

MOHAN BABU UNIVERSITY

Sree Sainath Nagar, Tirupati – 517 102



SCHOOL OF COMPUTING

Master of Computer Applications (M.C.A)

CURRICULUM AND SYLLABUS *(From 2025-26 Admitted Students)*

FULLY FLEXIBLE CHOICE BASED CREDIT SYSTEM (FFCBCS)

Vision

To be a globally respected institution with an innovative and entrepreneurial culture that offers transformative education to advance sustainability and societal good.

Mission

- ❖ Develop industry-focused professionals with a global perspective.
- ❖ Offer academic programs that provide transformative learning experience founded on the spirit of curiosity, innovation, and integrity.
- ❖ Create confluence of research, innovation, and ideation to bring about sustainable and socially relevant enterprises.
- ❖ Uphold high standards of professional ethics leading to harmonious relationship with environment and society.

SCHOOL OF COMPUTING

Vision

To lead the advancement of computer science research and education that has real-world impact and to push the frontiers of innovation in the field.

Mission

- ❖ Instil within our students fundamental computing knowledge, a broad set of skills, and an inquisitive attitude to create innovative solutions to serve industry and community.
- ❖ Provide an experience par excellence with our state-of-the-art research, innovation, and incubation ecosystem to realise our learners' fullest potential.
- ❖ Impart continued education and research support to working professionals in the computing domain to enhance their expertise in the cutting-edge technologies.
- ❖ Inculcate among the computing engineers of tomorrow with a spirit to solve societal challenges.

DEPARTMENT OF COMPUTER APPLICATIONS

Vision

To become a center of excellence in the field of computer science and applications.

Mission

- Imparting knowledge and skills through contemporary curriculum to the diverse group of students.
- Creating a talent pool of faculty in diverse domains of computer applications through continuous training.
- Domain and transferable skill development for the holistic personality of students to inculcate values and ethics for effective professional practice and as an entrepreneur.

MASTER OF COMPUTER APPLICATIONS (MCA)

PROGRAM EDUCATIONAL OBJECTIVES

After a few years of completion of the Program, the graduates of MCA will be able to:

- PEO1.** Pursue higher education in the core and allied areas of computer science by applying computing knowledge and domain-specific knowledge, demonstrating their innovative skills, and considering social and environmental concerns.
- PEO2.** Become professionals in industry and academia with the ability to investigate and solve complex computing problems using modern tools in evolving technologies in the core and allied areas of computer science.
- PEO3.** Become successful entrepreneurs to excel in diverse application skills in the core or allied area of computer science of societal importance.
- PEO4.** Exhibit professionalism, and uphold health, safety, legal, environmental, ethical, and cultural diversity issues for serving the society and communicating with local and national peers, bound within regulations and leading to lifelong learning.

PROGRAM OUTCOMES

On successful completion of the Program, the graduates of M.C.A Program will be able to:

- PO1 (Foundation Knowledge):** Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.
- PO2 (Problem Analysis):** Identify, review, formulate and analyze problems for primarily focusing on customer requirements using critical thinking frameworks.
- PO3 (Development of Solutions):** Design, develop and investigate problems with as an innovative approach for solutions incorporating ESG/SDG goals.
- PO4 (Modern Tool Usage):** Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.
- PO5 (Individual and Teamwork):** Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.
- PO6 (Project Management and Finance):** Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.
- PO7 (Ethics):** Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware
- PO8 (Life-long learning):** Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.



Centre for Distance and Online Education

M.C.A. – Master of Computer Applications

Academic Year 2025-26

Course Structure

I-Semester

Sl. No	Course Code	Course Title	Lect ure	Tuto rial	Pra ctic al	Project-based Learning	Cre dits
			L	T	P	S	
1.	CA25PG0101L	COMPUTATIONAL STATISTICS	3	-	-	-	3
2.	CA25PG0102L	COMPUTER ORGANIZATION AND ARCHITECTURE	3	-	-	-	3
3.	CA25PG0201L	PYTHON ESSENTIALS	3	-	2	-	4
4.	CA25PG0202L	DATABASE MANAGEMENT SYSTEMS	3	-	2	-	4
5.	CA25PG0103L	SOFTWARE ENGINEERING	3	-	-	-	3
6.	CA25PG0203L	CLOUD COMPUTING	3	-	2	-	4
7.	CA25PG0104L	ESSENTIALS OF CYBER SECURITY	2	-	-	-	2
TOTAL			20	-	6	-	23

**Centre for Distance and Online Education****M.C.A. – Master of Computer Applications****Academic Year 2025-26****Course Structure****II-Semester**

Sl. No	Course Code	Course Title	Lecture	Tutorial	Practical	Project-based Learning	Credits
			L	T	P	S	
1.	CA25PG0105L	MATHEMATICAL FOUNDATIONS FOR COMPUTER APPLICATIONS	3	-	-	-	3
2.	CA25PG0204L	DATA MINING	3	-	2	-	4
3.	CA25PG0205L	COMPUTER NETWORKS	3	-	2	-	4
4.	CA25PG0401L	OBJECT ORIENTED PROGRAMMING THROUGH JAVA	3	-	2	4	5
5.	CA25PG0106L	BUSINESS ANALYTICS	3	-	-	-	3
6.	CA25PG0107L	INTERNET OF THINGS	3	-	-	-	3
7.	CA25PG0108L	BLOCKCHAIN TECHNOLOGIES	3	-	-	-	3
TOTAL			21	-	6	4	25



MOHAN BABU UNIVERSITY

Sree Sainath Nagar, Tirupati 517 102

Centre for Distance and Online Education

M.C.A. – Master of Computer Applications

Academic Year 2025-26

Course Structure

III-Semester

Sl. No	Course Code	Course Title	Lecture	Tutorial	Practical	Project-based Learning	Credits
			L	T	P	S	
1.	CA25PG110L	INTERNSHIP	-	-	-	-	2
2.	CA25PG0109L	OPERATING SYSTEMS	3	-	-	-	3
3.	CA25PG0206L	DATA STRUCTURES AND ALGORITHMS WITH PYTHON	3	-	2	-	4
4.	CA25PG0207L	MACHINE LEARNING	3	-	2	-	4
5.	CA25PG0208L	NATURAL LANGUAGE PROCESSING	3	-	2	-	4
6.	CA25PG0209L	MERN STACK DEVELOPMENT	3	-	2	-	4
7.	CA25PG0210L	DEVOPS	3	-	2	-	4
8.	CA25PG0501L	GENERATIVE AI TOOLS AND PROMPT ENGINEERING	-	1	2	-	2
9.	CA25PG0110L	FULL STACK WEB DEVELOPMENT	3	-	-	-	3
10.	CA25PG0111L	ENTREPRENEURSHIP AND INNOVATION MANAGEMENT	3	-	-	-	3
TOTAL			24	1	12	-	33

Course Code	Course Title	L	T	P	S	C
CA25PG0101L	COMPUTATIONAL STATISTICS	3	-	-	-	3
Pre-Requisite	--					
Anti-Requisite	--					
Co-Requisite	--					

COURSE DESCRIPTION: This course covers measures of Central Tendency, measures of Dispersion and basics of 'R' language. Fitting the Probability distributions of Discrete Probability Distributions, Continuous Probability Distributions and implementing the probability distributions in R. Correlation and Regression Analysis is done and able to calculate Correlation and Regression Coefficients along with fitting lines of Regression in R tool. Testing of hypothesis can be done for Large sample Tests and Small sample tests.

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

- CO1.** Demonstrate the concepts of descriptive statistics, basics in R, Probability Distributions, Testing of Hypothesis, Correlation and Regression.
- CO2.** Apply testing of hypothesis on small and large sample testing using Z-test, t-test, F-test and chi-square test and to draw the valid inferences and solve the problems on probability distributions.
- CO3.** Use R programming tools to Simulate Binomial, Poisson and Normal distributions, and calculate Correlation and Regression Coefficients, Fitting lines of Regression.
- CO4.** Analyze the relationship between variables using Correlation and Regression.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	3	-	-	-	-
CO2	3	3	2	3	3	-	-	-
CO3	3	3	3	3	3	-	-	-
CO4	3	3	3	3	-	-	-	-
Course Correlation Mapping	3	3	3	3	2	-	-	-

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

MODULE 1: INTRODUCTION TO DESCRIPTIVE STATISTICS (10 Periods)

Measures of Central Tendency: The Arithmetic Mean, The Arithmetic Mean Computed from Grouped data, Median, Mode, Empirical Relation Between the Mean, Median, and Mode, Geometric Mean, Harmonic Mean.

Measures of Dispersion: The Range, The Mean Deviation, The Semi- Interquartile Range, The Standard Deviation, The Variance, coefficient of variation and Moments, measures of Skewness and Kurtosis.

MODULE 2: BASICS IN R (09 Periods)

Basic concepts of R, R Studio, run R, Variables, Data Types, Vectors, Data Frames, Lists, Matrices, Arrays, Classes, Functions, and diagrammatical presentations in R.

MODULE 3: PROBABILITY DISTRIBUTIONS (09 Periods)

Discrete Probability Distributions: Binomial Distribution- Mean and variance and fitting of Binomial distribution; Poisson distribution -Mean and variance and fitting of Poisson distribution.

Continuous Probability Distributions: Normal Distribution- Mean, variance and area properties of Normal distribution.

Implementing in R: Simulate Binomial, Poisson and Normal distributions

MODULE 4: CORRELATION AND REGRESSION ANALYSIS (08 Periods)

Correlation Analysis: Linear Correlation, scatter diagram, Karl Pearson's coefficient of Correlation and Spearman's Rank correlation coefficient (with and without tied ranks).

Regression Analysis: Regression Lines, Fitting of two lines of Regression, Regression coefficients and multiple regression.

Implementing in R: Calculate Correlation and Regression Coefficients and Fitting lines of Regression.

MODULE 5: TESTING OF HYPOTHESIS (09 Periods)

Large sample Tests: Null hypothesis and Alternative hypothesis, Type-I and Type-II errors, Level of significance, Critical Region, one tailed and two tailed tests; Test of Significance of single proportion, Difference of two Proportions, Single mean, Difference of two Means.

Total Periods: 45

EXPERIENTIAL LEARNING

1. For Covid 19 scenario, do the following using R Programming tool:

Globally, the number of new weekly cases has continued to decline since the peak in January 2022. During the week of 30 May to 5 June 2022, over three million cases were reported, a 12% decrease as compared to the previous week. The number of new weekly deaths also continues to decline, with over 7,600 fatalities reported, representing a 22% decrease as compared to the previous week. As of 5 June 2022, over 529 million confirmed cases and over six million deaths have been reported globally. Who do you think were the cybercriminals behind it, and what might have been their motive?

