using-recursion/>

- b. **Two Sum** — a classic easy problem that’s often solved using a function. <https://leetcode.com/problems/two-sum/>

- c. **Valid Anagram** — good practice for string/dictionary manipulations inside a function. <https://leetcode.com/problems/valid-anagram/>

- d. **FLOW001** — “Add Two Numbers”: simplest problem, good for writing a function to sum two numbers. <https://www.codechef.com/problems/FLOW001>

- e. **FLOW002** — “Compute the Average”: practice writing a function to compute averages. <https://www.codechef.com/problems/FLOW002>

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, 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.

<https://www.codechef.com/problems/STFOOD>

- b. Define Converter class for unit conversions (e.g., inches → feet).

<https://www.codechef.com/problems/CONV01>

- c. Create a Time class to convert seconds → minutes, hours.

<https://www.codechef.com/problems/TIMEMNG>

2. File Handling

- a. **Say "Hello, World!" from a file** (read input from file, print output):
<https://www.hackerrank.com/challenges/write-a-function/problem>
- b. **Reading from STDIN / file and processing lines:**
<https://www.hackerrank.com/challenges/whats-your-name/problem>
- c. **FLOW001 – Add Two Numbers:** <https://www.codechef.com/problems/FLOW001>

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 Behaviour of Terminal Based Programs and GUI - Based, Programs, Coding Simple GUI-Based Programs.

Practice:

1. GUI and Exceptions
 - a. Build a file-open dialog and display content in a text box.
<https://www.codechef.com/problems/ERROR404>
 - b. **Python Try/Except** — basic problem to handle division by zero and other exceptions:
<https://www.hackerrank.com/challenges/py-exceptions/problem>
 - c. **Handling Multiple Exceptions** — practise catching multiple exception types in one block:
<https://www.hackerrank.com/challenges/handling-multiple-exceptions/problem>
 - d. Show file handling using try/finally and with statements.
<https://www.codechef.com/problems/VALIDMIN>
2. Integrated Problem Solving
 - a. Implement the Hangman game using string logic and loops.
<https://www.codechef.com/problems/STRPALIN>

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/

Additional Practice:

LPYAS10 – <i>Print the Pattern</i>	https://www.codechef.com/practice/course/python/LPPYAS01/problems/LPYAS10
LPYAS09 – <i>Simple Input Output</i>	https://www.codechef.com/practice/course/python/LPPYAS01/problems/LPYAS09
LPYAS11 – <i>Check Even or Odd</i>	https://www.codechef.com/practice/course/python/LPPYAS02/problems/LPYAS11
LPYAS19 – <i>Sum of Digits</i>	https://www.codechef.com/practice/course/python/LPPYAS02/problems/LPYAS19
REMMON – <i>Remaining Monsters</i>	https://www.codechef.com/problems/REMMON
EPISODES – <i>Episodes</i>	https://www.codechef.com/problems/EPISODES
TRIANGLE7 – <i>Triangle Type</i>	https://www.codechef.com/problems/TRIANGLE7
COLORB – <i>Color Blindness</i>	https://www.codechef.com/problems/COLORB
CABRIDE – <i>Cab Ride</i>	https://www.codechef.com/problems/CABRIDE
SKONG – <i>Strong Language</i>	https://www.codechef.com/problems/SKONG
SELLCOIN – <i>Sell Coins</i>	https://www.codechef.com/problems/SELLCOIN
MIXINGLIQ – <i>Mixing Liquids</i>	https://www.codechef.com/problems/MIXINGLIQ
GAMBLING – <i>Gambling</i>	https://www.codechef.com/problems/GAMBLING
INCAT – <i>Increase the Number</i>	https://www.codechef.com/problems/INCAT
INDDAY – <i>Independence Day Offer</i>	https://www.codechef.com/problems/INDDAY
CWCTH – <i>Count the Holidays</i>	https://www.codechef.com/problems/CWCTH
Palindrome Number	https://leetcode.com/problems/palindrome-number
Valid Perfect Square	https://leetcode.com/problems/valid-perfect-square
Count the Digits That Divide a Number	https://leetcode.com/problems/count-the-digits-that-divide-a-number
Subtract the Product and Sum of Digits of an Integer	https://leetcode.com/problems/subtract-the-product-and-sum-of-digits-of-an-integer
Three Divisors	https://leetcode.com/problems/three-divisors

Smallest Even Multiple	https://leetcode.com/problems/smallest-even-multiple
Happy Number	https://leetcode.com/problems/happy-number
Ugly Number	https://leetcode.com/problems/ugly-number
Power of Two	https://leetcode.com/problems/power-of-two
Power of Three	https://leetcode.com/problems/power-of-three

	Maya	Code chef	Hacker rank	Leet code	Geeksfor Geeks	Total
Unit-1	4	3	3	0	0	10

Power of Four	https://leetcode.com/problems/power-of-four
Arranging Coins	https://leetcode.com/problems/arranging-coins
Excel Sheet Column Title	https://leetcode.com/problems/excel-sheet-column-title
Excel Sheet Column Number	https://leetcode.com/problems/excel-sheet-column-number
Maya	
Integer Division	https://maya.technicalhub.io/owl-program-details/6694b1ef519a4355f15c590f
Area of a Circle	https://maya.technicalhub.io/owl-program-details/6694b694519a4355f15ca75f
Compound interest	https://maya.technicalhub.io/owl-program-details/6694bce1d5d1d7533859f1d8
Simple Interest	https://maya.technicalhub.io/owl-program-details/6694bfb7d5d1d7533859f3ae
Neon Number	https://maya.technicalhub.io/owl-program-details/668b6b34d24b0e6bee8982b0
Nearest Fibonacci	https://maya.technicalhub.io/owl-program-details/668b7ce1126457feec5c8de6
Reverse Integer	https://maya.technicalhub.io/owl-program-details/668b6e86d24b0e6bee89a871
Amicable Numbers	https://maya.technicalhub.io/owl-program-details/668b7461126457feec5c4679
Automorphic Number	https://maya.technicalhub.io/owl-program-details/668cff57aa293a75dcf28f92
Unique Number	https://maya.technicalhub.io/owl-program-details/668d0275aa293a75dcf2c620
Pronic Number or Heteromecic Number	https://maya.technicalhub.io/owl-program-details/6694fe34e0258d3db2167aa9

Unit-2	0	1	1	5	0	7
Unit-3	0	2	0	2	1	5
Unit-4	0	4	2	0	0	6
Unit-5	0	3	2	0	0	5
Additional Programs	11	16	0	16	0	43
Total	16	29	8	23	1	76

Programming for Problem Solving using C
(Common to CE, EEE, ME, ECE, CSE, IT, AIML, CSE(DS), PT & Min.E)

Course Code: 2501CS01	L	T	P	C
	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.
<https://www.codechef.com/learn/course/c/CDEVINTRO/problems/CDEV004B>
 - 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>

<https://maya.technicalhub.io/owl-program-details/668b9ec2e069313510b0235a>

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>

<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.

<https://www.codechef.com/practice/course/c/LPCAS03/problems/LCAS30>

<https://maya.technicalhub.io/owl-program-details/66a0c0dda7d31805366af641>

e. Distance travelled by an object.

<https://maya.technicalhub.io/owl-program-details/6687c4e9fd042085d9ec254a>

f. Calculate Simple interest and compound interest.

<https://maya.technicalhub.io/owl-program-details/6687b498dfe02cea4b53e0ab>

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.

<https://maya.technicalhub.io/owl-program-details/6686546930bfbd35127c869c>

<https://maya.technicalhub.io/owl-program-details/6688cf91e467bfe336f8352f>

https://www.hackerrank.com/challenges/rectangle-area/problem?utm_source=chatgpt.com

https://www.hackerrank.com/challenges/rectangle-area/problem?utm_source=chatgpt.com

<https://maya.technicalhub.io/owl-program-details/6688ea3ce467bfe336f8ab9d>

d. Find the maximum of three numbers using conditional operator.

<https://maya.technicalhub.io/owl-program-details/668690d230bfbd35127c9bf5>

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.
<https://maya.technicalhub.io/owl-program-details/66868f6730bfd35127c9ba7>
 - 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>
<https://maya.technicalhub.io/owl-program-details/66864717f56afec5c0c917e1>
 - c. Given number is a prime or not. (Also Prime numbers between a given range.)
<https://maya.technicalhub.io/owl-program-details/6687b69ddfe02cea4b5404a3>
 - d. Armstrong Number or not.
 - e. Palindrome or not.
<https://maya.technicalhub.io/owl-program-details/6687af17dfe02cea4b5376ae>
 - 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.
<https://www.codechef.com/practice/course/c/LPCAS02/problems/LCAS20>

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-Dimensional 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.
 - c. Search an element in array (Linear Search).
<https://maya.technicalhub.io/owl-program-details/6687c40ffd042085d9ec15d6>
 - d. Find min and max elements in array.

<https://maya.technicalhub.io/owl-program-details/66879493dfe02cea4b529041>

- e. Insert an element into array.
- f. Eliminate duplicate elements from array.
<https://maya.technicalhub.io/owl-program-details/66e920a1aca93b27f3364b57>
- g. Sorting of elements in an array using Bubble sort.
2. 2-Dimensional Arrays
 - a. Sum of two 2-D arrays.
<https://maya.technicalhub.io/owl-program-details/66864fa9f56afec5c0c924f6>
 - 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.
 - b. Count number of alphabets (lowercase, uppercase, consonants, vowels) and digits
Lowercase to Uppercase, Uppercase to Lowercase, Toggle case, Sentential case
<https://maya.technicalhub.io/owl-program-details/6698e2fbc6bd470f35b73d8>
<https://maya.technicalhub.io/owl-program-details/6698ef03cb6bd470f35b7c3e>
 - c. 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.
<https://maya.technicalhub.io/owl-program-details/6698ed2dcb6bd470f35b7ae7>

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: Nth factorial using recursion.
4. Objective: Find the super digit of the integer.
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.