- a. Requirement gathering

- o Identify the variables and methods
- o Draw Histogram for statistical analysis
- o Write use cases for Patient information & Recovery percentage

- b. Analysis

- Fit the Linear Regression model for the system
 - Draw Mathematical relationship to find the Patient information
- Identify the Patient current condition and to provide right Medication

c. Design

- Design Data Visualization for "patient entry "and conditions
- Draw R-Chart based on the Statistical analysis.

d. Evaluate quality of design

- Given R-Chart, diagrammatic Representation of Histogram, evaluation of mathematical relationships will provide the patient recovery percentage in Visualization.

RESOURCES

TEXT BOOKS:

1. T. K. V. Iyengar, B. Krishna Gandhi et al., "*Probability and Statistics*," S. Chand and Company Ltd: New Delhi, 3rd Edition, 2011.
2. Allerhand M. "*Tiny Handbook of R*," Springer Briefs in Statistics, 2011.

REFERENCE BOOKS:

1. Shanaz Bhatul, "*Probability and Statistics*", RIDGE Publications, 2nd Edition, 2006.
2. S.C. Gupta and V.K. Kapoor, "*Fundamentals of Applied Statistics*," S. Chand and Sons, New Delhi, 2010.
3. Baayen R. "*Analyzing Linguistic Data - A Practical Introduction to Statistics using R*," 2008.
4. Alain F. Zuur, Elena N. Ieno, Erik H.W.G. Meesters, "*Beginner's Guide to R*," Springer, 2009.

SOFTWARE/TOOLS:

1. R Programming
2. R studio

VIDEO LECTURES:

1. <https://www.youtube.com/watch?v=GuTd8Yb2jUk>
2. <https://www.youtube.com/watch?v=V8eKsto3Ug>

WEB RESOURCES:

1. <http://www.nptelvideos.in/2012/11/probability-and-statistics.html>
2. <https://www.classcentral.com/course/swayam-probability-and-statistics-5228>.

Course Code	Course Title	L	T	P	S	C
CA25PG0102L	COMPUTER ORGANIZATION AND ARCHITECTURE	3	-	-	-	3
Pre-Requisite	--					
Anti-Requisite	--					
Co-Requisite	--					

COURSE DESCRIPTION: This course deals with basic function, structure and components of computer system, Central Processing Unit, Control Unit and I/O Unit. This course also imparts knowledge on different types of Memory Systems, Parallel Processing techniques and Multicore architectures with its issues.

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

- CO1.** Demonstrate knowledge on computer organization, architecture and interconnection structures of a digital computer.
- CO2.** Analyze Arithmetic Operations, Addressing Modes, Instruction Formats, Processor and Register organization.
- CO3.** Design digital circuits for the given functional description of microoperations and memory elements.
- CO4.** Investigate the performance of Input/Output Systems, Memory systems, parallel processors, Multicore architectures to evaluate the cost-performance trade-offs.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	-	-	-	-	-	-
CO2	3	2		-	-	-	-	-
CO3	3	2	1	-	-	-	-	-
CO4	3	2	-	-	-	-	-	-
Course Correlation Mapping	3	2	1	-	-	-	-	-

Correlation Levels: 1: Low 3: High; 2: Medium

COURSE CONTENT

MODULE 1: INTRODUCTION TO COMPUTER SYSTEM

(08 Periods)

Basic Functional units of Computers: Functional units, basic Operational concepts, Bus structures. Software, Performance, Multiprocessors, Multicomputer. Data Representation: Signed number representation, fixed and floating point Representations.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms. Error detection and correction codes.

MODULE 2: CENTRAL PROCESSING UNIT

(09 Periods)

Arithmetic logic unit, Integer multiplication- Booth's algorithm; Floating point representation principles, Machine instruction characteristics, Addressing modes, Instruction formats – Instruction length, Allocation of bits; Processor organization, Register organization – User visible registers, Control and status registers; Instruction cycle.

MODULE 3: CONTROL UNIT AND INPUT/OUTPUT

(10 Periods)

Control Unit: Micro operations, Control of the processor – Functional requirements, Control signals, Internal processor organization; Hardwired implementation, Microinstructions, Microprogrammed control unit, Micro instruction sequencing – Design considerations, sequencing techniques, Address generation; Micro instruction execution- A taxonomy of microinstructions, Microinstruction encoding.

Input/Output: External devices, I/O modules, Direct memory access function, I/O Channels and Processors, PCI Physical and Logical Architecture.

MODULE 4: MEMORY SYSTEMS

(09 Periods)

Semiconductor Memories: Computer Memory System overview, Semiconductor Main Memory- Organization, DRAM and SRAM, Types of ROM, Chip logic, Chip packaging, Module organization, Interleaved memory; DDR DRAM, Flash Memory, Newer Nonvolatile Solid-State Memory Technologies.

Cache Memory: Cache Memory Principles, Elements of Cache Design-Cache Addresses, Mapping Functions.

External Memory: Solid State Drives.

MODULE 5: PARALLEL ORGANIZATION

(09 Periods)

Parallel Processing: Multiple Processor Organizations, Instruction Pipelining, Symmetric Multiprocessors, Nonuniform Memory Access, Multicore Organization.

Multicore Computers: Hardware performance issues, Software performance issues, Multicore organization, Intel Core i7-990X.

Total Periods: 45

1. Write a report on Product Specifications such as CPU, Memory, Processor Graphics and advanced technologies of Intel Core i9-12900K -New performance hybrid architecture. The World's Best Gaming Processor: Available with up to 16 cores and 24 threads, the new 12th Gen Intel Core processor family includes the world's best gaming processor, the Core i9-12900K, unleashing gaming experiences across top titles. (<https://ark.intel.com/content/www/us/en/ark/products/134599/intel-core-i912900k-processor-30m-cache-up-to-5-20-ghz.html>)
2. Write a report on OpenCL(Open Computing Language) framework used for writing programs that execute across heterogeneous platforms consisting of central processing units, graphics processing units, digital signal processors, field-programmable gate arrays and other processors or hardware accelerators. (<https://www.khronos.org/api/opencl>)

EXPERIENTIAL LEARNING

RESOURCES

TEXT BOOKS:

- 1 William Stallings, Computer Organization and Architecture: Design for performance, Pearson, 11th Edition, 2020.

REFERENCE BOOKS:

1. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, Computer Organization, 5th Edition, McGraw, Hill Education, 2013.
2. David A. Patterson and John L. Hennessy, Computer Organization and Design - A Hardware software interface, 5th Edition, Morgan Kaufmann, 2014.

VIDEO LECTURES:

1. https://onlinecourses.nptel.ac.in/noc21_cs37/preview
2. https://onlinecourses.nptel.ac.in/noc20_cs64/preview
3. https://onlinecourses.nptel.ac.in/noc21_cs47/preview
4. <https://freevideolectures.com/course/2277/computer-organization>
5. <https://www.youtube.com/watch?v=4TzMyXmzL8M>

WEB RESOURCES:

1. <https://www.javatpoint.com/computer-organization-and-architecture-tutorial>
2. <https://www.geeksforgeeks.org/last-minute-notes-computer-organization/>
3. <https://gateoverflow.in/blog/9728/some-good-resources-for-computer-organisation-architecture>
4. <https://tutorialspoint.dev/computer-science/computer-organization-and-architecture/>

Course Code	Course Title	L	T	P	S	C
CA25PG0201L	PYTHON ESSENTIALS	3	-	2	-	4

Pre-Requisite -

Anti-Requisite -

Co-Requisite -

COURSE DESCRIPTION: This course introduces Python programming with fundamentals of input/output, data types, operators, control structures, and exception handling. It covers strings, data structures, functions, recursion, and object-oriented programming concepts. Learners explore data analysis and visualization using NumPy, Pandas, and Matplotlib. The course also provides basics of web development with Flask, leading to a mini project.

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

- CO1.** Demonstrate understanding of Python basics including input/output, data types, operators, and expressions.
- CO2.** Apply decision-making constructs, looping techniques, and comprehensions to solve computational problems.
- CO3.** Develop Python applications using strings, data structures, functions, recursion, and functional programming tools to solve problems.
- CO4.** Design applications using object-oriented programming features – encapsulation, inheritance, polymorphism, and exception handling.
- CO5.** Utilize Python libraries such as NumPy, Pandas, Matplotlib for data analysis and visualization, and develop simple web applications using Flask.
- CO6.** Work independently to solve problems with effective communication.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	-	-	3	-	-	-
CO2	3	2	-	-	3	-	-	-
CO3	3	3	3	3	3	-	-	-
CO4	3	2	2	2	3	-	-	-
CO5	3	3	3	3	3	-	-	-
CO6	-	-	-	-	-	2	2	2
Course Correlation Mapping	3	3	2	2	2	1	1	1

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

MODULE 1: INTRODUCTION TO PYTHON, INPUT/OUTPUT AND OPERATORS

(07 Periods)

Features of Python, history, installation, interpreter vs compiler, Python environments: Jupyter, VS Code; Input using input(), output using print(), formatted strings using f-strings and format(), Data types: int, float, str, bool; dynamic typing, type conversions: explicit, implicit; arithmetic, relational, logical, bitwise, assignment, membership, identity operators, expression evaluation, operator precedence and associativity. Single-line and multi-line comments.

MODULE 2: DECISION MAKING AND LOOPING

(08 Periods)

Conditional statements using if, if-else, elif, nested conditions, Loops: for loop, while loop, break, continue, pass; enumerate() and zip() function, else with loops, List and dictionary comprehensions, combining loops with conditionals for iterative logic, basic exception handling using try-except.

MODULE 3: DATA STRUCTURES, STRINGS, FUNCTIONS & RECURSION

(10 Periods)

Strings: creation, indexing, slicing, immutability; common methods: upper, lower, find, replace, split, join; Data structures: lists, tuples, dictionaries, sets, operations and use cases; Functions: definition, parameters, return values, default/keyword arguments; recursion concepts and examples (factorial, Fibonacci), function integration with strings and collections. Lambda functions, map() and filter()

MODULE 4: OBJECT-ORIENTED PROGRAMMING IN PYTHON

(08 Periods)

Classes and objects, defining classes, instance variables and methods, Constructors (__init__), destructors, encapsulation with access conventions: public, protected, private; Inheritance and its types, method overriding, polymorphism, abstraction using abstract classes (abc module).