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/>

Additional Practice:

S.No	Difficulty	Problem Name	Link
1	Easy	Add Two Numbers	https://www.codechef.com/practice/course/cpp/PCPP05/problems/FLOW001
2	Easy	Enormous Input Test	https://www.codechef.com/practice/course/basic-programming-concepts/DIFF500/problems/INTEST
3	Easy	ATM	https://www.codechef.com/practice/course/1-star-difficulty-problems/DIFF1200/problems/ATM2
4	Easy	Number Mirror	https://www.codechef.com/practice/course/python/PPY01/problems/START01
5	Easy	Sum of Digits	https://www.codechef.com/practice/course/basic-math/BASICMATH/problems/FLOW006
6	Easy	Remainder	https://www.codechef.com/practice/course/basic-math/BASICMATH/problems/FLOW002
7	Easy	First and Last Digit	https://www.codechef.com/practice/course/basic-programming-concepts/DIFF500/problems/FLOW004
8	Easy	Reverse The Number	https://www.codechef.com/practice/course/logical-problems/DIFF800/problems/FLOW007
9	Easy	Life, the Universe, and Everything	https://staging.codechef.com/practice/course/cpp/PRACTICECPP2/problems/TEST
10	Easy	Chef and Operators	https://www.codechef.com/practice/course/logical-problems/DIFF800/problems/CHOPRT
11	Easy	Difference and Sum	https://www.codechef.com/problems/DIFFSUM
12	Easy	Second Largest	https://www.codechef.com/practice/course/arrays/ARRAYS/problems/LARGESECOND
13	Easy	Two Sum	https://leetcode.com/problems/two-sum/
14	Easy	Reverse Integer	https://leetcode.com/problems/reverse-integer/
15	Easy	Palindrome Number	https://leetcode.com/problems/palindrome-number/
16	Easy	Roman to Integer	https://leetcode.com/problems/roman-to-integer/
17	Easy	Remove Duplicates from Sorted Array	https://leetcode.com/problems/remove-duplicates-from-sorted-array/
18	Easy	Remove Element	https://leetcode.com/problems/remove-element/
19	Easy	Find the Index of the First Occurrence in a String	https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string/
20	Easy	Length of Last Word	https://leetcode.com/problems/length-of-last-word/
21	Easy	Plus One	https://leetcode.com/problems/plus-one/
22	Easy	Merge Sorted Array	https://leetcode.com/problems/merge-sorted-array/
23	Easy	Majority Element	https://leetcode.com/problems/majority-element/
24	Easy	Single Number	https://leetcode.com/problems/single-number/
25	Easy	Move Zeroes	https://leetcode.com/problems/move-zeroes/

26	Easy	Missing Number	https://leetcode.com/problems/missing-number/
27	Easy	Contains Duplicate	https://leetcode.com/problems/contains-duplicate/
28	Easy	Valid Anagram	https://leetcode.com/problems/valid-anagram/
29	Easy	Valid Parentheses	https://leetcode.com/problems/valid-parentheses/
30	Easy	Fibonacci Number	https://leetcode.com/problems/fibonacci-number/
31	Easy	Climbing Stairs	https://leetcode.com/problems/climbing-stairs/
32	Easy	Binary Search	https://leetcode.com/problems/binary-search/
33	Easy	Number of 1 Bits	https://leetcode.com/problems/number-of-1-bits/
34	Easy	Hamming Weight	https://leetcode.com/problems/number-of-1-bits/description/
35	Easy	Power of Two	https://leetcode.com/problems/power-of-two/
36	Easy	Power of Three	https://leetcode.com/problems/power-of-three/
37	Easy	Same Tree	https://leetcode.com/problems/same-tree/
38	Easy	Maximum Subarray	https://leetcode.com/problems/maximum-subarray/
39	Medium	Chef and Notebooks	https://www.codechef.com/problems/CNOTE
40	Medium	Chef and Wildcard Matching	https://www.codechef.com/search?q=TWOSTR
41	Medium	Minimum Moves	https://www.codechef.com/search?q=SALARY
42	Medium	Lucky Four	https://www.codechef.com/search?q=LUCKFOUR
43	Medium	Maximum Difference	https://www.codechef.com/search?q=MAXDIFF
44	Medium	Uncle Johny	https://www.codechef.com/search?q=JOHNY
45	Medium	Chef and Strings	https://www.codechef.com/search?q=CHEFSTLT
46	Medium	Two vs Rib	https://www.codechef.com/search?q=TWOVRIB
47	Medium	Chef and Array	https://www.codechef.com/search?q=CHEFARRP
48	Medium	Add Two Numbers	https://leetcode.com/problems/add-two-numbers/
49	Medium	Longest Substring Without Repeating Characters	https://leetcode.com/problems/longest-substring-without-repeating-characters/
50	Medium	Longest Palindromic Substring	https://leetcode.com/problems/longest-palindromic-substring/
51	Medium	3Sum	https://leetcode.com/problems/3sum/
52	Medium	Container With Most Water	https://leetcode.com/problems/container-with-most-water/
53	Medium	Integer to Roman	https://leetcode.com/problems/integer-to-roman/
54	Medium	String to Integer (atoi)	https://leetcode.com/problems/string-to-integer-atoi/
55	Medium	Divide Two Integers	https://leetcode.com/problems/divide-two-integers/
56	Medium	Rotate Image	https://leetcode.com/problems/rotate-image/

57	Medium	Set Matrix Zeroes	https://leetcode.com/problems/set-matrix-zeroes/
58	Medium	Spiral Matrix	https://leetcode.com/problems/spiral-matrix/
59	Medium	Jump Game	https://leetcode.com/problems/jump-game/
60	Medium	Permutations	https://leetcode.com/problems/permutations/
61	Medium	Combination Sum	https://leetcode.com/problems/combination-sum/
62	Medium	Search in Rotated Sorted Array	https://leetcode.com/problems/search-in-rotated-sorted-array/
63	Medium	Group Anagrams	https://leetcode.com/problems/group-anagrams/
64	Medium	Sort Colors	https://leetcode.com/problems/sort-colors/
65	Medium	Subsets	https://leetcode.com/problems/subsets/
66	Medium	Reverse Linked List II	https://leetcode.com/problems/reverse-linked-list-ii/
67	Medium	Add Two Numbers II	https://leetcode.com/problems/add-two-numbers-ii/
68	Medium	Reorder List	https://leetcode.com/problems/reorder-list/
69	Medium	Linked List Cycle II	https://leetcode.com/problems/linked-list-cycle-ii/
70	Medium	K-th Largest Element in an Array	https://leetcode.com/problems/kth-largest-element-in-an-array/
71	Medium	Product of Array Except Self	https://leetcode.com/problems/product-of-array-except-self/
72	Medium	Word Search	https://leetcode.com/problems/word-search/
73	Medium	Unique Paths	https://leetcode.com/problems/unique-paths/
74	Medium	Minimum Path Sum	https://leetcode.com/problems/minimum-path-sum/
75	Medium	Temperature Converter - II	https://www.codechef.com/practice/course/c/LPCAS03/problems/LCAS30B
76	Medium	Reverse	https://www.hackerrank.com/challenges/reverse-array-c/problem?isFullScreen=true
77	Medium	Sentence in a new line	https://www.hackerrank.com/challenges/printing-tokens-/problem?isFullScreen=true
78	Hard	Median of Two Sorted Arrays	https://leetcode.com/problems/median-of-two-sorted-arrays/
79	Hard	Regular Expression Matching	https://leetcode.com/problems/regular-expression-matching/
80	Hard	Wildcard Matching	https://leetcode.com/problems/wildcard-matching/
81	Hard	Trapping Rain Water	https://leetcode.com/problems/trapping-rain-water/
82	Hard	First Missing Positive	https://leetcode.com/problems/first-missing-positive/
83	Hard	Longest Valid Parentheses	https://leetcode.com/problems/longest-valid-parentheses/
84	Hard	Merge k Sorted Lists	https://leetcode.com/problems/merge-k-sorted-lists/
85	Hard	Reverse Nodes in k-	https://leetcode.com/problems/reverse-nodes-in-k-

		Group	group/description/
86	Hard	Sudoku Solver	https://leetcode.com/problems/sudoku-solver/
87	Hard	N-Queens	https://leetcode.com/problems/n-queens/description/
88	Hard	Edit Distance	https://leetcode.com/problems/edit-distance/
89	Hard	Word Ladder II	https://leetcode.com/problems/word-ladder-ii/
90	Hard	Maximal Rectangle	https://leetcode.com/problems/maximal-rectangle/
91	Hard	Binary Tree Maximum Path Sum	https://leetcode.com/problems/binary-tree-maximum-path-sum/
92	Hard	Serialize and Deserialize Binary Tree	https://leetcode.com/problems/serialize-and-deserialize-binary-tree/
93	Hard	Variadic functions in C	https://www.hackerrank.com/challenges/variadic-functions-in-c/problem?isFullScreen=true
94	Hard	Small Triangles, Large Triangles	https://www.hackerrank.com/challenges/small-triangles-large-triangles/problem?isFullScreen=true
95	Hard	Permutations of Strings	https://www.hackerrank.com/challenges/permutations-of-strings/problem?isFullScreen=true

Maya Programs

S.No.	Problem Name	Maya Program Link
1	Can Cross the Bridge	https://maya.technicalhub.io/owl-program-details/6686308df56afec5c0c8e2dd
2	Hypotenuse	https://maya.technicalhub.io/owl-program-details/6686404ef56afec5c0c90135
3	Heron's Formula	https://maya.technicalhub.io/owl-program-details/66864474f56afec5c0c911f5
4	Loss Percentage	https://maya.technicalhub.io/owl-program-details/66864868f56afec5c0c91a4a
5	King Tours	https://maya.technicalhub.io/owl-program-details/66864c96f56afec5c0c921f1
6	Profit Percentage	https://maya.technicalhub.io/owl-program-details/66865237f56afec5c0c92694
7	Area and Perimeter of Square	https://maya.technicalhub.io/owl-program-details/6686546930bfd35127c869c
8	Volume of Sphere	https://maya.technicalhub.io/owl-program-details/6686565530bfd35127c8818
9	Required Points	https://maya.technicalhub.io/owl-program-details/66865b1130bfd35127c8a04
10	Instant Noodles	https://maya.technicalhub.io/owl-program-details/66865ddb30bfd35127c8c0a
11	Find Second Number	https://maya.technicalhub.io/owl-program-details/668660a530bfd35127c8cde
12	Kmph to Mps	https://maya.technicalhub.io/owl-program-details/668663ce30bfd35127c8dfb
13	Inches to Centimeters	https://maya.technicalhub.io/owl-program-details/6686665e30bfd35127c8f3a

14	Capacity	https://maya.technicalhub.io/owl-program-details/668678aa30bfd35127c95b9
15	Average of Two Numbers	https://maya.technicalhub.io/owl-program-details/6687b002dfe02cea4b5389d6
16	Compound Interest	https://maya.technicalhub.io/owl-program-details/6687b498dfe02cea4b53e0ab
17	Average Weight	https://maya.technicalhub.io/owl-program-details/6687b93edfe02cea4b54367a
18	Arithmetic Operations	https://maya.technicalhub.io/owl-program-details/6687bda0dfe02cea4b5488a1
19	Distance Between Two Points	https://maya.technicalhub.io/owl-program-details/6687c4e9fd042085d9ec254a
20	Days into Years, Weeks	https://maya.technicalhub.io/owl-program-details/6687cad1fd042085d9ec95a8
21	Convert Seconds to Hours, Minutes and Seconds	https://maya.technicalhub.io/owl-program-details/6688c926e467bfe336f8007b
22	Hours and Minutes	https://maya.technicalhub.io/owl-program-details/6688e4f0e467bfe336f89673
23	Romeo and Juliet	https://maya.technicalhub.io/owl-program-details/6688e068e467bfe336f87a8b
24	Gross Salary of an Employee	https://maya.technicalhub.io/owl-program-details/6688dd95e467bfe336f87682
25	Surface Area and Volume of a Cube	https://maya.technicalhub.io/owl-program-details/6688d909e467bfe336f872ca
26	Area of Trapezium	https://maya.technicalhub.io/owl-program-details/6688d637e467bfe336f86c5e
27	Swap Two Numbers	https://maya.technicalhub.io/owl-program-details/6688d2d0e467bfe336f84f55
28	Last Two Digits of a Given Year	https://maya.technicalhub.io/owl-program-details/6688ccfae467bfe336f81f90
29	Arithmetic Operations	https://maya.technicalhub.io/owl-program-details/6687bda0dfe02cea4b5488a1
30	Uppercase Letters in a String	https://maya.technicalhub.io/owl-program-details/669a2453c03fc56b320b215c
31	Factors Finding	https://maya.technicalhub.io/owl-program-details/66866c4a30bfd35127c90b4
32	String Copy	https://maya.technicalhub.io/owl-program-details/669a03ded62716a9c6d4fa9a
33	Recursive Digit Sum	https://www.hackerrank.com/challenges/recursive-digit-sum/problem

	Basic Programs	Aditya's Internal Coding Platform (Maya)	Code chef	Hacker rank	Leet code	Total
Unit-1	4	8	2	5	0	19
Unit-2	4	4	1	4	0	13
Unit-3	8	7	0	2	0	17
Unit-4	3	0	0	3	0	6
Unit-5	7	0	4	4	0	15
Additional Practice	0	33	22	5	68	128
Total	26	52	29	23	68	198

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

Course Code: 2501CS03	L	T	P	C
	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.
<https://www.hackerrank.com/contests/hw1/challenges/merge-sort>
2. Implement Quick sort using arrays
<https://www.hackerrank.com/challenges/quicksort1/problem>
3. Implement Radix Sort using arrays
https://www.hackerrank.com/contests/hw1/challenges/radix-sort?utm_source

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.
https://leetcode.com/problems/design-linked-list/description/?utm_source
2. Perform different operations in double linked list
https://leetcode.com/problems/design-linked-list/?utm_source
3. Circular Linked List
https://www.codechef.com/practice/course/linked-lists/LINKLISTF/problems/PREP58?utm_source

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
<https://www.hackerrank.com/challenges/balanced-brackets/problem>
https://leetcode.com/problems/design-linked-list/description/?utm_source
2. Implement a program to evaluate a postfix expression.
<https://leetcode.com/problems/evaluate-reverse-polish-notation/description/>
3. Queue and its operations using arrays and Linked List
<https://leetcode.com/problems/implement-queue-using-stacks/description/>
https://leetcode.com/problems/design-linked-list/description/?utm_source
4. Implement Circular Queue using Arrays
<https://leetcode.com/problems/design-circular-queue/description/>

UNIT – V

Non-linear Data Structures:

Trees: Definition 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).
<https://leetcode.com/problems/insert-into-a-binary-search-tree/description/>
<https://leetcode.com/problems/delete-node-in-a-bst/description/>
2. Implement Binary search Tree (BST) Traversals.
<https://www.hackerrank.com/challenges/tree-inorder-traversal/problem>
<https://www.hackerrank.com/challenges/tree-preorder-traversal/problem>
<https://www.hackerrank.com/challenges/tree-postorder-traversal/problem>
3. Graph Traversal using Breadth First Search (BFS)
<https://www.hackerrank.com/challenges/bfsshortreach/problem>
4. Graph Traversal using Depth First Search (DFS)
<https://leetcode.com/problems/number-of-islands/description/>

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>

Additional Practice:

SNO	Difficulty	Problem Name	URL
1	Easy	Structs and enums	https://www.hackerrank.com/challenges/too-high-boxes/problem
2	Easy	Structs and enums	https://www.codechef.com/problems/HS08TEST?tab=statement
3	Medium	Structs and enums	https://www.hackerrank.com/challenges/small-triangles-large-triangles/problem
4	Easy	Arrays and Hashtables	https://leetcode.com/problems/two-sum/description/
5	Easy	Arrays and Pointers	https://leetcode.com/problems/remove-duplicates-from-sorted-array/description/?utm_source
6	Medium	Arrays and Pointers	https://leetcode.com/problems/two-sum-ii-input-array-is-sorted/description/?utm_source
7	Easy	Linked List	https://leetcode.com/problems/merge-two-sorted-lists/description/
8	Medium	Arrays and Matrix	https://leetcode.com/problems/diagonal-traverse/description/
9	Easy	Linked List	https://www.hackerrank.com/challenges/compare-two-linked-lists/problem
10	Medium	Stack applications	https://www.geeksforgeeks.org/problems/tower-of-hanoi-1587115621/1?utm
11	Medium	Binary Tree	https://leetcode.com/problems/binary-tree-right-side-view/description/
12	Medium	Stack applications	https://leetcode.com/problems/remove-k-digits/description/
13	Medium	Graph Theory	https://www.hackerrank.com/challenges/journey-to-the-moon/problem
14	Medium	Graph Traversals	https://leetcode.com/problems/number-of-provinces/description/
15	Medium	Graph Theory	https://www.hackerrank.com/challenges/torque-and-development/problem

	Basic Programs	Aditya's Internal coding Platform(Maya)	Code chef	Hacker rank	Leet code	Geek for Geeks
Unit-1	3	0	0	0	0	0
Unit-2	0	0	0	3	0	0
Unit-3	0	0	1	0	2	0
Unit-4	0	0	0	1	5	0
Unit-5	0	0	0	4	3	0
Additional Practice	0	0	1	5	8	1
Total	3	0	2	13	18	1

Business Intelligence Lab

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

Course Code: 2501IT01	L	T	P	C
	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 Modelling in Power BI**
 - a. Understanding relationships between tables
 - b. Creating calculated columns and measures using DAX

8. **Data Modelling 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 drilldown 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 drillthrough

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>

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

Course Code: 2501MA08	L	T	P	C
	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) :

1. 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):

1. Compute the n^{th} 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):

1. 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, Bernard 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/>

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

Course Code: 2501MA09	L	T	P	C
	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):

1. Calculation of Mean and Variance.

UNIT-II

Probability Distributions:

Distribution function –Binomial, Poisson, Normal and Exponential distributions.

Practice(Using any computational tool):

1. 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, Wolfram 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>

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

Course Code: 2501AI02	L	T	P	C
	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>

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

Course Code: 2501CS08	L	T	P	C
	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:

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

Coding Practice:

- Palindrome Number
Link: <https://leetcode.com/problems/palindrome-number/description/>
- First n Fibonacci

Link:<https://www.geeksforgeeks.org/problems/print-first-n-fibonacci-numbers1002/1?page=1&category=Dynamic%20Programming&sortBy=submissions>

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, Nested class. Introduction to Constructors and Destructor, Characteristics of Constructor and Destructor, Application with Constructor, Constructor with Arguments, Copy Constructor, Destructors, 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.

Coding Practice

1. Function With No Argument

Link:<https://www.geeksforgeeks.org/problems/function-with-no-arguments-1605703188/1?page=3&category=CPP&sortBy=submissions>

2. Sum of all array elements

Link:<https://www.geeksforgeeks.org/problems/sum-all-array-elements/1?page=4&category=CPP&sortBy=submissions>

3. Class

Link:<https://www.hackerrank.com/challenges/c-tutorial-class/problem?isFullScreen=true>

4. Class and Objects

Link: <https://www.hackerrank.com/challenges/classes-objects/problem?isFullScreen=true>

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.

Coding Practice:

1. Inheritance

Link: <https://www.hackerrank.com/challenges/inheritance-introduction/problem?isFullScreen=true>

2. Pointer

Link: <https://www.hackerrank.com/challenges/c-tutorial-pointer/problem?isFullScreen=true>

3. Virtual Functions

Link: <https://www.hackerrank.com/challenges/virtual-functions/problem?isFullScreen=true>

4. Operator Overloading

Link: <https://www.hackerrank.com/challenges/operator-overloading/problem?isFullScreen=true>

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.

Coding Practice:

1. C++ Class Templates

Link: <https://www.hackerrank.com/challenges/cpp-class-template-specialization/problem?isFullScreen=true>

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.

Coding Practice:

1. Exception Handling

Link: <https://www.hackerrank.com/challenges/cpp-exception-handling/problem?isFullScreen=true>

2. C++ STL(Vector)

Link: <https://www.geeksforgeeks.org/problems/c-stl-set-1-vector/1?page=5&category=CPP&sortBy=submissions>

3. Binary Search Tree Iterator

Link: <https://leetcode.com/problems/binary-search-tree-iterator/?envType=problem-list-v2&envId=iterator>

Text Books:

- 1 The Complete Reference C++, Herbert Schildt, TMH, Fourth Edition, ISBN-10. 0070411832.
- 2 The C++ Programming Language, Bjarne Stroustrup, Pearson, Fourth Edition, ISBN · 9780133522884.