MODULE 5: DATA VISUALIZATION & WEB DEVELOPMENT

(12 Periods)

Importance and purpose of data visualization, Numpy, Pandas, Matplotlib: line graphs, bar charts, histograms, scatter plots; Flask overview and environment setup, App structure and routing, Template rendering with Jinja2, Form handling: GET/POST methods, Basic CRUD operations, HTML/CSS integration, Mini project implementation

Total Periods: 45

EXPERIENTIAL LEARNING

1) INTRODUCTION TO PYTHON, INPUT/OUTPUT AND OPERATORS **1. FORMATTED INPUT/OUTPUT PROGRAM**

Exercise: Create a Receipt Formatter

Write a Python program that takes user input for item names, quantities, and prices, then prints a formatted receipt using string formatting.

Example Input:

Enter item: Apple

Quantity: 3

Price per item: 25

Enter item: Bread

Quantity: 2

Price per item: 40

Example Output:

----- RECEIPT -----

Apple x3 ₹75

Bread x2 ₹80

Total: ₹155

Concepts Reinforced:

- f-strings and .format() for clean alignment
- input() function and basic control flow
- Improves fluency in creating readable CLI outputs
- Introduces beginners to formatted design and user interaction

2. TYPE CONVERSION & ARITHMETIC EXPRESSION EXERCISES

Exercise: Calculate Average Marks

Prompt the user to enter marks of five subjects. Convert the input to integers or floats, calculate the average, and display the result.

Example Input:

Enter marks (comma-separated): 87,92,78,90,85

Example Output:

Average Marks: 86.4

Grade: A

Concepts Reinforced:

- split() and map() for type conversion
- Arithmetic operations: sum and division
- Decision-making for grading logic (optional extension)
- Strengthens understanding of parsing and converting data types

3. OPERATOR-BASED MINI CALCULATOR

Exercise: Interactive Calculator

Build a simple calculator that allows users to perform basic operations (+, -, *, /, %, //, **).

Example Input:

Enter first number: 12

Enter operator: *

Enter second number: 4

Example Output:

Result: 48

Concepts Reinforced:

- Operators and precedence
- Control flow with if-elif statements
- User-friendly interaction via input()
- Solidifies arithmetic logic implementation

2) DECISION MAKING AND LOOPING

Exercise: Grade Evaluator

Create a program that takes exam marks as input and outputs a grade based on predefined criteria using conditional statements.

Example Input:

Enter your marks: 76

Example Output:

You scored 76 marks.

Grade: B

Grading Rules (Example):

- A: 90–100

- B: 75–89
- C: 60–74
- D: 40–59
- F: Below 40

Concepts Reinforced:

- if-elif-else statements
- Comparison operators
- Control flow logic and decision-making
- Real-world scenario application

Exercise: Guess the Number

Write a game where the user must guess a randomly generated number between 1 and 100. The program should give hints and allow multiple attempts.

Example Output:

Guess a number between 1 and 100: 50

Too low!

Guess again: 75

Too high!

Guess again: 64

Correct! You guessed it in 3 tries.

Concepts Reinforced:

- while loops
- Random number generation (random.randint)
- Loop control with break conditions
- Interactive CLI experience and feedback

Exercise: Print a Pyramid Pattern

Create a program that prints a pyramid of stars based on user-defined height using nested loops.

Example Input:

Enter the number of rows: 4

Example Output:

```
*
***
*****
*****
```

Concepts Reinforced:

- Nested for loops
- String multiplication and spacing
- Visualization of loop nesting through output
- Creativity with logic-based formatting

Exercise: Display Index and Value

Use enumerate() to print items of a list with their index.

```
colors = ["red", "green", "blue"]
```

```
for index, color in enumerate(colors):
```

```
    print(f'{index}: {color}')
```

Exercise: Match Items from Two Lists

Use zip() to pair elements from two lists and print them.

```
names = ["Alice", "Bob", "Charlie"]
```

```
scores = [95, 80, 88]
```

```
for name, score in zip(names, scores):
```

```
    print(f'{name} scored {score}')
```

Concepts Reinforced:

- Advanced iteration techniques
- Pairing data logically

- Cleaner, more Pythonic loops
- Enhanced readability and flexibility

3) **DATA STRUCTURES, STRINGS, FUNCTIONS & RECURSION**

Exercise: Text Analyzer Tool

Write a Python script that accepts a sentence from the user and performs the following operations:

- Counts total words
- Converts all letters to uppercase and lowercase
- Removes punctuation
- Reverses the sentence
- Extracts unique words

Example Input:

Enter a sentence: Hello, world! Welcome to Python.

Example Output:

Original: Hello, world! Welcome to Python.

Uppercase: HELLO, WORLD! WELCOME TO PYTHON.

Lowercase: hello, world! welcome to python.

Word count: 5

Unique words: {'Hello', 'world', 'Welcome', 'to', 'Python'}

Reversed: .nohtyP ot emocleW !dlrow ,olleH

Concepts Reinforced:

- String methods: .lower(), .upper(), .split(), .replace(), slicing
- Sets and basic data structures
- Input handling and output formatting
- Sharpening problem-solving through text-based logic

Exercise: Phonebook Manager

Create a simple contact book that allows users to add, search, update, and delete contact information using a dictionary.

Example Interaction:

1. Add Contact
2. Search Contact
3. Update Contact
4. Delete Contact
5. Exit

Choose option: 1

Enter name: Priya

Enter number: 9876543210

Contact added.

Choose option: 2

Enter name to search: Priya

Number: 9876543210

Concepts Reinforced:

- Dictionaries for key-value storage
- Control flow with menus and choices
- Exception handling and validation
- Real-world simulation of data management systems

4) **OBJECT ORIENTED PROGRAMMING IN PYTHON**

Exercise: Create a Student Class

Write a class Student with a constructor (__init__) that initializes attributes like name, age, and grade. Include a method to display student details.

Example Output:

Name: Arjun

Age: 17

Grade: A

Concepts Reinforced:

- Class creation and instance variables
- Use of `__init__` constructor
- Object instantiation and method definition
- Encourages understanding of encapsulation and code reuse

Exercise: Company Employee Structure

Design a base class `Employee` with attributes such as name and salary. Derive two subclasses `Manager` and `Developer` that add specific attributes like department and programming language.

Example Output:

Manager: Ravi, Salary: ₹80,000, Department: Sales

Developer: Maya, Salary: ₹70,000, Language: Python

Concepts Reinforced:

- Single and multi-level inheritance
- Use of `super()` to access parent class methods
- Demonstrates code reuse and organization
- Encourages a real-world perspective on class relationships

Exercise: Unified Interface for Animals

Create a base class `Animal` with a method `make_sound()`. Define subclasses `Dog`, `Cat`, and `Cow`, each overriding `make_sound()` with their unique behavior.

Example Output:

Dog says: Woof!

Cat says: Meow!

Cow says: Moo!

Concepts Reinforced:

- Method overriding
- Polymorphism through dynamic method calls
- Promotes writing flexible and extendable code
- Highlights the importance of interfaces and consistency

5) DATA VISUALIZATION**Example 1: Student Performance Analysis**

Objective: Analyze and visualize students' exam scores across subjects.

Sample Dataset:

```
data = {  
    'Name': ['Asha', 'Kiran', 'Neha', 'Ravi'],  
    'Math': [88, 76, 90, 65],  
    'Science': [92, 70, 85, 60],  
    'English': [85, 80, 78, 70]  
}  
} Programming For Problem
```

Concepts Reinforced:

- Use of pandas DataFrame for organizing tabular data
- Bar charts for comparative analysis
- Legends, axis labels, and titles for clarity
- Understanding performance trends across categories

Example 2: Population Growth Trends

Objective: Plot population growth of countries over years to identify patterns.

Sample Dataset:

```
data = {  
    'Year': [2000, 2010, 2020],  
    'India': [1.05, 1.21, 1.38],  
    'China': [1.26, 1.34, 1.41],  
    'USA': [0.28, 0.31, 0.33]  
}  
}
```

Concepts Reinforced:

- Transposing data for time series plotting
- Line plots to show progression over time
- Grid lines for readability
- Data-driven storytelling using real-world metrics

6) WEB DEVELOPMENT IN PYTHON

Exercise: Create a user registration page that includes fields like email, password, and password confirmation. Ensure all fields are properly validated:

- Email must be in correct format
- Password must meet length or strength requirements
- Confirm password must match the original password

Skills Developed:

- Route handling in Flask
- Form creation using Flask-WTF and WTForms
- Server-side validation logic
- Delivering real-time feedback to users

Exercise: Build a web-based student record manager where users can perform CRUD (Create, Read, Update, Delete) operations:

- Add student details like name, roll number, and marks
- View a list of students
- Edit or delete entries when needed

Skills Developed:

- Integrating Flask with a database (like SQLite) using SQLAlchemy
- RESTful routing and templates with Jinja2
- Managing data dynamically through web pages
- Creating responsive user interfaces

Exercise: Develop a small personal portfolio showcasing projects, hobbies, or profile info. Include a feedback form where visitors can submit comments or messages.

Skills Developed:

- Designing static and dynamic web pages
- Using Flask blueprints for clean structure
- Building feedback forms and storing responses
- Enhancing layout with CSS and front-end tools

RESOURCES

TEXT BOOKS:

1. Ultimate Python Programming – Deepali Srivastava, BPB publishers, 2024
2. Intro to Python for Computer Science and Data Science - Paul J. Deitel, Pearson, 2022

REFERENCE BOOKS:

1. Learn Python the Hard Way - ZED Shaw's Hard Way Series, 5th Edition Pearson, 2024
2. Hands-on Data Analysis & Visualization with Pandas, BPB publishers, 2020
3. Flask Framework Cookbook - Third Edition, Packt Publishing, 2020

SOFTWARE/TOOLS:

1. Python 3.10
2. Jupyter Notebook/Pycharm/Vscode
3. Any browser

VIDEO LECTURES:

1. https://onlinecourses.nptel.ac.in/noc19_cs41/preview
2. <https://www.coursera.org/specializations/python>
3. <https://www.coursera.org/learn/python-for-applied-data-science-ai>
4. <https://www.youtube.com/watch?v=ix9cRaBkVe0&pp=ygUPcHI0aG9uIHR1dG9yaWFs>
5. <https://www.youtube.com/playlist?list=PL-osiE80TeTs4UjLw5MM6OjgkjFeUxCYH>
6. <https://www.youtube.com/watch?v=GPVsHOIRBBI&pp=ygUbZGF0YSB2aXN1YWxpemF0aW9uIHBsYXIsaXN0>

WEB RESOURCES:

1. <https://www.programiz.com/python-programming>
2. <https://www.geeksforgeeks.org/python-programming-language/>
3. <https://electronstudio.github.io/pygame-zero-book/>
4. <https://www.learnpython.org/>
5. <https://riptutorial.com/ebook/flask>

Course Code	Course Title	L	T	P	S	C
CA25PG0202L	DATABASE MANAGEMENT SYSTEMS	3	-	2	-	4
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This Course provides theoretical concepts and hands-on experience on Database systems, Database design, Relational model, Relational algebra, SQL queries, Constraints and triggers, PL/SQL, Schema refinement and normal forms, Transaction management, Concurrency control, Overview of storage and indexing.