Reference Books:

- 1 A First Book Of C++, Gary Bronson, Cengage Learning, Fourth Edition. ISBN-13. 978-8131532713
- 2 C++ Primer Plus By Stephen Prata, Pearson, Sixth Edition. ISBN: 978-0321776402
- 3 C++ Programming:From Problem Analysis To Program Design, Ds Malik , Cengage Learning, Eighth Edition. 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_01350157284861542410101
- 4 <https://www.udemy.com/course/beginning-c-plus-plus-programming/>

Additional Practice:

	Problem Name	Link
1	Hello World	https://www.hackerrank.com/challenges/cpp-hello-world/problem
2	Input & Output	https://www.hackerrank.com/challenges/cpp-input-and-output/problem
3	If-Else	https://www.hackerrank.com/challenges/c-tutorial-conditional-if-else/problem
4	For Loop	https://www.hackerrank.com/challenges/c-tutorial-for-loop/problem
5	Functions	https://www.hackerrank.com/challenges/c-tutorial-functions/problem
6	String Basics	https://www.hackerrank.com/challenges/c-tutorial-strings/problem
7	CamelCase	https://www.hackerrank.com/challenges/camelcase/problem
8	StringStream	https://www.hackerrank.com/challenges/c-tutorial-stringstream/problem
9	Array Introduction	https://www.hackerrank.com/challenges/arrays-introduction/problem
10	Vector Sort	https://www.hackerrank.com/challenges/vector-sort/problem
11	Vector Erase	https://www.hackerrank.com/challenges/vector-erase/problem
12	Sets	https://www.hackerrank.com/challenges/cpp-sets/problem
13	Maps	https://www.hackerrank.com/challenges/cpp-maps/problem
14	Deque	https://www.hackerrank.com/challenges/deque-stl/problem
15	Lower Bound	https://www.hackerrank.com/challenges/cpp-lower-bound/problem
16	Classes	https://www.hackerrank.com/challenges/c-tutorial-class/problem
17	Structs	https://www.hackerrank.com/challenges/c-tutorial-struct/problem
18	Virtual Functions	https://www.hackerrank.com/challenges/virtual-functions/problem
19	Inheritance Introduction	https://www.hackerrank.com/challenges/inheritance-introduction/problem
20	Pointer Basics	https://www.hackerrank.com/challenges/c-tutorial-pointer/problem
21	Intro to Sorting	https://www.hackerrank.com/challenges/tutorial-intro/problem
22	Mark and Toys	https://www.hackerrank.com/challenges/mark-and-toys/problem
23	Time Conversion	https://www.hackerrank.com/challenges/time-conversion/problem
24	Linked List Basics	https://www.hackerrank.com/challenges/print-the-elements-of-a-linked-list/problem
25	Tree Height	https://www.hackerrank.com/challenges/tree-height-of-a-binary-tree/problem
26	Interview Kit	https://www.hackerrank.com/interview/interview-preparation-kit
27	FLOW001 - Add Two Numbers	https://www.codechef.com/problems/FLOW001
28	FLOW002 - Find Remainder	https://www.codechef.com/problems/FLOW002

29	HS08TEST - ATM	https://www.codechef.com/problems/HS08TEST
30	INTEST - Input Test	https://www.codechef.com/problems/INTEST
31	FSQRT - Square Root	https://www.codechef.com/problems/FSQRT
32	AMR15A - Mahasena	https://www.codechef.com/problems/AMR15A
33	FLOW004 - Reverse Number	https://www.codechef.com/problems/FLOW004
34	LAPIN - Lapindrome	https://www.codechef.com/problems/LAPIN
35	MUFFINS3 - Muffins	https://www.codechef.com/problems/MUFFINS3
36	RECIPE - Recipe	https://www.codechef.com/problems/RECIPE
37	Two Sum	https://leetcode.com/problems/two-sum/
38	Palindrome Number	https://leetcode.com/problems/palindrome-number/
39	Roman to Integer	https://leetcode.com/problems/roman-to-integer/
40	Valid Parentheses	https://leetcode.com/problems/valid-parentheses/
41	Merge Two Sorted Lists	https://leetcode.com/problems/merge-two-sorted-lists/
42	Maximum Subarray	https://leetcode.com/problems/maximum-subarray/
43	Plus One	https://leetcode.com/problems/plus-one/
44	Climbing Stairs	https://leetcode.com/problems/climbing-stairs/
45	Best Time to Buy and Sell Stock	https://leetcode.com/problems/best-time-to-buy-and-sell-stock/
46	Binary Tree Inorder Traversal	https://leetcode.com/problems/binary-tree-inorder-traversal/

	Aditya's Internal Coding Platform (Maya)	Code chef/greek for greeks	Hacker rank	Leet code	Total
Unit-1	0	1	0	1	2
Unit-2	0	2	2	0	4
Unit-3	0	0	4	0	4
Unit-4	0	0	1	0	1
Unit-5	0	1	1	1	3
Additional Practice	0	10	26	10	46
Total	0	14	34	12	60

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

Course Code: 2501IT05

L	T	P	C
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	2									
CO2	2	3	2								
CO3	3		2								
CO4		2	3								
CO5			3	3	2						

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT – I

Introduction to DBMS: History and Architecture, Need and Purpose of DBMS, Applications of DBMS, Data and Metadata, Levels of abstraction, Three-Schema Architecture, structure of DBMS, Database Users & Roles of DBA, Basics of Parallel and Distributed Databases, Specialty Databases – Spatial, Temporal, Multimedia, NoSQL.

View of Data : Concepts of Schema, Data Models(Relational, ER, Object-based, Semi-structured), Steps in Database Design, Database system environment, Centralized and Client Server architecture for the database.

Practice

1. Familiarization with installation of DBMS(oracle 10g).
2. Draw the schema model for a University Database System by identifying the main entities using draw.io/diagrams.net or any similar tool.

UNIT – II

Introduction to the Relational Model: Structure of Relational Model, concepts of domain, attribute, tuple, relation, Database Schema and Schema Diagrams, Keys in Relational Model: Primary, Candidate, Alternate, Foreign, Importance and Handling of NULL Values, Relational Algebra: Fundamental Operations.

Basic SQL: Data types, table definitions (create, alter), DML operations (insert, delete, update), basic SQL querying (select, where clause, projection), arithmetic & logical operations, SQL functions, Introduction to GROUP BY and HAVING clauses.

Practice

1. Querying and modifying the database using Data Manipulation Language commands - select, insert, update, delete.
2. Implementation of SQL Functions – Use group-by and having clause.
 - Find Customer Referee using DML Operations
 - <https://leetcode.com/problems/find-customer-referee/description/?envType=study-plan-v2&envId=top-sql-50>
 - Average Time of Process per Machine using aggregate functions.
 - <https://leetcode.com/problems/average-time-of-process-per-machine/description/?envType=study-plan-v2&envId=top-sql-50>
 - DML Operations on given schema
 - <https://www.hackerrank.com/challenges/revising-the-select-query-2/problem?isFullScreen=true>
 - Aggregate Operators
 - <https://www.hackerrank.com/challenges/earnings-of-employees/problem?isFullScreen=true>
 - Basic Queries for salaries of employees
 - <https://www.hackerrank.com/challenges/salary-of-employees/problem?isFullScreen=true>
 - [/salary-of-employees/problem?isFullScreen=true](https://www.hackerrank.com/challenges/salary-of-employees/problem?isFullScreen=true)

UNIT – III

ER Model – Introduction to entities, attributes, entity sets, relationships, relationship sets, constraints. Specialization, generalization, inheritance, subclasses and super classes using ER diagrams.

Advanced SQL – Creating tables with relationships, implementing key and integrity constraints, nested and correlated subqueries, grouping, aggregation, ordering, different types of joins (natural, equi, outer, left, right, inner), views (updatable and non-updatable),

relational set operations.

Practice:

1. Perform join operations: natural, equi-join, outer join, left, right, inner; assess query performance.
 2. Perform set operations: union, intersection, difference.
 3. Implement nested and correlated subqueries.
 4. Creating and querying views and materialized views.
- Queries using join operations Ex: Product Sales Analysis I
 - <https://leetcode.com/problems/product-sales-analysis-i/description/?envType=study-plan-v2&envId=top-sql-50>
 - Queries using Nested queries. Ex: Employees Whose Manager Left the Company
 - <https://leetcode.com/problems/employees-whose-manager-left-the-company/description/?envType=study-plan-v2&envId=top-sql-50>
 - Queries using join operations
 - <https://leetcode.com/problems/primary-department-for-each-employee/description/?envType=study-plan-v2&envId=top-sql-50>
 - Queries challenge the pads problem
 - <https://www.hackerrank.com/challenges/the-pads/problem?isFullScreen=true>
 - Queries type of triangle problem
 - <https://www.hackerrank.com/challenges/what-type-of-triangle/problem?isFullScreen=true>
 - Joins
 - <https://www.hackerrank.com/challenges/the-report/problem?isFullScreen=true>

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** using the given university database structure. The schema consists of the following tables:
 - **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, savepoint, and rollback.
- Understanding Transaction, Commit and Rollback
 - <https://www.codechef.com/learn/course/sql/SQ00LS10A/problems/TRA01>
 - Queries on normalized database.
 - <https://www.hackerrank.com/challenges/symmetric-pairs/problem?isFullScreen=true>
 - Queries on occupations
 - <https://www.hackerrank.com/challenges/occupations/problem?isFullScreen=true>

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 Foundations Certificate Course offered by Oracle Academy at the end of the Practice Session.

Textbooks:

- 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 Database Systems, Ramez Elmasri, Shamkant B. Navathe, 7th Edition, Pearson, ISBN: 978-0137504277.
- 2 Database Systems, Carlos Coronel, Steven Morris, Peter Rob, 10th Edition, Cengage, ISBN: 978-0357517864.
- 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/

Additional Practice:

S.No	Difficulty	Problem Name	Link
1.	Easy	Select	https://www.codechef.com/learn/course/sql/SQ00LS04/problems/GSQ17
2.	Easy	DML queries	https://leetcode.com/problems/second-highest-salary/description/
3.	Easy	Select All	https://www.hackerrank.com/challenges/select-all-sql/problem?isFullScreen=true
4.	Easy	Basic employee names	https://www.hackerrank.com/challenges/name-of-employees/problem?isFullScreen=true
5.	Easy	Aggregate functions	https://www.hackerrank.com/challenges/revising-aggregations-sum/problem?isFullScreen=true
6.	Easy	Basic SQL Querying	https://www.codechef.com/learn/course/sql/SQ00LS04/problems/GSQ21
7.	Easy	Basic Join	https://www.hackerrank.com/challenges/the-report/problem?isFullScreen=true
8.	Easy	Basic Join	https://www.hackerrank.com/challenges/average-population-of-each-continent/problem?isFullScreen=true
9.	Easy	Views	https://leetcode.com/problems/article-views-i/
10.	Easy	Database Normalization	https://www.hackerrank.com/challenges/placements/problem?isFullScreen=true
11.	Easy	1NF	https://www.hackerrank.com/challenges/database-normalization-1-1nf/problem?isFullScreen=true
12.	Easy	1NF/2NF/3NF	https://www.hackerrank.com/challenges/database-normalization-123nf/problem?isFullScreen=true

13.	Easy	2NF/3NF	https://www.hackerrank.com/challenges/database-normalization-4/problem?isFullScreen=true
14.	Easy	Select	https://www.hackerrank.com/challenges/average-population-of-each-continent/problem?isFullScreen=true
15.	Easy	Select ID	https://www.hackerrank.com/challenges/select-by-id/problem?isFullScreen=true
16.	Easy	attibutes	https://www.hackerrank.com/challenges/japanese-cities-attributes/problem?isFullScreen=true
17.	Easy	Select	https://www.hackerrank.com/challenges/select-by-id/problem?isFullScreen=true
18.	Easy	Select	https://www.hackerrank.com/challenges/weather-observation-station-3/problem?isFullScreen=true
19.	Easy	Select	https://www.hackerrank.com/challenges/weather-observation-station-1/problem?isFullScreen=true
20.	Easy	Select	https://www.hackerrank.com/challenges/more-than-75-marks/problem?isFullScreen=true
21.	Easy	Indexing	https://leetcode.com/problems/create-a-new-column/description/
22.	Medium	BCNF	https://www.hackerrank.com/challenges/database-normalization-5/problem?isFullScreen=true
23.	Medium	4NF	https://www.hackerrank.com/challenges/database-normalization-10/problem?isFullScreen=true
24.	Medium	select	https://www.hackerrank.com/challenges/the-company/problem?isFullScreen=true
25.	Medium	joins	https://www.hackerrank.com/challenges/harry-potter-and-wands/problem?isFullScreen=true
26.	medium	Indexing	https://leetcode.com/problems/primary-department-for-each-employee/description/
27.	Medium	Group BY	https://leetcode.com/problems/product-price-at-a-given-date/description/
28.	Medium	Group BY	https://leetcode.com/problems/group-sold-products-by-the-date/description/
29.	Hard	Delete index	https://leetcode.com/problems/delete-duplicate-emails/description/
30.	Hard	Insert and update	https://leetcode.com/problems/trips-and-users/description/
31.	Hard	Insert and update	https://leetcode.com/problems/students-and-examinations/description/
32.	Hard	Delete index	https://leetcode.com/problems/immediate-food-delivery-ii/description/
33.	Hard	Views	https://www.hackerrank.com/challenges/interviews/problem?isFullScreen=true
34.	Hard	joins	https://www.hackerrank.com/challenges/15-days-of-learning-sql/problem?isFullScreen=true

	Aditya's Internal Coding Platform (Maya)	Code chef	Hacker rank	Leet code	Total
Unit-1	0	0	0	0	0
Unit-2	0	0	3	2	5
Unit-3	0	0	3	3	6
Unit-4	0	1	2	0	0
Unit-5	0	0	0	0	0
Additional Practice	0	2	22	10	34
Total	0	3	30	15	45

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

Course Code: 2501AI03	L	T	P	C
	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.

Practice:

1. Illustrate KDD process and explore the Data Mining tool WEKA/KNIME for Data Preprocessing, Classification and Clustering.
2. Explain ETL process important for moving data in the data warehouse.

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 on contact lenses. 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.

Cornerstone Project:

“Student Activity & Trend Detection using Data Mining (Library, Attendance, Events, Feedback)”

Introduction

This project focuses on identifying patterns and trends in student activities using data mining techniques. The analysis integrates multiple datasets such as Library Usage, Attendance, Event Participation, and Student Feedback.

Dataset Details

You will work with four primary datasets. Each dataset can be downloaded from publicly available sources such as Kaggle. Below are the complete details:

Project 1: Library Dataset

- Fields: StudentID, BookID, BorrowDate, ReturnDate, Category
- Purpose: Measure academic involvement through book borrowing patterns.
- Download: Kaggle – Book Recommendation or Library Management datasets.

1. Book Recommendation Dataset (contains user–book interactions)

 <https://www.kaggle.com/datasets/mohamedharris/book-recommendation-dataset>

2. Library Management System Dataset

 <https://www.kaggle.com/datasets/saurabhbagchi/books-dataset>

3. Goodreads Book Borrow/User Activity Dataset

 <https://www.kaggle.com/datasets/gnanesh/goodreads-books-dataset>

Project 2: Attendance Dataset

- Fields: StudentID, AttendancePercentage, Class, Section, Subject
- Purpose: Identify correlation between attendance and performance.
- Download: Kaggle – Student Performance/Attendance datasets.

1. Student Performance & Attendance Dataset

<https://www.kaggle.com/datasets/rodsaldanha/student-performance-data>

2. Students Exam Performance (contains attendance proxy + scores)

<https://www.kaggle.com/datasets/spscientist/students-performance-in-exams>

3. Student Academic Performance Dataset (includes attendance)

<https://www.kaggle.com/datasets/whenamancodes/students-performance>

Project 3: Events Participation Dataset

- Fields: StudentID, EventName, ParticipationType, CertificatesWon, EventCategory

- Purpose: Analyze co-curricular engagement.

- Download: Kaggle – Student activities & performance datasets.

1. Student Activity & Performance Dataset

<https://www.kaggle.com/datasets/ardianumam/student-activity-and-performance>

2. Extra-curricular Activities & Participation Dataset

<https://www.kaggle.com/datasets/lokhman/extra-curricular-activities-students>

3. Co-Curricular Activity Participation Dataset

<https://www.kaggle.com/datasets/abhinavmoudgil95/student-activity-data>

Project 4: Student Feedback Dataset

- Fields: StudentID, CourseRating, TeacherRating, Comment, Sentiment

- Purpose: Extract student satisfaction trends (text + numeric feedback).

- Download: Kaggle – Feedback/Evaluations datasets.

1. Student Course Evaluation Dataset

<https://www.kaggle.com/datasets/mateusdmachado/students-course-evaluations>

2. Student Satisfaction Survey (Feedback)

<https://www.kaggle.com/datasets/ananthu017/student-feedback-survey>

3. Teaching Evaluation Dataset (Large Text Feedback)

<https://www.kaggle.com/datasets/xplenty/academic-success-with-student-evaluations>

Student can select any one of the above and implement the capstone project using the following steps.

I. Data Preprocessing

1. Cleaning:

- Remove missing values.
- Normalize date formats (BorrowDate, ReturnDate).
- Remove duplicate entries.
- Convert categorical values (Department, Event Category).

2. Feature Engineering:

- LibraryScore = Total books borrowed per semester.
- AttendanceScore = Attendance percentage.
- EventScore = Count of events attended + certificates.
- FeedbackSentiment = Polarity score extracted using NLP.

3. Dataset Integration:

- Merge all datasets using StudentID as the primary key.
- Generate consolidated activity metrics for each student.

II. Model Implementation Steps

A. Clustering (Trend Detection)

- Algorithms: K-Means, DBSCAN.
- Input Features: LibraryScore, AttendanceScore, EventScore, FeedbackSentiment.
- Output: Activity clusters such as High, Medium, Low engagement.

B. Classification

- Algorithms: Decision Tree, Random Forest, Naïve Bayes.
- Target: Predict student performance or engagement levels.

C. Topic Modeling (Feedback)

- Algorithm: LDA
- Purpose: Extract themes from feedback comments such as:
 - Teaching Quality
 - Course Difficulty
 - Infrastructure
 - Stress Factors

III. Data Visualization Steps

Charts to be produced after analysis:

- Heatmap: Attendance vs CGPA
- Bar Chart: Library usage by department
- Pie Chart: Event participation percentages
- Line Chart: Monthly book borrowing trend
- Word Cloud: Feedback comments

Tools: Matplotlib, Seaborn, Plotly, Power BI.

IV. Insights & Interpretation

Based on combined analysis:

- Students with high library usage often have higher attendance and performance.
- Event participation is highest among CSE and AIML departments.
- Feedback sentiment indicates improvement needed in lab infrastructure.
- Attendance dips during mid-semester months but library usage peaks.

These insights help universities in academic planning and student support.

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/>

Note:

- The capstone project team size shall be 3 or 4 students.
- Students may select any one of the above prescribed projects or a project of their own choice with the prior permission of the Course instructor.
- The CIE – DDA & Internal Lab shall be evaluated for 25 marks based on project implementation, oral presentation, 10-minute video presentation, report and viva voce.
- The video presentation should consists of the working procedure of the project along with contribution of each student for a minimum of 2 minutes.
- Finally, the Source code of the capstone Project has to be pushed into the Students GitHub repository.

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

Course Code: 2501IT06	L	T	P	C
	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.
https://leetcode.com/problems/count-primes/description/?utm_source
- b. Find the sum of all even terms in the Fibonacci sequence up to the given range N.
<https://leetcode.com/problems/fibonacci-number/description/>
- c. Check whether a given number is Armstrong or not.
<https://www.hackerrank.com/contests/iiitdm-contest1/challenges/armstrong-number-2>

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

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

2. Strings

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

<https://leetcode.com/problems/valid-anagram/description/>

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

<https://www.hackerrank.com/challenges/java-inheritance-1/problem>

<https://www.hackerrank.com/challenges/java-interface/problem>

- Implement multi level Inheritance.
- Implement multiple Inheritance.

<https://www.hackerrank.com/challenges/java-interface/problem>

- Find the areas of different shapes using abstract classes.

<https://www.hackerrank.com/challenges/java-abstract-class/problem>

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, BufferedReader and BufferedWriter, Wrapper Classes.

Practice:**1. Packages & java.io**

a. Import and use user defined package.

<https://www.geeksforgeeks.org/java/packages-in-java/>

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.

Cornerstone Project:**PROJECT 1: Student Attendance Management System****Project Implementation, Documentation and Viva Voce**

- **Implementation mainly focuses on**

- Adding student details, marking and viewing attendance.
- JDBC operations: insert, update, search by roll number.
- Input validation (no duplicate roll numbers, days \leq total days).
- Attendance percentage calculation.