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

- CO1.** Analyze and apply the concepts of ER-modeling and normalization to design viable data models for a given problem.
- CO2.** Formulate relational database schemas, apply suitable integrity constraints, for querying databases.
- CO3.** Use SQL to store, query, and manipulate data in relational databases.
- CO4.** Develop PL/SQL blocks to centralize database applications for maintainability
- CO5.** Analyze transaction processing, concurrency control and storage methods for databased management.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	2	3	-	-	-	-	-
CO2	1	3	2	2	3	-	-	-
CO3	1	2	3	2	3	-	-	-
CO4	2	3	3	3	3	2	-	-
CO5	-	2	-	2	1	-	-	-
Course Correlation Mapping	2	3	2	2	2.5	1	-	-

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

MODULE 1: INTRODUCTION TO DATABASE SYSTEMS AND DATABASE DESIGN (08 Periods)

Introduction to Database Systems: Database system applications, Purpose of database systems, View of data - Data abstraction, Instances and schemas, Data models; Database languages - Data Definition Language, Data Manipulation Language; Database architecture, Database users and administrators.

Introduction to Database Design: Database design and ER diagrams, Entities, attributes and entity sets, Relationships and relationship sets, Additional features of ER model.

MODULE 2: RELATIONAL MODEL AND RELATIONAL ALGEBRA (08 Periods)

Relational Model: Creating and modifying relations, Integrity constraints over relations, Enforcing integrity constraints, Querying relational data, Logical database design, Introduction to views, Destroying/altering tables and views.

Relational Algebra: Preliminaries, Relational Algebra operators.

MODULE 3: SQL AND PL/SQL (10 Periods)

SQL: Form of basic SQL query, Nested queries, Aggregate operators, Null values, Complex integrity constraints in SQL, Triggers.

PL/SQL: Generic PL/SQL block, PL/SQL data types, Control structure, Procedures and functions, Cursors, Database triggers.

MODULE 4: SCHEMA REFINEMENT AND TRANSACTIONS (10 Periods)

Schema Refinement: Problems caused by redundancy, Decompositions, Problems related to decomposition, Functional dependencies, Reasoning about FDs, First normal form, Second normal form, Third normal form, Boyce-Codd normal form, Multivalued dependencies, Fourth normal form, Join dependencies, Fifth normal form.

Transactions: Transaction concept, Transaction atomicity and durability, Concurrent Executions – Serializability, Recoverability, Implementation of isolation, Testing for serializability.

MODULE 5: CONCURRENCY CONTROL, STORAGE AND INDEXING (09 Periods)

Concurrency Control: Lock Based Protocols, Timestamp Based Protocols, Validation Based Protocols, Multiple Granularity, Deadlock Handling.

Storage and Indexing: Data on external storage, File organizations and indexing – Clustered indexes, Primary and secondary indexes; Index data structures – Hash based indexing, Tree based indexing; Comparison of file organizations.

Total Periods: 45

EXPERIENTIAL LEARNING

- 1) Design and analyze an ER-Model for the following use case.
Road way Travels is in business since 1977 with several buses connecting different places in India. Its main office is in Hyderabad. The company wants to computerize its operations in the following areas:
 - a) Reservations
 - b) Ticketing
 - c) Cancellations

Reservations:

Reservations are directly handled by booking office. Reservations can be made 60 days in advance in either cash or credit. In case the ticket is not available, await listed ticket is issued to the customer. This ticket is confirmed against the cancellation.

Cancellation and Modification:

Cancellations are also directly handed at the booking office. Cancellation charges will be charged. Wait listed tickets that do not get confirmed are fully refunded.
- 2)
 - a) Implement Single Row functions-Character, Numeric and Date functions.
 - b) Implement Data Definition Language commands>Create, Alter, Drop, Truncate, and Rename.
 - c) Implement Data Manipulation Language commands-Insert, Select, Update, and Delete.
- 3) Implement various types of integrity constraints-NOT NULL constraint, DEFAULT constraint, UNIQUE constraint, PRIMARY key, FOREIGN key, CHECK constraint.
- 4)
 - a) Implement group functions with different operators such as aggregate operators, group by, having and order by.
 - b) Implement nested and correlated nested queries using set operators and set comparison operators.
- 5)
 - a) Creation of views, synonyms, sequence, indexes and save point
 - b) Implement various types of joins-outer join and inner join.

Basic PL/SQL:

- 6) Construct PL/SQL block for the following:
 - a) To determine whether a number is palindrome
 - b) To determine whether a number is an Armstrong number
 - c) To find greatest of three numbers
 - d) To display Fibonacci series

Control Structures:

- 7)
 - a) Write a programming PL/SQL to update the salary of a specific employee by 8% if the salary exceeds the mid-range of the salary against this job and update up to mid-range if the salary is less than the mid-range of the salary, and display a suitable message.
 - b) Write a PL/SQL program to display the description against a student's grade using CASE statement.

Exception Handling:

- 8) a) Develop a PL/SQL program that displays the name and address of a student whose ID is given. If there is no student with the given student ID in the database, the program should raise a run-time exception NO_DATA_FOUND, which should be captured in the EXCEPTION block.
- b) Construct the user-defined exceptions to get the salary of an employee and check it with the job's salary range. If the salary is below the range, raise an exception BELOW_SALARY_RANGE. If his salary is above the range, raise the exception ABOVE_SALARY_RANGE.

Functions

- 9) a) Write a function that accepts two numbers A and B and performs the following operations.
- o Addition
 - o Subtraction
 - o Multiplication
 - o Division
- b) Write a PL/SQL block that updates salary of an employee in Employee table by using incr function which takes employee number as argument and calculates increment and returns increment based on the following criteria.
- o If salary ≤ 3000, increment = 30% of salary
 - o If salary > 3000 and ≤ 6000, increment = 20% of salary
 - o Else increment = 10% of salary

Procedures:

- 10) a) Write a procedure that accepts two numbers and displays their sum.
- b) Write procedures to demonstrate IN, INOUT and OUT parameters.

Cursors:

- 11) a) Write a block in PL/SQL to create a Cursor that displays the employee name and number of jobs he/she has done in the past.
- b) Write a program in PL/SQL to create a cursor to display the name and salary of each employee in the EMPLOYEES table whose salary is less than that specified by a passed-in parameter value.

Triggers:

- 12) Develop a suitable student database application by considering appropriate attributes. Couple of attributes to be maintained is the attendance of a student in each subject for which he/she has enrolled and internal assessment Using TRIGGERS for the following:

RESOURCES

TEXT BOOKS:

1. Raghu Ramakrishnan, Johannes Gehrke, Database Management Systems, McGraw Hill, 3rd Edition, 2014.
2. Abraham Silberschatz, Henry. F. Korth, S. Sudarshan, Database System Concepts, McGraw Hill, 7th edition, 2019.

REFERENCE BOOKS:

1. Ivan Bayross, SQL, PL/SQL: The Programming Language of Oracle, BPB publications, 4th Edition, 2017.
2. Ramez Elmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson, 2015.
3. Thomas Connolly, Carolyn Begg, Database Systems, Pearson, 6th edition, 2019.
4. S.K.Singh, Database System Concepts, Design and Applications, 1st Edition, Pearson Education, 2006.
5. SatishAnsani, Oracle Database 11g: Hands-on SQL and PL/SQL, PHI, 2010.
6. Dr. Rajiv Chopra, Database Management Systems, Sultan Chand, 2016.
7. Pranab Kumar Das Gupta, P. Radha Krishna, Database Management System Oracle SQL and PL/SQL, PHI, 2nd Edition, 2009.

VIDEO LECTURES:

1. https://swayam.gov.in/nd1_noc19_cs46/preview
2. <https://nptel.ac.in/courses/106105175>
3. https://onlinecourses.nptel.ac.in/noc21_cs04/preview
4. <https://www.youtube.com/watch?v=MDQxqYVXiVU>
5. <https://www.youtube.com/watch?v=c5HAWKX-suM>

WEB RESOURCES:

1. <https://www.classcentral.com/course/swayam-introduction-to-database-systems-17660>
2. <https://www.scaler.com/topics/dbms/>
3. https://www.academia.edu/27988617/Database_Management_System_DBMS_Tutorial
4. <https://nptel.ac.in/courses/106104135>
5. <https://downloads.mysql.com/docs/mysql-tutorial-excerpt-5.7-en.pdf>

Course Code	Course Title	L	T	P	S	C
CA25PG0103L	SOFTWARE ENGINEERING	3	-	-	-	3
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This course provides a detailed discussion on concepts of Software Engineering, Software Process Models, Conventional and Agile Process Models, Software Requirements Engineering Process, System Analysis, Architectural Design, User Interface Design and Re-engineering, Software Testing, Risk and Quality Management.

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

- CO1.** Understand fundamental concepts of software engineering and analyze process models required to develop a software system.
- CO2.** Analyze software requirements and model requirements for the given scenario.
- CO3.** Apply design concepts and metrics for software development.
- CO4.** Apply testing strategies and techniques for quality software.
- CO5.** Analyze risks in software development life cycle and apply risk strategies to mitigate risks.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	-	-	-	-	-	-
CO2	3	2	3	3	-	-	-	-
CO3	3	2	2	3	3	-	-	-
CO4	3	2	2	3	-	-	-	-
CO5	2	-	-	-	-	2	-	-
Course Correlation Mapping	3	2	2	3	3	2	-	-

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

MODULE 1: SOFTWARE ENGINEERING AND SOFTWARE PROCESS (10 Periods)

A Generic view of process: The Nature of Software, Software Engineering- Software Engineering Layers; The Software Process, Software Engineering Practice, Software myths.

Process models: A Generic Process Model, Prescriptive Process Models-The Waterfall Model, Incremental Process Models, Specialized Process Models; The Unified Process, Agile Development-Agility, Agile Process, Extreme Programming (XP), Scrum, Dynamic System Development Method, Agile Modeling (AM), Agile Unified Process (AUP).

MODULE 2: REQUIREMENTS ENGINEERING AND MODELING (08 Periods)

Requirements Engineering: Functional and non-functional requirements, The software requirements document, Requirements specifications, Requirements engineering processes, Requirements elicitation and analysis, Requirements validation, Requirements management.