- Menu-driven console workflow with exception handling.
- **Documentation must contain**
 - Objective & problem definition
 - Attendance table structure
 - ER diagram (Student → Attendance)
 - Module explanations: Add, Mark, View, Search
 - JDBC connection flow
 - Screenshots of sample input/output
 - Enhancements added (login, validation, normalization)

PROJECT 2: Library Management System

- **Implementation mainly focuses on**
 - Adding, searching, updating, deleting book records.
 - JDBC CRUD operations on books table.
 - Validation of price, quantity.
 - Implementing search by ID, title, and author.
- **Documentation must contain**
 - Objective of library automation
 - Books table schema
 - Flowchart for Add → View → Delete operations
 - Module descriptions
 - Exception handling (invalid ID)
 - Screenshots
 - Enhancements (issue records, author search)

PROJECT 3: Student Course Registration System

Implementation Mainly Focuses On

- Course registration, view, update, delete operations.
- Database design with student, course, registration tables.
- Validation for duplicate registration.
- JDBC connectivity and table relationships.

Documentation Must Contain

- Objective
- Table design (student, course, registration)
- ER diagram
- Module explanation: Register, Display, Edit
- SQL constraints used
- Screenshots of registration process

PROJECT 4: Online Student Result Management System

Implementation Mainly Focuses On

- Marks entry and validation.
- Automatic total, percentage, grade calculation.
- Result storage and retrieval using JDBC.
- Web-based form input using HTML.

Documentation Must Contain

- Objective
- Marks table schema
- Grade calculation logic

- HTML form screenshots
- JDBC code explanation
- Enhancements (validation, grade generation)

PROJECT 5: Employee Leave Management System

Implementation Mainly Focuses On

- Leave application submission.
- Balance update logic.
- Validations on leave days and employee ID.
- JDBC-based leave approval or rejection.

Documentation Must Contain

- Objective
- Employee & leave tables
- Flowchart: Apply → Validate → Approve
- Module explanation
- Screenshots of leave form and output

PROJECT 6: Online Shopping Cart Simulation

Implementation Mainly Focuses On

- Displaying products and adding items to cart.
- Stock validation before adding to cart.
- Bill calculation with discounts/coupons.
- JDBC for storing cart and order history.

Documentation Must Contain

- Objective
- Product, cart, order tables
- Module flow: Add → Update → Bill
- Billing logic with formulas
- Screenshots of cart process

PROJECT 7: Bus Ticket Booking System

Implementation Mainly Focuses On

- Booking, cancelling, viewing seat availability.
- Preventing double booking of same seat.
- Route table and seat allocation logic.
- JDBC insert/update operations.

Documentation Must Contain

- Objective
- Route and seat tables
- Seat allocation flowchart
- Module descriptions
- SQL queries
- Screenshots of booking/cancel pages

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/106105191/>
- 2 <http://java.sun.com/docs/books/tutorial/>

Note:

- Students must submit the Certificate of Completion offered by any Industry.
- The cornerstone project team size shall be four (4) students.
- Students may select any one of the above prescribed projects or a project of their own choice with the prior permission of the Course instructor.
- The SEE – Lab shall be evaluated for 50 marks based on project implementation, oral presentation, 10-minute video presentation, report and viva voce.
- The video presentation should consist of the working procedure of the project along with contribution of each student for a minimum of 2 minutes.
- Finally, the Source code of the cornerstone Project has to be pushed into the Students GitHub repository.

Additional Practice:

UNIT-I		
1	Finding 3-Digit Even Numbers	Link:https://leetcode.com/problems/finding-3-digit-even-numbers/description/
2	Loops	Link: https://www.hackerrank.com/challenges/30-loops/problem
3	Unique 3-Digit Even Numbers	Link:https://leetcode.com/problems/unique-3-digit-even-numbers/description/
4	Best Time to Buy and Sell Stock	Link:https://leetcode.com/problems/best-time-to-buy-and-sell-stock/description/
5	Single Number	Link: https://leetcode.com/problems/single-number/description/
6	Majority Element	Link: https://leetcode.com/problems/majority-element/description/
7	Contains Duplicate	Link: https://leetcode.com/problems/contains-duplicate/description/
8	Missing Number	Link: https://leetcode.com/problems/missing-number/description/

9	Second Largest Number	Link:https://www.naukri.com/code360/problem-details/second-largest-number_13762781
10	Java Program to Find if a Given Year is a Leap Year	Link:https://www.geeksforgeeks.org/java/java-program-to-find-if-a-given-year-is-a-leap-year/
11	Bitwise Operators	Link:https://www.geeksforgeeks.org/problems/bitwise-operators-1597314674/0
12	Program to Convert Octal Number to Binary Number	Link:https://www.geeksforgeeks.org/dsa/program-to-convert-octal-number-to-binary-number/
13	Neon Number	Link: https://www.geeksforgeeks.org/dsa/neon-number/
14	Automorphic Number	Link: https://www.geeksforgeeks.org/dsa/automorphic-number/
15	Harshad (Or Niven) Number	Link: https://www.geeksforgeeks.org/dsa/harshad-or-niven-number/
UNIT-II		
16	Longest Common Prefix	Link: https://leetcode.com/problems/longest-common-prefix/description/
17	Find the index of the first occurrence in a string	Link:https://leetcode.com/problems/find-the-index-of-the-first-occurrence-in-a-string/description/
18	Valid-Parentheses	Link: https://leetcode.com/problems/valid-parentheses/description/
19	Length of last word	Link: https://leetcode.com/problems/length-of-last-word/description/
20	Add Binary	Link: https://leetcode.com/problems/add-binary/description/
21	Isomorphic Strings	Link: https://leetcode.com/problems/isomorphic-strings/description/
22	Jewels and Stones	Link: https://leetcode.com/problems/jewels-and-stones/description/
23	Buddy Strings	Link: https://leetcode.com/problems/buddy-strings/description/
24	Reverse Only Letters	Link: https://leetcode.com/problems/reverse-only-letters/description/
25	Find Common Characters	Link: https://leetcode.com/problems/find-common-characters/description/
26	Find Pattern	Link: https://www.geeksforgeeks.org/problems/find-pattern/0
27	Largest Substring Between Two Equal Characters	Link:https://leetcode.com/problems/largest-substring-between-two-equal-characters/description/
28	Remove All Adjacent Duplicates In String	Link:https://leetcode.com/problems/remove-all-adjacent-duplicates-in-string/description
29	Permutations of given string	Link:https://www.geeksforgeeks.org/dsa/write-a-c-program-to-print-all-permutations-of-a-given-string

30	Longest Repeating Subsequence	Link: https://www.geeksforgeeks.org/dsa/longest-repeating-subsequence/
UNIT-III		
31	Java Inheritance-1	Link: https://www.hackerrank.com/challenges/java-inheritance-1/problem
32	Java Inheritance-2	Link: https://www.hackerrank.com/challenges/java-inheritance-2/problem
33	Java Abstract Class	Link: https://www.hackerrank.com/challenges/java-abstract-class/problem
34	Java Interface	Link: https://www.hackerrank.com/challenges/java-interface/problem
35	Java Method Overriding	Link: https://www.hackerrank.com/challenges/java-method-overriding/problem
36	Java Method Overriding-2 (super keyword)	Link: https://www.hackerrank.com/challenges/java-method-overriding-2-super-keyword/problem
UNIT-IV		
37	Java program to append a string in an existing file	Link: https://www.geeksforgeeks.org/java/java-program-to-append-a-string-in-an-existing-file/
38	Java End-of-file	Link: https://www.hackerrank.com/challenges/java-end-of-file/problem
39	Java Stdin and Stdout II	Link: https://www.hackerrank.com/challenges/java-stdin-stdout/problem
40	Java Output Formatting	Link: https://www.hackerrank.com/challenges/java-output-formatting/problem
41	Find Duplicate File in System	Link: https://leetcode.com/problems/find-duplicate-file-in-system/description/
UNIT-V		
42	Java Exception Handling (Try-catch)	Link: https://www.hackerrank.com/challenges/java-exception-handling-try-catch
43	Java Exception Handling	Link: https://www.hackerrank.com/challenges/java-exception-handling
44	Java Factory Pattern	Link: https://www.hackerrank.com/challenges/java-factory
45	Print in Order	Link: https://leetcode.com/problems/print-in-order/description/
46	Print Zero Even Odd	Link: https://leetcode.com/problems/print-zero-even-odd/description/

	Aditya's Internal Coding Platform (Maya)	Code chef/greek for greeks	Hacker rank	Leet code	Total
Unit-1	0	0	1	2	3
Unit-2	0	0	0	1	1
Unit-3	0	0	4	0	4
Unit-4	0	1	0	0	1
Unit-5	0	0	0	0	0
Additional Practice	0	10	13	23	46
Total	0	11	18	26	55

Language Processors
(Common to AIML & CSE(DS))

Course Code: 2501CS09	L	T	P	C
	2	1	0	3

Course Outcomes:

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

- CO1:** Explain the equivalence between NFA, DFA and Regular Expressions
- CO2:** Illustrate the concepts of Context Free Grammars, Push Down Automata and Turing Machines
- CO3:** Analyze the phases in compilation process.
- CO4:** Apply different parsing algorithms to develop parsers for a given grammar
- CO5:** Discuss code generation and code optimization techniques

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	-	-	1	-	-	-	-	-	-	-
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 and Regular Expressions: Formal Languages and Regular expressions, Deterministic and Non Deterministic Finite Automata, Finite Automata with ϵ moves, Equivalence of NFA and DFA, Minimization of Finite Automata, Applications of Finite Automata.

UNIT-II

Grammars , PDA and Turing Machines: Chomsky hierarchy, Properties of regular sets, Pumping Lemma for regular languages, ContextFree Grammars – Derivation trees, Ambiguous and unambiguous grammars , Left recursion, Left factoring, Pushdown Automata and ContextFree Languages, Turing machines (TM) – Turing Machine constructions – Variations of TMs.

UNIT-III

Phases of Compiler and Lexical Analyzer : Compilers Analysis of the source program The phases of a compiler Cousins of the compiler Compiler construction tools Lexical Analysis The role of the lexical analyzer Input buffering Specification of tokens Recognition of tokens A language for specifying lexical analyzers Design of a lexical analyzer.

UNIT-IV

Syntax Analysis and Syntax Directed Translation: The role of the parser Context free grammars Top down parsing Bottom up parsing Operator precedence parsing – automatic construction of efficient parser – predictive parser LR parsers Parser generators. Syntax directed definitions Construction of syntax trees Bottom up evaluation of Sattributed definitions Lattributed definitions Analysis of syntax directed definitions.

UNIT – V

Intermediate Code Generation and Code Generation: Intermediate languages Declarations Assignment statements Boolean expressions Back patching Procedure calls. Issues in the design of a code generator Runtime storage management Basic blocks and flow graphs The DAG representation of Basic Block Next use information simple code generator Register allocation and assignment –Code Optimization Peephole optimization Generating code from DAGs.

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 Compilers Principles, Techniques And Tools, Alfred V Aho, J. D. Ullman, Ravi Sethi, PEA, Second Edition, 2007, ISBN: 978-9357054119

Reference Books:

- 1 Theory of Computer Science-Automata, Languages and Computation, K. L. P. Mishra and N. Chandrasekharan, PHI, 3rd Edition, ISBN: 978-8120329683
- 2 Elements of Theory of Computation, Lewis H.P. & Papadimitriou C.H., Pearson /PHI, ISBN: 978-0132624787
- 3 Theory of Automata, Languages and Computation, Rajendra Kumar, McGraw Hill, ISBN: 978-0070702042
- 4 Compiler Construction-Principles and Practice, Kenneth C Loudon, Cengage Learning, ISBN: 978-0534939724
- 5 The Compiler Design Handbook, Y.N. Srikant and Priti Shankar, CRC Press, 2nd ed, ISBN: 978-1420043822

Web Links:

- 1 <http://nptel.ac.in/courses/111103016>
- 2 <https://nptel.ac.in/courses/106/104/106104148/>
- 3 <https://nptel.ac.in/courses/106/108/106108113/>

- 4 <https://nptel.ac.in/courses/106/104/106104123/>
- 5 <https://nptel.ac.in/courses/106/104/106104072/>

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

Course Code: 2501IT07	L	T	P	C
	2	0	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 Capstone 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, Pearson, Ninth Edition, ISBN: 978-9332582699.
- 2 Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results, Prentice Hall, David J. Anderson; Eli Schragenheim, ISBN: 978-0131424609

Web Links:

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

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

Course Code: 2501CS13	L	T	P	C
	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	1	-	-
CO2	1	3	-	1	-	-	-	2	2	-	1
CO3	1	3	-	1	-	-	-	1	2	-	1
CO4	1	3	-	1	-	-	-	1	1	-	1
CO5	2	2	-	-	-	-	-	1	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, OpenSource Operating Systems.

System Structures: Operating System Services, User and OperatingSystem 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. Multi-programming Memory management Implementation of fork (), wait (), exec() and exit (), System calls

UNIT-II

Process Concept: Process scheduling, Operations on processes, Interprocess 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:(a) FCFS (b) SJF
2. Simulate the following CPU scheduling algorithms:(a) Priority (b) Round Robin

UNIT-III

Interprocess 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 onwrite, Page replacement, Frame allocation, Thrashing, Memorymapped 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 (a) Sequenced (b) Indexed (c) 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 Operating Systems, William Stallings, Internals and Design Principles. Prentice-Hall, 7th Ed, 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 Education, 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 <http://www.geeksforgeeks.org/operating-systems/>
- 5 <https://in.udacity.com/auth?next=/course/introduction-to-operating-systems--ud923>

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

Course Code: 2501CS07	L	T	P	C
	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/PSO	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 framing 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 policies, Approaches to Congestion Control, Traffic Aware Routing, Admission Control, Traffic Throttling, Load Shedding.

Traffic Control Algorithms: 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>

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

Course Code: 2501AI05	L	T	P	C
	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	PO 3	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 Machine Learning. Statistical Learning: Introduction, Supervised and Unsupervised Learning, The 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 the 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 and wrong 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 heart patients.

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 to perform 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 k Means 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>
- 3 https://onlinecourses.nptel.ac.in/noc21_cs24/preview

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>

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

Course Code: 2501AI11	L	T	P	C
	2	0	2	4

Course Outcomes:

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

- CO1:** Demonstrate the basic statistical concepts in deep learning.
- CO2:** Explain architecture and mathematical foundation for various deep neural networks.
- CO3:** Illustrate the challenges and optimization strategies in Deep Learning.
- CO4:** Build a convolutional neural network using different activation functions.
- CO5:** Build and train RNN and LSTMs using sequence modelling.

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	-	3
CO2	-	2	1	-	2	-	-	1	1	-	3
CO3	-	2	-	3	1	-	-	1	1	-	1
CO4	2	2	3	-	1	-	-	1	1	-	-
CO5	1	2	3	-	1	-	-	1	1	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT-I

DEEP LEARNING CONCEPTS: Fundamentals about Deep Learning, Perception Learning Algorithms, Probabilistic modelling, Early Neural Networks, How Deep Learning different from Machine Learning, Scalars, Vectors, Matrixes, Higher Dimensional Tensors, Manipulating Tensors. Vector Data, Hyper parameters Vs Parameters - validation sets - Estimators, Bias, Variance, Overfitting and Underfitting, Introduction to Keras and TensorFlow and PyTorch.

Practice:

1. Installation and basic commands of scikit, TensorFlow and PyTorch
2. Implement Random Forest by using scikit, TensorFlow and PyTorch
3. Implement multilayer perceptron algorithm for MNIST Handwritten Digit Classification

UNIT-II

NEURAL NETWORKS : About Neural Network, Building Blocks of Neural Network, Introduction to Neural Networks, Feed-forward Networks, Deep Feed-forward Networks - Learning XOR, Gradient Based learning, Hidden Units, Back-propagation and other Differential Algorithms, Data Pre-processing for neural networks, Feature Engineering.

Practice:

1. Design a neural network for classifying movie reviews (Binary Classification) using IMDB dataset.
2. a) Design a neural Network for classifying news wires (Multi class classification) using Reuters dataset
b) Design a neural network for predicting house prices using Boston Housing Price dataset.

UNIT-III

OPTIMIZATION TECHNIQUES: Optimizers – SGD (Stochastic Gradient Descent), Mini-batch Gradient Descent, MS prop (Root Mean Square Propagation), Adagrad (Adaptive Gradient Algorithm), Adadelta (Adaptive Delta), Adam (Adaptive Moment Estimation), Adamax (Adaptive Moment Estimation with Infinity Norm).

Activation Functions : sigmoid, tanh, ReLU, leaky ReLU, Soft Max, Linear Loss Functions - cross-entropy loss, the mean-squared error, the Huber loss, the hinge loss, L2 loss, mean absolute loss Regularization – L1, L2 and drop out. Learning Rate, Normalization

Practice:

1. Implement word embeddings for IMDB dataset.
2. Implement one hot encoding of words or characters.
3. Consider Patient Dataset. Apply linear classification technique(SVM) to identify the rate of heart patients. Also apply the optimization techniques.
4. Write a Python program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set

UNIT-IV

CONVOLUTIONAL NEURAL NETWORK : About CNN, Building a convolutional neural network, Input Layers, Convolution Layers, Pooling Layers, Dense Layers, Backpropagation Through the Convolutional Layer, Filters and Feature Maps, Backpropagation Through the Pooling Layers, Dropout Layers and Regularization, Batch Normalization

Pre trained CNN: LeNet, Alex Net, VGG16, ResNet.

Practice:

1. Build a Convolution Neural Network for MNIST Handwritten Digit Classification.
2. a) Use a pre-trained convolution neural network (VGG16) for image classification
b) Build a Convolution Neural Network for simple image (dogs and Cats) Classification
3. Student Portfolio Creation using GitHub, Hugging Face & Kaggle

UNIT – V

RNN : Recurrent Neural Networks ,Sequence-to-Sequence Modelling – Embedding - Recurrent Neural Networks - Bidirectional RNNs, Analysing Variable Length Inputs – Tackling seq2seq Problem – Beam Search and Global Normalization – Recurrent Neural Networks (RNN)– Hidden States – Perplexity – Character-level Language Models –Modern RNNs: Gated Recurrent Units (GRU), Long Short-Term Memory (LSTM).

Practice:

1. Implement a Recurrent Neural Network for IMDB movie review classification problem.
2. Implement Boosting algorithms using scikit, Tensorflow and PyTorch

Additional Practice:

1. Assuming a set of documents that need to be classified, use the naïve Bayesian Classifier model to perform 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 k-Means 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:

Face Mask Detection Using Convolutional Neural Networks

Objective: Detect whether a person in an image is wearing a mask or not using real-time webcam input.

Domain: Healthcare / Public Safety

Techniques & Concepts Used:

- * Image Classification (CNNs – using TensorFlow/Keras)
- * Data Augmentation & Preprocessing
- * Transfer Learning (e.g., MobileNetV2)
- * Real-time camera integration using OpenCV

Text Books:

- 1 Deep Learning, Ian Goodfellow, Yoshua Bengio, Aaron Courville, MIT Press, ISBN: 9780262035613
- 2 Deep learning: A practitioner's approach, Josh Patterson and Adam Gibson, O'Reilly Media, First Edition, ISBN: 978-1491914250

Reference Books:

- 1 Fundamentals of Deep Learning, Designing next generation machine intelligence algorithms, Nikhil Buduma, O'Reilly, Shroff Publishers, ISBN: 978-9352135608
- 2 Deep learning Cookbook, Practical recipes to get started Quickly, Douwe Osinga, O'Reilly, Shroff Publishers, ISBN: 978-9352137572

Web Links:

- 1 https://onlinecourses.nptel.ac.in/noc20_cs62/preview
- 2 <https://keras.io/datasets/>

Natural Language Processing & Prompt Engineering

	L	T	P	C
Course Code: 2501AI20	1	0	2	3

Course Outcomes:

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

- CO1:** Explain Text Normalization and utilize Ngram language models for various NLP tasks.
- CO2:** Explain Text preprocessing and stemming algorithms.
- CO3:** Apply similarity measures and word embedding models
- CO4:** Summarize morphology and NLP applications
- CO5:** Explain prompt engineering and its 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	2	1	-	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	-	1
CO5	2	2	3	-	1	-	-	1	1	-	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

Introduction: Introduction to Natural Language Processing Overview of NLP Introduction to Levels of NLP –Regular Expressions, Text Normalization, Edit Distance Language Modelling: Ngram Language Models.