Requirements Modeling: Requirements Analysis, Data Modeling Concepts, Flow-Oriented Modeling, Scenario based Modeling, UML Models that supplement the Use Case.

MODULE 3: DESIGN ENGINEERING AND METRICS (09 Periods)

Design using UML: Class Diagram - Terms and concepts, Use case Diagram - Terms and concepts, Activity Diagrams - Terms and concepts, Interaction diagrams - Terms and concepts, State machine Diagram- Terms and concepts, Component Diagram- Terms and concepts, Deployment Diagram- Terms and concepts.

Process and Project Metrics: Metrics in the process and project domains, Software Measurement, Metrics for software quality.

MODULE 4: SOFTWARE TESTING STRATEGIES AND APPLICATIONS (08 Periods)

Testing strategies: A strategic approach to software testing, Strategic issues, Test strategies for conventional software, Test strategies for object oriented software, Validation testing, System testing.

Testing Conventional Applications: Software testing fundamentals, White box testing-Basis path testing, Control structure testing; Black box testing, Object oriented testing methods.

MODULE 5: RISK, QUALITY MANAGEMENT AND REENGINEERING (10 Periods)

Risk and Quality Management: Reactive and Proactive risk strategies, Software risks, Risk Mitigation Monitoring and Management (RMMM), RMMM plan, Formal Technical Reviews (FTR), Software Quality Assurance (SQA)-Tasks, Goals and Metrics; Software reliability.

Reengineering: Introduction, Business Process Reengineering (BPR), Software Reengineering, Restructuring, Reverse engineering, Forward engineering.

Total Periods: 45

EXPERIENTIAL LEARNING

1. Identify Functional and Non-Functional Requirements for:
 - a. Online Ticket Reservation for Railways
 - b. Recruitment Procedure for Software Industry

2. **Online Ticket Reservation for Railways**

Problem Statement: Computer plays an integral part of the day in today's life. It makes the entire job easier and faster, every job is computerized so as the ticket reservation we can book over the online ticket reservation system. During the booking of the ticket reservation passenger has to select origin, date of journey, destination, class of train etc. The reservation counter keeps track of passenger's information. Thus the system will have all the details about the trains and facilities provided by them. There are various trains with the different level of convenience for the passengers. The whole database will be maintained by database administrator. There are varieties of trains where the passengers can select the train according to the convenience for their destination journey. The journey could be within the state or across the India. Each train has the three types of classes i.e. Sleeper class, First class and the AC compartment. Design the application for the above problem description. Model using Visual modelling tools in different views i.e. component view, Deployment view.

3. **Recruitment Procedure for Software Industry**

Problem Statement: In the software industry the recruitment procedure is the basic thing that goes in the hand with the requirement as specified by the technical management team. HR first gives an advertisement in leading Newspapers, Journals, Weeklies and Websites. The job seekers can apply for it through by Post or by e-mail to the company. The technical skill and the experience of the candidates are reviewed and the short listed candidates are called for the interview. There may be different rounds for interview like the written test, technical interview, and HR interview. After the successful completion of all rounds of interview, the selected candidates' names are displayed. Meanwhile HR gives all the details about the salary, working hours, terms and conditions and the retirement benefit to the candidate. Model using Visual modelling tools in different views i.e. Use case view, logical view

RESOURCES

TEXT BOOKS:

1. Roger S. Pressman, Software Engineering - A Practitioner's Approach, McGraw-Hill International Edition, Eighth Edition, 2015.
2. Ian Sommerville, Software Engineering, Pearson Education, Ninth Edition, 2011.

REFERENCE BOOKS:

1. Grady Booch, James Rumbaugh and Ivar Jacobson, "The Unified Modeling Language User Guide," Second Edition, Pearson Education, 2009.
2. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers, Third Edition, 2007.
3. Shelly Cashman Rosenblatt, Systems Analysis and Design, Thomson Publications, Sixth Edition, 2006.

VIDEO LECTURES:

1. <https://nptel.ac.in/courses/106105087/>
2. <https://nptel.ac.in/courses/106105182/>

WEB RESOURCES:

1. AgileModeling:<https://www.techtarget.com/searchsoftwarequality/definition/agile-software-development>
2. Functional and Non Functional Requirements:
<https://www.linkedin.com/learning/software-design-developing-effective-requirements/functional-vs-non-functional>
3. Software Metrics and Measures:
https://www.youtube.com/watch?v=bnydxXPN_rI
4. Risk Management: <https://www.youtube.com/watch?v=8DstdqQBjps>

Course Code	Course Title	L	T	P	S	C
CA25PG0203L	CLOUD COMPUTING	3	-	2	-	4
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This Course provides a detailed discussion on Virtualization, Cloud Computing Fundamentals along with Deployment Models. Able to understand Cloud Computing Architecture to work with Cloud adopting Cloud Computing Mechanisms and implementing Cloud Security Mechanisms in Cloud Service Models.

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

- CO1.** Demonstrate the concepts of Cloud Computing, Cloud-Enabling Technology, Cloud Architectures.
- CO2.** Apply virtualization and develop virtual environments for the deployment of cloud applications.
- CO3.** Design applications using cloud service models salesforce.com (SaaS), Google App Engine (PaaS), Amazon (IaaS) and deploy in cloud.
- CO4.** Identify and analyze the Cloud-Enabling Technologies and architectures for developing the applications to solve e-commerce problems.
- CO5.** Adhere to ethics and adapt cloud security mechanisms and Cloud-Based Security Groups for providing security to societal applications.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	-	-	-	-	-
CO2	3	3	3	-	3	-	-	-
CO3	3	3	-	-	3	-	-	-
CO4	2	3	3	-	3	-	-	-
CO5	-	-	-	-	-	2	2	-
Course Correlation Mapping	3	3	3	-	3	1	1	-

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

MODULE 1: FUNDAMENTAL CLOUD COMPUTING (10 Periods)

Understanding Cloud Computing: Origins and Influences, Concepts and Terminology, Goals and Benefits, Risks and Challenges.

Fundamental Concepts and Models: Roles and Boundaries, Cloud Characteristics, Cloud Delivery Models, Cloud Deployment Models.

MODULE 2: CLOUD COMPUTING MECHANISMS AND ARCHITECTURE (10 Periods)

Cloud-Enabling Technology: Broadband Networks and Internet Architecture, Data Center Technology, Virtualization Technology, Web Technology, Multitenant Technology, Service Technology.

Fundamental Cloud Architectures: Workload Distribution, Resource Pooling, Dynamic Scalability, Elastic Resource Capacity, Service Load Balancing, Cloud Bursting, Elastic Disk Provisioning.

MODULE 3: CLOUD COMPUTING ADVANCED ARCHITECTURES (09 Periods)

Advanced Cloud Architectures: Hypervisor Clustering, Load Balanced Virtual Server Instances, Non-Disruptive Service Relocation, Zero Downtime, Cloud Balancing, Resource Reservation, Dynamic Failure Detection and Recovery, Bare-Metal Provisioning, Rapid Provisioning, Storage Workload Management.

Specialized Cloud Architectures: Direct I/O Access, Direct LUN Access, Dynamic Data Normalization, Elastic Network Capacity, Cross-Storage Device Vertical Tiering, Intra Storage Device Vertical Data Tiering, Load Balanced Virtual Switches, Multipath Resource Access, Persistent Virtual Network Configuration, Redundant Physical Connection for Virtual Servers.

MODULE 4: CLOUD SECURITY (08 Periods)

Fundamental Cloud Security: Threat Agents, Cloud Security Threats, Additional Considerations, Case Study Example.

Cloud Security Mechanisms: Encryption, Hashing, Digital Signature, Public Key Infrastructure, Identity and Access Management, Single Sign-On, Cloud-Based Security Groups, Hardened Virtual Server Images.

MODULE 5: CLOUD SERVICE MODELS (08 Periods)

Cloud Service Models: Software as a Service (SaaS) - Characteristics, Examples and Applications. Platform as a Service (PaaS) - Characteristics, Examples and Applications. Infrastructure as a Service (IaaS) - Characteristics, Examples and Applications.

Case Study: SaaS: Salesforce.com, Facebook.com; PaaS: Google App Engine, MS-Azure and IBM Bluemix; IaaS: Amazon EC2, Amazon S3.

Total Periods: 45

EXPERIENTIAL LEARNING

1. Create a word document of your class time table and store locally and on the cloud with doc, and pdf format.
2. Create a spread sheet which contains employee salary information and calculate gross and total sal using the formula DA=10% OF BASIC HRA=30% OF BASIC PF=10% OF BASIC IF BASIC<=3000 12% OF BASIC IF BASIC>3000 TAX=10% OF BASIC IF BASIC<=1500 =11% OF BASIC IF BASIC>1500 AND BASIC<=2500 =12% OF BASIC IF BASIC>2500 NET_SALARY=BASIC_SALARY+DA+HRA-PF-TAX.
3. Prepare a ppt on cloud computing – Introduction, models, services, and Architectures. PPTs should contain explanations, images and at least 20 slides
4. Create your resume in a neat format using google cloud Programs on PaaS.
5. Create an EC2 instance and invoke Ubuntu operating system with a given set of configuration on amazon web services under IaaS.
6. Create S3 bucket and store a file in it using AWS.
7. Configure web server on Amazon Linux instance with ElasticIP.
8. Generating reports in Salesforce admin.
9. Process of User Management in Salesforce admin.
10. Procedure of Data Management in Salesforce admin.
11. Procedure of providing data security using control access to data using point and click tools in Salesforce admin.

RESOURCES

TEXT BOOKS:

1. Thomas Erl, Zaigham Mahmood, and Ricardo Puttini "CloudComputing-Concepts, Technology & Architecture," Pearson Publication, 2014.
2. Barrie Sosinsky, "Cloud Computing Bible", Wiley India Pvt. Ltd, 2011.
3. George Reese "Cloud Application Architectures," O'Reilly Publications, 2009.

REFERENCE BOOKS:

1. Thomas Erl and Ricardo Puttini "Cloud Computing- Concepts, Technology & Architecture," Pearson, 2013.
2. John W. Rittinghouse, James F. Ransome, "Cloud Computing implementation, Management and Security," CRC Press, ISBN: 9788120341609, Taylor & Francis group, 2010.
3. Barrie Sosinsky, "Cloud Computing Bible," Wiley India Pvt Ltd, 2011.
4. Rajkumar Buyya, James Broberg and Andrzej Goscinski, "Cloud computing principles and paradigms", John Wiley and Sons, 2011.