Practice:

1. Word Analysis Learn about morphological features of a word by analysing it
2. Word Generation generate word forms from root and suffix information

UNIT-II

Text Processing: Tokenization, Stemming, Text Preprocessing & Feature Representation Introduction to Corpora, Sentence Segmentation, Stemming: Porter Stemmer, Bag of words and Vector Space Model, Topic Modelling, Ngram Language Model, Smoothing,

Practice:

1. Implement Porter Stemming Algorithm
2. NGrams learn to calculate bigrams from a given corpus and calculate probability of a sentence

UNIT-III

Vector Semantics and Embeddings: Lexical Semantics, Vector, Words and Vectors, Cosine for measuring similarity, TFIDF: Weighing terms in the vector, Pointwise Mutual Information (PMI) , Applications of the tfidf or PPMI vector models , . Word Embeddings: Word2Vec, Glove and Fasttext, Visualizing Embeddings, Semantic properties of embeddings , Bias and Embeddings , Evaluating VectorModels .

Practice:

1. Apply term frequency"(TFIDF) to Measures the importance of a word to a specific document
2. Implement word embedding using word2vec

UNIT-IV

Morphology: Sequence Labelling for Parts of Speech and Named Entities, PartofSpeech Tagging, Named Entities and Named Entity Tagging , HMM PartofSpeech Tagging , Conditional Random Fields (CRFs), Evaluation of Named Entity Recognition Applications of NLP Machine Translation Encoder & Decoder Model, Attention Models, Question Answering Knowledge based Q&A Chatbots & Dialogue Systems, Automatic Speech Recognition and Text to Speech.

Practice:

1. Morphology understanding the morphology of a word by the use of Add Delete table.
2. POS Tagging – calculate emission and transition matrix which will be helpful for tagging Parts of Speech using Hidden Markov Model.

UNIT – V

Introduction to Prompt Engineering : What is prompt engineering and why it matters

Prompt types: explicit, implicit, and creative prompts, Best Practices for Crafting Effective Prompts.

Practical Applications of Prompt Engineering : Improving NLP Tasks with Custom Prompts, Enhancing Creativity and Diversity in AI Generated Content, Addressing AI Ethics and Bias through Thoughtful Prompt Engineering, Personalization and Adaptability in AI Generated Content.

Future Directions and Emerging Trends in Prompt Engineering : Leveraging Advanced AI Models and Techniques, The Convergence of Human and AI Creativity, The Role of Prompt Engineering in the AI Driven Economy

Practice:

1. Chunking understand the concept of chunking and get familiar with the basic chunk tagset.
2. Building Chunker the importance of selecting proper features for training a model and size of training corpus in learning how to do cunking.

Additional Practice:

1. Text normalization is a crucial process in natural language processing (NLP) that involves cleaning and preparing text data to ensure consistency and usability across different NLP tasks. By standardizing text, we reduce data variability, making it easier for NLP models to identify patterns and generate accurate predictions.
Apply text normalization to the following text
Jaron Lanier said: “It would be unfair to demand that people cease pirating files when those same people aren’t paid for their participation in very lucrative network schemes. Ordinary people are relentlessly spied on, and not compensated for information taken from them. While I’d like to see everyone eventually pay for music and the like, I’d not ask for it until there’s reciprocity.”
2. Demonstrate text smoothing using Laplace Smoothing (Additive Smoothing) for a simple bigram language model.
3. For the give documents data Interpret the documents that are closely related or similar in content, apply TFIDF and Cosine Similarity
Doc1 = "The quick brown fox jumps over the lazy dog"
Doc2 = "Never jump over the lazy dog quickly"
Doc3 = "Bright blue fox jumps quickly"
Doc4 = "The fox is quick and the dog is lazy"

Capstone Project

Sentiment Analysis and Emotion Detection in Tweets or Product Reviews

Objective: Analyze social media or review texts to detect sentiment (positive/negative) and optionally the emotion (joy, anger, etc.).

Domain: Social Media / E-commerce / Healthcare Communication

Techniques & Concepts Used:

- Text Preprocessing and Tokenization (Unit II)
- Word Embeddings (Word2Vec/FastText) (Unit III)
- Sequence Labeling for emotion classes (Unit IV)
- Prompt Engineering (for custom sentiment questions) (Unit V)

Text Books:

- 1 Speech and Language Processing, Dan Jurafsky, James H. Martin Draft of 3rd Edition, Prentice Hall 2022, ISBN: 978-0130950697
- 2 Springer Handbook of Speech Processing, Jacob Benesty, M. M. Sondhi, Yiteng Huang Springer, ISBN: 978-3662533000

Reference Books:

- 1 Speech and Language Processing, Daniel Jurafsky and James H. Martin. 2020. 3rd Edition (draft), ISBN: 978-0135041963
- 2 Foundations of Statistical Natural Language Processing, Christopher, D. Manning and Hinrich Schütze, MIT Press, ISBN: 978-0262133609

Web Links:

- 1 <https://www.nltk.org/data.html>
- 2 <https://www.analyticsvidhya.com/blog/2017/01/ultimateguidetounderstandimplement>

- 3 `naturallanguageprocessingcodesinpython/`
https://datahack.analyticsvidhya.com/contest/linguipediacodefestnaturallanguageprocessing1/?utm_source=ultimateguidetounderstandimplementnaturallanguageprocessingcodesinpython&utm_medium=blog

Reinforcement Learning & Gen AI

Course Code: 2501AI21	L	T	P	C
	1	0	2	3

Course Outcomes:

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

- CO1:** Outline the basic concepts of Reinforcement learning.
- CO2:** Identify the appropriate learning tasks for Reinforcement learning techniques.
- CO3:** Implement adversarial training techniques to train GAN models.
- CO4:** Analyse the applications of VAEs in various domains such as image generation, anomaly detection, and data imputation.
- CO5:** Evaluate the ethical considerations and societal impact of reinforcement learning and generative AI technologies.

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	-	-
CO2	2	2	-	-	2	-	-	-	1	-	-
CO3	3	2	-	-	2	-	-	-	1	-	-
CO4	3	2	1	-	2	-	-	-	1	-	-
CO5	2	1	1	-	2	-	-	-	1	-	-

Mapping of Course Outcomes with Program Specific Outcomes:

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

UNIT-I

Basics of reinforcement learning (RL), RL components: agents, environments, rewards
 Markov Decision Processes (MDPs), Exploration vs. Exploitation dilemma
 Basic algorithms: Qlearning, SARSA, MonteCarlo (MC) Learning, TemporalDifference (TD) Learning, TDLambda and Eligibility Traces.

Practice:

1. Installation of TensorFlow and implement the basic programs.
2. Installation of PYTORCH and implement the basic programs.

UNIT-II

Advanced Reinforcement Learning Techniques: Deep QNetworks (DQN), Policy Gradient methods, ActorCritic architectures, Advantage ActorCritic (A2C) and Proximal Policy Optimization (PPO), Continuous action spaces and algorithms like Deep Deterministic

Policy Gradient (DDPG).

Practice:

1. Implement a simple grid world environment where an agent learns to navigate from a starting point to a goal while avoiding obstacles.
2. Train an agent to balance a pole on a cart using discrete actions such as pushing the cart left or right.

UNIT-III

Generative Adversarial Networks (GANs): Introduction to generative models, Basics of GANs: generator, discriminator ,Training GANs: adversarial training ,Variants of GANs: Conditional GANs, Wasserstein GANs, etc, Applications of GANs in image generation, style transfer, and data augmentation.

Practice:

1. Model an agent learning to drive a car up a steep mountain by applying the correct amount of throttle and braking.
2. Train an agent to play the classic game of Pong, where it learns to control a paddle to hit a ball back to the opponent while preventing it from reaching its own side.

UNIT-IV

Variational Autoencoders (VAEs): Introduction to autoencoders , Variational inference and latent variable models ,Encoder and decoder architectures in VAEs , Training VAEs: maximizing evidence lower bound (ELBO) , Applications of VAEs in image generation, anomaly detection, and data imputation.

Practice:

1. Use deep reinforcement learning to train an agent to play various Atari games, such as Breakout, Space Invaders, or PacMan.
2. Experiment with different strategies for solving the multiarmed bandit problem, where an agent must decide which arm of a slot machine to pull to maximize cumulative reward.

UNIT – V

Advanced Topics in Reinforcement Learning and Generative AI: Model based RL and world models, multiagent reinforcement learning ,Transfer learning and meta learning in RL, Adversarial attacks and defences in generative models, Ethical considerations and societal impact of RL and generative AI, LangChain Framework

Practice:

1. Train an agent to play Flappy Bird, a sidescrolling game where the agent controls the flight of a bird through a series of pipes.
2. Develop an agent capable of playing TicTacToe optimally against a human opponent or another agent.

Additional Practice:

1. Train a robotic arm to perform various tasks, such as reaching a target location or

- manipulating objects, using reinforcement learning.
2. Develop a trading agent that learns to make profitable trades in financial markets by analysing historical data and adapting its trading strategy over time

Capstone Project

Smart Traffic Signal Control Using Q-Learning

Objective: Optimize traffic signal timings at intersections to reduce vehicle waiting time using RL.

Domain: Smart Cities / Traffic Management

Techniques & Concepts Used:

- *Q-Learning (Tabular RL)
- *State and Reward Modeling
- *Traffic Simulation Environment (e.g., SUMO or custom gridworld)
- *Visualization of Learning Process
- *Performance Metrics: Average wait time, throughput

Text Books:

- 1 Reinforcement Learning An Introduction, R. S. Sutton and A. G. Bart., MIT Press, 2018, ISBN: 978-0262039246

Reference Books:

- 1 Algorithms for Reinforcement Learning, Szepesvári, Csaba, United States: Morgan & Claypool, ISBN: 978-1627050278
- 2 Markov Decision Processes, Discrete Stochastic Dynamic Programming, Puterman, Martin L., Germany, Wiley, ISBN: 978-0471727200
- 3 Artificial Intelligence, Dr. R. Sravanth Kumar, Dr. A. Mallikarjuna Reddy, Dr T. Prabhakara Rao and Dr. Gayatri Parasa Published By Pandit Publications- 2024, ISBN: 978-93-93769-74-9

Web Links:

- 1 https://onlinecourses.nptel.ac.in/noc20_cs74/preview
- 2 <https://www.coursera.org/learn/fundamentalsofreinforcementlearning>

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

Course Code: 2501AI04	L	T	P	C
	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 and work with big Data
- CO5:** Make use of Spark tool to work with 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, Dirk de Roos, 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>