SOFTWARE/TOOLS:

1. Google App Engine
2. Amazon Web Services
3. Sales Force

VIDEO LECTURES:

1. <https://youtu.be/uroryFU78gM>
2. <https://youtu.be/Pg5nj90xh68>
3. <https://youtu.be/2Dd2ducs-ic>
4. <https://youtu.be/Ijkvx1u0w6o>
5. <https://youtu.be/sMIOsYBMRag>

WEB RESOURCES:

1. www.salesforce.com/tutorial
2. [https:// trailhead.salesforce.com/en/home](https://trailhead.salesforce.com/en/home)
3. <https://MKyong.com/tutorials/google-App-engine-tutorial/>
4. <https://aws.amazon.com>

Course Code	Course Title	L	T	P	S	C
CA25PG0104L	ESSENTIALS OF CYBER SECURITY	2	-	-	-	2

Pre-Requisite -
Anti-Requisite -
Co-Requisite -

COURSE DESCRIPTION: This course provides a detailed discussion on Cyber Security Fundamentals, Cyber Security Fundamentals, Attacker techniques and motivations, Fraud techniques, Threat infrastructure, Exploitation, Malicious code, Defense and analysis techniques, Intrusion detection techniques

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

- CO1.** Understanding the fundamental concepts of cyber security concepts
- CO2.** Identify the pattern of launching attacker and fraud techniques to reduce risk and impact of cyber-attacks.
- CO3.** Identify the vulnerabilities using the SQL injection and web exploitation techniques in a system for securing data.
- CO4.** Apply code obfuscation techniques to prevent any unauthorized party from accessing logic of an application
- CO5.** Apply honey pots and malicious code-naming techniques to defend against attacks in memory.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	-	-	-	-	-	-
CO2	3	2	-	-	-	-	-	-
CO3	3	3	3	-	-	-	-	-
CO4	3	3	3	2	-	-	-	-
CO5	3	2	3	2	-	-	-	-
Course Correlation Mapping	3	3	3	2	-	-	-	-

Correlation Levels: **3: High;** **2: Medium;** **1: Low**

COURSE CONTENT

MODULE 1: CYBER SECURITY FUNDAMENTALS

(05 Periods)

Network Security Concepts: Information assurance fundamentals, Basic cryptography, Symmetric encryption, Public key encryption, The Domain Name System (DNS), Firewalls, Virtualization, Radio-Frequency Identification.

MODULE 2: ATTACKER TECHNIQUES

(07 Periods)

Attacker techniques and motivations: Anti forensics, proxy usage, Tunneling techniques: HTTP, DNS, ICMP, Intermediaries, Steganography and other concepts, Detection and prevention.

Fraud techniques: Phishing, smishing, vishing and mobile malicious code, rogue antivirus, click fraud.

Threat infrastructure: Botnets, Fast Flux, Advanced Fast Flux.

MODULE 3: EXPLOITATION

(06 Periods)

Shellcode, Integer overflow vulnerabilities, Stack based buffer overflows, Format string vulnerabilities, SQL injection, Malicious PDF files, Race conditions, Web exploit tools, DoS conditions, Brute force and dictionary attacks.

MODULE 4: MALICIOUS CODE

(06 Periods)

Worms, viruses, Evading detection and elevating privileges: obfuscation, Virtual Machine obfuscation Persistent software techniques, Token kidnapping, Virtual machine Detection, Rootkits, Spyware, Attacks against privileged user accounts and escalation of privileges, Stealing information and Exploitation.

MODULE 5: DEFENSE AND ANALYSIS TECHNIQUES

(06 Periods)

Importance of memory forensics, Capabilities of memory forensics, Memory analysis frameworks, Dumping physical memory, Installing and using volatility, Finding hidden processes, Volatility analyst pack.

Honeypots, Malicious code naming, Automated malicious code analysis systems, Intrusion detection techniques

Total Periods:30

EXPERIENTIAL LEARNING

1. Observe the firewall settings on your personal computer or smartphone.
 - What configurations are enabled?
 - How does this firewall protect your device from threats?
2. Compare phishing, smishing, and vishing using real-life examples. Which of these do you think people are most vulnerable to, and why?
2. Research a recent DoS attack in the news.
 - What services were affected?
 - What preventive measures could have reduced the impact?
3. Explore your antivirus software logs.
 - What types of threats were blocked recently?
 - Were any of them worms, viruses, or spyware?
4. Reflect on the importance of intrusion detection systems (IDS).
 - How does an IDS differ from a firewall?
 - Why are both needed in an organisation's security framework?

RESOURCES

TEXT BOOKS:

1. James Graham, Richard Howard, Ryan Olson, "Cyber Security Essentials", CRC Press, 2011.
2. Chwan-Hwa(john) Wu, J. David Irwin, "Introduction to Cyber Security", CRC Press T&F Group.

REFERENCE BOOKS:

1. Nina Godbole and SunitBelpure, "Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", Wiley publications.
2. B.B.Gupta, D.P.Agrawal, HaoxiangWang, "Computer and Cyber Security: Principle s, Algorithm, Applications, and Perspectives", CRC Press, ISBN 9780815371335, 2018.

VIDEO LECTURES:

1. <https://nptel.ac.in/courses/106106129>
2. <https://www.coursera.org/professional-certificates/ibm-cybersecurity-analyst>

WEB RESOURCES:

1. <https://www.interpol.int/en/Crimes/Cybercrime>
2. <https://www.geeksforgeeks.org/ethical-hacking/cyber-security-tutorial/>
3. <https://owasp.org/www-project-top-ten/>
4. <https://www.netacad.com/courses/cybersecurity-essentials?courseLang=en-US>

Course Code	Course Title	L	T	P	S	C
CA25PG0105L	MATHEMATICAL FOUNDATIONS FOR COMPUTER APPLICATIONS	3	-	-	-	3

Pre-Requisite -

Anti-Requisite -

Co-Requisite -

COURSE DESCRIPTION: This course addresses the challenges of the relevance of Boolean, Lattice and algebraic systems to computer science and their corresponding systems. To apply number theory, graph theory and their related concepts and algorithm to develop security levels and intelligent systems.

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

- CO1.** Apply mathematical logic and predicate calculus to validate precise statements
- CO2.** Analyze the basic structures of lattice and Boolean algebra.
- CO3.** Demonstrate the importance of algebraic properties with regard to working within various number systems.
- CO4.** Formulate recurring Problems and solve their recurrence relations.
- CO5.** Apply the concepts of graph theory and trees to implement computer Algorithm

CO - PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	-	1	-	-	-	-
CO2	3	3	-	2	2	-	-	-
CO3	3	2	-	2	2	-	-	-
CO4	3	2	-	2	-	-	-	-
CO5	3	3		2				
Course Correlation Mapping	3	3	-	2	2	-	-	-

Correlation Levels: **3: High; 2: Medium; 1: Low**

COURSE CONTENT

MODULE 1: MATHEMATICAL REASONING

(07 Periods)

Concepts of mathematical logic, Normal Forms, The Predicate Calculus, Rule of Inference, Consistency, Proof of Contradiction, quantifiers.

MODULE 2: LATTICES AND BOOLEAN ALGEBRA**(09 Periods)**

Lattices: Partially Ordered Relations, Hasse Diagram, Lattices as Posets, Properties of Lattices.

Boolean Algebra: Basic Definitions, Truth Tables, Boolean Functions, Representation and Minimization of Boolean Functions, Karnaugh Map.

MODULE 3: ALGEBRAIC STRUCTURES AND NUMBER THEORY**(11 Periods)**

Algebraic Structures: Binary Operations and Algebraic Structures Groups, Subgroups, Lagrange's Theorem.

Number Theory: Division Algorithm, The Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Modular Arithmetic (Fermat's Theorem (without proof))

MODULE 4: RECURRENCE RELATIONS**(09 Periods)**

Generating Functions of Sequences, calculating coefficients of generating function, Homogeneous Recurrence relation, solving recurrence relations by substitution and Generating functions, Methods of Characteristic Roots.

MODULE 5: GRAPH THEORY AND ITS APPLICATIONS**(09 Periods)**

Graphs: Representation of Graphs and Graph Isomorphism, Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring.

Trees: Introduction to Trees, Properties of Trees, Applications of Trees, Spanning Trees, Counting trees, Depth-First Search, Breadth-First Search, Minimum Spanning Trees.

Total Periods: 45**EXPERIENTIAL LEARNING**

(This is an indicative. The Course Instructor may modify the activities, and the same shall be reflected in the Course Handout.)

1. Implement Boolean expressions using logic gate simulation software such as Logisim.
2. Explore symmetry operations in geometric shapes and identify group properties.
3. Solve route-planning problems similar to delivery or travel optimization.
4. Develop a simple program to compute GCD and compare efficiency with manual calculations.
5. Conduct a "Detective Investigation" activity where students solve cases using logical deductions.

RESOURCES**TEXT BOOKS:**

1. Kenneth H. Rosen, "Discrete Mathematics and its Applications," Tata McGraw Hill, 8th Edition, 2019.
2. Jon Pierre Fortney, "Discrete Mathematics for Computer Science," CRC Press, Taylor & Francis Group, 1st Edition, 2021.

REFERENCE BOOKS:

1. Richard Johnsonbaugh, "*Discrete Mathematics*," Prentice Hall, 8th Edition, 2019.
2. NarasingDeo, "*Graph Theory with application to Engineering and Computer Science*," Prentice Hall India 2016.
3. J.P. Trembly and R. Manohar, "*Discrete Mathematical Structures with Applications to Computer Science*," Tata McGraw Hill, 37th Edition, 2017

VIDEO LECTURES:

1. <https://nptel.ac.in/courses/106106183>
2. <https://nptel.ac.in/courses/106106094>

WEB RESOURCES:

1. <https://www.coursera.org/learn/discrete-mathematics>
2. <https://people.cs.pitt.edu/~milos/courses/cs441/>
3. <https://web.stanford.edu/class/cs103x/cs103x-notes.pdf>

Course Code**Course Title****L T P S C****CA25PG0204L****DATA MINING**

3 - 2 - 4

Pre-Requisite - DATABASE MANAGEMENT SYSTEMS**Anti-Requisite** -**Co-Requisite** -

COURSE DESCRIPTION: The course covers the basic concepts of data mining and select and apply appropriate data pre-processing techniques, Data Cube Computation Methods and data mining algorithms. Association Rule Mining, Classification, and Clustering methods implemented to infer the valid conclusions in addition to forecasting time series, data mining trends and research frontiers.

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

- CO1.** Analyse pre-processing techniques, Data Cube Technology, data mining algorithms and identify frequent itemsets.
- CO2.** Apply data mining techniques and tools to discover knowledge from text, and multimedia data.
- CO3.** Select and apply Classification and Clustering algorithms on pre-processed datasets to infer predictions effectively.
- CO4.** Apply data mining methods to forecasting time series and provide analytical solutions for societal issues.
- CO5.** Work independently or in team to solve problems with effective communications.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	-	-	-	-	-
CO2	3	3	3	-	3	-	-	-
CO3	2	3	3	-	3	-	-	-
CO4	2	3	3	-	3	-	-	-
CO5	-	-	-	-	-	-	-	-
Course Correlation Mapping	3	3	3	-	3	-	-	-

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

Module 1: DATA MINING AND DATA PREPROCESSING (09 Periods)

Data Mining: Motivated Data Mining, Kinds of Data to mined, kinds of patterns mined, Technologies Used, Kinds of applications targeted, Major issues in Data Mining.

Data Preprocessing: Need for Pre-processing the Data, Major Tasks in Data Preprocessing, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization

Module 2: DATA CUBE TECHNOLOGY AND MINING FREQUENT PATTERNS, ASSOCIATIONS, AND CORRELATIONS (10 Periods)

Data Cube Technology :Data Cube Computation, Data Cube Computation Methods, Multidimensional Data Analysis in Cube Space.

Association Rule Mining: Basic Concepts, Frequent Itemset Mining Methods: The Apriori algorithm for finding frequent itemsets using candidate generation, generating association rules from frequent itemsets, Mining frequent itemsets without candidate generation, Pattern Evaluation Methods

Module 3: CLASSIFICATION (09 Periods)

Classification: Basic Concepts of Classification, Classification by Decision Tree Induction, Bayesian Classification, Rule-based classification, Model Evaluation and Selection, Techniques to Improve Classification Accuracy, Bayesian Belief Networks, Classification by Backpropagation, Support Vector Machines, Classification Using Frequent Patterns

Module 4: CLUSTER ANALYSIS (08 Periods)

Clustering: Cluster Analysis, Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning methods - k-means and k-medoids methods, CLARANS, Hierarchical Methods-Agglomerative and divisive hierarchical clustering, **Density-Based Methods:** DBSCAN, **Grid-Based Methods:** STING, Evaluation of Clustering, Outliers and Outlier Analysis.

Module 5: FORECASTING TIME SERIES AND DATA MINING TRENDS AND RESEARCH FRONTIERS (09 Periods)

Forecasting Time Series: Concepts of time series, Evaluating time series models, Analysis of the components of time series, Exponential smoothing models.

Data Mining Trends and Research Frontiers: Mining Complex Data Types, Data Mining Applications, Data Mining and Society, Data Mining Trends.

Total Periods: 45

EXPERIENTIAL LEARNING

1. Create data sets in ARFF and CSV formats and load into the Weka Explorer.
2. Perform data preprocessing steps on weather nominal and student information data sets as follows:
 - a) Handle missing values for categorical and nominal values.
 - b) Select relevant attributes.
 - c) Apply normalization techniques
3. Generate strong Association rules by applying Apriori algorithm on preprocessed dataset with Min_Sup=60% and Min_Conf=80%.
4. i) Implement the Classification using Decision Tree algorithm on 'Weather' dataset. Draw the confusion matrix and report the model with accuracy.
(ii) Implement Bayesian Classification and analyze the results on 'iris' dataset.
5. Implement simple k-Means clustering algorithm on 'iris' dataset.
6. Use Experimenter WEKA component to rank the performance of j48, oneR, ID3, algorithms on 'weather nominal' dataset.

7. Verify ID3 classifier performance using Gain ratio and Ranker method using a Knowledge flow WEKA component.
8. Implement SVM / Decision tree classification techniques in R tool.
9. Implement clustering techniques in R tool.
10. Visualize data using any plotting framework in R tool.

RESOURCES

TEXT BOOKS:

1. Jiawei Han, Micheline Kamber and Jian Pei, "Data Mining: Concepts and Techniques," Elsevier, 3rd Edition, 2013.
2. G. K. Gupta, "Introduction to Data Mining with Case Studies," Eastern Economy Edition, Prentice Hall of India, 2006.

REFERENCE BOOKS:

1. Ian H. Witten, Eibe Frank, and Mark, "A Data Mining: Practical Machine Learning Tools and Techniques," Hall Morgan Kaufmann, 3rd Edition, 2011.
2. Mining of Massive Datasets by Anand Rajaraman and Jeff Ullman.
3. G. K. Gupta, "Introduction to Data Mining with Case Studies," PHI, New Delhi, 3rd Edition, 2009.
4. Arun K Pujari, "Data Mining Techniques," Universities Press (India) Pvt. Ltd, 2nd Edition, 2001.

SOFTWARE/TOOLS:

1. WEKA
2. R Studio

VIDEO LECTURES:

1. <https://www.youtube.com/watch?v=a4M3GdI5UFY>
2. https://www.youtube.com/watch?v=ykZ-_UGcYWg&list=PLLspfyOYoQcI6Nno3gPkq0h5YSe81hsc
3. https://www.youtube.com/watch?v=ykZ-_UGcYWg&list=PLLspfyOYoQcI6Nno3gPkq0h5YSe81hsc
4. https://www.youtube.com/watch?v=N_whCVtFL9M&list=PLLspfyOYoQcI6Nno3gPkq0h5YSe81hsc&index=4
5. <https://www.youtube.com/watch?v=hEQkqpmTx-Y&list=PLLspfyOYoQcI6Nno3gPkq0h5YSe81hsc&index=5>

WEB RESOURCES:

1. <https://www.cs.waikato.ac.nz/ml/weka/book.html>
2. <http://infolab.stanford.edu/~ullman/mmds/book.pdf>
3. Open source tools for data mining: <http://eprints.fri.uni-lj.si/893/1/2008-OpenSourceDataMining.pdf>
4. <https://www.kaggle.com/datasets>
5. <https://nptel.ac.in/courses/106/105/106105174/>

Course Code	Course Title	L	T	P	S	C
CA25PG0205L	COMPUTER NETWORKS	3	-	2	-	4

Pre-Requisite -

Anti-Requisite -

Co-Requisite -

COURSE DESCRIPTION: This course provides a detailed discussion and hands-on experience on Network topology, Protocols, Physical layer, Error detection and Correction, Medium access, Ethernet, Routing protocols, IP addressing, TCP, UDP, Congestion control, DNS, Email, HTTPS.

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

- CO1.** Interpret different types of network topologies, layers and protocols.
- CO2.** Solve problems related to flow control, error control and congestion control in data transmission.
- CO3.** Evaluate sub netting and routing algorithms for finding optimal paths in networks.
- CO4.** Analyze various congestion control mechanisms and identify appropriate transport layer protocols.
- CO5.** Assess the impact of wired and wireless networks in the context of network protocols such as Ethernet, Bluetooth, DNS, WWW, E-Mail, and Streaming.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	1	-	-	-	-	-	-
CO2	3	2	3	3	-	-	-	-
CO3	3	2	2	3	-	-	-	-
CO4	3	2	2	3	-	-	-	-
CO5	2	-	-	-	-	2	-	-
Course Correlati on Mapping	3	2	2	3	-	1	-	-

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

MODULE 1: INTRODUCTION AND PHYSICAL LAYER (09 Periods)

Introduction: Uses of computer networks, Types, Network technology from local to global, Examples of networks, Network protocols, Reference models.

Physical Layer: Guided transmission media, Wireless transmission, Using the spectrum for transmission, Communication satellites, Comparing different access networks.

MODULE 2: DATA LINK LAYER AND MEDIUM ACCESS CONTROL SUBLAYER (10 Periods)

Data Link Layer: Data link layer design issues, Error detection and correction, Elementary data link protocols.

Medium Access Control Sub layer: ALOHA, Carrier sense multiple access protocols, Collision free protocols. Ethernet, Wireless LANs, Bluetooth, Data link layer switching.

MODULE 3: NETWORK LAYER (10 Periods)

Network layer design issues, Routing algorithms - Shortest path algorithm, Flooding, Distance vector routing, Link state routing, Hierarchical routing, Broadcast routing, Multicast routing, Any cast routing; Traffic management at the network layer, Software Defined Networking (SDN), Network layer in the internet-The IP version-4 protocol, IP addresses, IP version-6, Internet control protocols.

MODULE 4: TRANSPORT LAYER (08 Periods)

UDP – Introduction to UDP, Remote procedure call, Real-time transport protocols; TCP – service model, Protocol, Segment header, Connection establishment, Connection release, Sliding window, Timer management, Congestion control.

MODULE 5: APPLICATION LAYER (08 Periods)

Domain Name System - The DNS Lookup Process, The DNS Name Space and Hierarchy, DNS Queries and Responses, Name Resolution. Electronic mail-Architecture and services, User agent, Message formats, Message transfer, Final delivery; The World Wide Web- Architectural overview, HTTP, HTTPS; Streaming audio and video-Digital Audio, Video, Streaming Stored Media.

Total Periods: 45

EXPERIENTIAL LEARNING

1. Implement the following data link layer framing methods.
 - i) Character count
 - ii) Character stuffing
 - iii) Bit Stuffing
2. Design and develop a program to compute checksum for the given frame 1101011011 using CRC-12, CRC-16, and CRC-CCIP. Display the actual bit string transmitted. Suppose any bit is inverted during transmission. Show that this error is detected at the receiver's end.
3. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.

4. Implement Dijkstra's algorithm to compute the shortest path through a network.
5. Implement distance vector routing algorithm for obtaining routing tables at each node.

Packet Tracer Simulator:

6.
 - a) Create a simple LAN with hubs and switches and monitor the data flow.
 - b) Construct a LAN with single/multiple router and share data to local and remote network.
7. Create and implement a standard access list that blocks the student's section from accessing the server section.
8. Simulate OSPF routing protocol using packet tracer.

NS2 Simulator:

9. Simulate star/mesh network topology used in LAN using TCP/UDP connection.
10. Simulate distance vector/link state routing algorithm.

RESOURCES

TEXT BOOKS:

1. Andrew S. Tanenbaum and David J. Wetherall, Computer Networks, Pearson, 6th Edition, 2022.
2. William Stallings, Data and Computer Communications, 10th Edition, Pearson Education, 2013.

REFERENCE BOOKS:

1. Behrouz A. Forouzan, Data Communications and Networking, McGraw Hill, 5th Edition, 2017.
2. James F. Kurose and Keith W. Ross, Computer Networking: A Top-Down Approach, Pearson, 8th Edition, 2022.

SOFTWARE/TOOLS:

1. Software: J2SDK, Eclipse/Netbeans
2. NS-2, Packet Tracer

VIDEO LECTURES:

1. <http://nptel.ac.in/courses/106105081>
2. <http://nptel.ac.in/courses/106106091>
3. <https://archive.nptel.ac.in/courses/106/105/106105183/>
4. <https://www.youtube.com/watch?v=frUQMHXhnvs>
5. https://www.youtube.com/watch?v=k_lfL93zDNQ

WEB RESOURCES:

1. Virtual Labs (Computer Networks Lab – http://vlabs.iitb.ac.in/vlabs-dev/labs_local/computer-networks/labs/explist.php)
2. Virtual Labs (Advanced Network Technologies Virtual Lab – <http://vlabs.iitkgp.ernet.in/ant>)
3. <https://www.cisco.com/c/en/us/solutions/small-business/resource-center/networking/networking-basics.html>
4. <https://www.ibm.com/in-en/cloud/learn/networking-a-complete-guide>
5. <https://ipcisco.com/cisco-packet-tracer-configuration-examples/>
6. <https://www.isi.edu/nsnam/ns/>
7. https://www.cisco.com/c/dam/global/fi-fi/assets/docs/SMB_University_120307_Networking_Fundamentals.pdf

Course Code	Course Title	L	T	P	S	C
CA25PG0401L	OBJECT ORIENTED PROGRAMMING THROUGH JAVA	3	-	2	4	5
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This course introduces Java programming and object-oriented concepts, covering platform independence, data types, operators, and control structures, along with arrays, strings, methods, recursion, and debugging techniques to enhance problem-solving skills. It emphasizes core object-oriented principles such as classes, objects, inheritance, polymorphism, abstraction, and interfaces. Advanced topics, including exception handling and multithreading are also covered. The course further explores the Java Collection Framework to enable the development of efficient, modular, and reusable code.

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

- CO1.** Demonstrate knowledge of Java fundamentals, including OOP Concepts, JVM, data types, I/O methods and use classes, objects, constructors, and methods to create modular and reusable Java programs.
- CO2.** Apply control structures, arrays, strings methods and recursion to develop modular and efficient Java programs and effectively debug and manage code.
- CO3.** **Develop** object-oriented applications using polymorphism, inheritance, interfaces, abstract classes, and packages to promote code extensibility and reusability.
- CO4.** Utilize string handling methods, exception handling, multithreading, and collection framework classes to solve real-world problems.
- CO5.** Demonstrate effective individual teamwork skills and effectively communicate experimental results and project outcomes.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	-	-	3	-	-	-
CO2	3	2	-	-	3	-	-	-
CO3	3	3	3	3	3	-	-	-
CO4	3	2	2	2	3	-	-	-
CO5	-	-	-	-	-	2	2	2
Course Correlation Mapping	3	3	2	2	2	1	1	1

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

Module 1: INTRODUCTION TO JAVA AND OOP FUNDAMENTALS (10 Periods)

Features of Java, Platform Independence, **Java Environment**, Java Program Structure, Source and Class Files, Data Types, Variables, Type Conversion, Type Casting, Basic Input and Output Methods.

Operators: Arithmetic, Relational, Logical, Assignment.

String Operations: String Length, String Literals, Character Extraction: `charAt()`; String Comparison: `equals()`.

Classes and Objects: Introduction to classes, objects, Constructors, Garbage Collection, this keyword, Access Control, Features of Object Oriented Programming.

Module 2: CONTROL STRUCTURES AND ARRAYS (9 Periods)

Control Structures: Decision-Making Statements: if, if-else, switch; Looping Constructs: for, while, do-while, break, continue.

Arrays: One-dimensional Arrays, Multi-Dimensional Arrays, Storage Layouts, Basic Operations.

Module 3: METHODS AND POLYMORPHISM (8 Periods)

Methods: Method Structure, Parameter Passing, Return Mechanisms, Recursion.

Polymorphism: Method Overloading, Constructor Overloading.

Module 4: INHERITANCE, PACKAGES AND INTERFACES (8 Periods)

Inheritance: Inheritance Basics, Using Super, Multi-Level Hierarchy, Method Overriding, Abstract Classes, Final Keyword.

Packages: Basics, Creating and Accessing a package, CLASSPATH, Importing packages.

Interfaces: Definition, Implementing Interfaces, Nested Interfaces, Applying Interfaces, Variables in Interfaces, Extending Interfaces.

Module 5: EXCEPTION HANDLING, MULTITHREADING, STRING HANDLING AND COLLECTION FRAMEWORK (10 Periods)

Exception Handling: Exception, Types of Exception, Keywords: try, catch, throw, throws and finally, Built-in Exceptions, User Defined Exceptions.

Multithreading: Process, Thread, Thread Model, Creating a thread, Priorities, Thread Synchronization, Inter-thread Communication.

String Handling: String constructors, String Conversion and `toString()`, Search Strings, String Buffer.

Collection Framework: Collection Interface: List, Set; Collection Classes: ArrayList, LinkedList, HashSet, HashMap.

Total Periods: 45

EXPERIENTIAL LEARNING

(This is an indicative. The Course Instructor may modify the activities, and the same shall be reflected in the Course Handout.)

(All source code should be maintained and version-controlled on GitHub)

1) INTRODUCTION TO JAVA AND OOP FUNDAMENTALS

Leetcode: [#2235. Add Two Integers](#)

Given two integers num1 and num2, return the sum of the two integers.

Example 1: Input: num1 = 12, num2 = 5 Output: 17

Example 2: Input: num1 = -10, num2 = 4 Output: -6

Constraints: $-100 \leq \text{num1}, \text{num2} \leq 100$

✓ This problem reinforces the fundamentals of Java arithmetic operators and integer data types. It helps beginners understand how basic expressions are evaluated in Java and builds confidence in writing and testing simple methods.

2) INTRODUCTION TO JAVA AND OOP FUNDAMENTALS

A retail store wants to manage its product inventory using object-oriented principles.

Design a Java program using classes, objects, and constructors to model this scenario.

Program Specifications:

Create a class named **Product** with attributes: String productName, int productId, double price, int quantity.

Include a parameterized constructor to initialise all attributes.

Add a method displayProduct() that prints all product details in a formatted manner.

Add a method updateStock(int units) that increases quantity by the given units and prints the updated stock.

Create an **Inventory** class with a main method that creates at least two Product objects, displays their details, and demonstrates updateStock().

Sample Output:

Product ID: 101

Product Name: Laptop

Price: Rs.55000.0

Quantity in Stock: 10

Updating stock for Laptop by 5 units.

New Stock: 15

✓ This problem reinforces the creation of classes and objects, the use of parameterized constructors, and the concept of encapsulation. It gives students hands-on practice in modelling real-world entities using Java OOP principles.

3) **CONTROL STRUCTURES AND ARRAYS**

Leetcode: #1480. Running Sum of 1d Array

Given an array nums, return the running sum of nums. $\text{runningSum}[i] = \text{sum}(\text{nums}[0] + \text{nums}[1] + \dots + \text{nums}[i])$.

Example 1: Input: nums = [1,2,3,4] Output: [1,3,6,10]

Example 2: Input: nums = [1,1,1,1,1] Output: [1,2,3,4,5]

Example 3: Input: nums = [3,1,2,10,1] Output: [3,4,6,16,17]

Constraints: $1 \leq \text{nums.length} \leq 1000$ $-10^6 \leq \text{nums}[i] \leq 10^6$

✓ This problem strengthens one-dimensional array traversal and in-place accumulation techniques. It helps learners practice iterating through arrays using loops and applying cumulative logic, which is foundational to many data-processing algorithms.

4) **CONTROL STRUCTURES, ARRAYS AND METHODS**

Leetcode: #2798. Number of Employees Who Met the Target

There are n employees, numbered 0 to n-1. Employee i worked hours[i] hours. The company requires at least target hours. Return the count of employees for whom $\text{hours}[i] \geq \text{target}$.

Example 1: Input: hours = [0,1,2,3,4], target = 2 Output: 3

Example 2: Input: hours = [5,1,4,2,2], target = 6 Output: 0

Constraints: $1 \leq n = \text{hours.length} \leq 50$ $0 \leq \text{hours}[i], \text{target} \leq 10^5$

✓ This problem develops array traversal skills combined with conditional logic. It gives students practice in counting elements that satisfy a criterion, which is a fundamental pattern used in searching and filtering operations.

5) **CONTROL STRUCTURES, ARRAYS AND METHODS**

Leetcode: #2011. Final Value of Variable After Performing Operations

A language has four operations on variable X: ++X and X++ increment X by 1; --X and X-- decrement X by 1. Initially X = 0. Given an array of strings operations, return the final value of X.

Example 1: Input: operations = ["--X","X++","X++"] Output: 1

Example 2: Input: operations = ["++X","++X","X++"] Output: 3

Constraints: $1 \leq \text{operations.length} \leq 100$ Each operation is ++X, X++, --X, or X--.

✓ This problem reinforces looping constructs and string comparison in Java. It trains students to iterate over an array, parse string tokens, and apply conditional updates, which are core skills for building control-flow logic in real applications.

6) **CONTROL STRUCTURES, ARRAYS AND METHODS**

Leetcode: #2160. Minimum Sum of Four Digit Number After Splitting Digits

You are given a four-digit positive integer num. Split num into two new integers new1 and new2 using its digits. Leading zeros are allowed; all digits must be used. Return

the minimum possible sum of new1 and new2.

Example 1: Input: num = 2932 Output: 52 (new1=29, new2=23)

Example 2: Input: num = 4009 Output: 13 (new1=04, new2=09)

Constraints: $1000 \leq \text{num} \leq 9999$

✓ This problem deepens understanding of integer digit extraction using the modulo and division operators. It challenges students to apply sorting on a small array and construct numbers from individual digits, reinforcing data-type manipulation skills covered in Module 2.

7) **METHODS AND STRING**

Leetcode: #2283. Check if Number Has Equal Digit Count and Digit Value

You are given a 0-indexed string num of length n. Return true if for every index i in range 0 to n-1, digit i occurs exactly num[i] times in num; otherwise return false.

Example 1: Input: num = "1210" Output: true

Example 2: Input: num = "030" Output: false

Constraints: $n == \text{num.length}$ $1 \leq n \leq 10$ num consists of digits.

✓ This problem combines loop traversal with character-to-integer conversion and frequency counting. It helps students practice iterating over character arrays and strengthens their ability to map problem conditions to array indices, a key skill in algorithmic thinking.

8) **METHODS AND STRING**

Leetcode: #412. Fizz Buzz

Given an integer n, return a string array (1-indexed) where: answer[i] == "FizzBuzz" if i is divisible by 3 and 5; answer[i] == "Fizz" if divisible by 3; answer[i] == "Buzz" if divisible by 5; otherwise answer[i] is i as a string.

Example 1: Input: n = 3 Output: ["1","2","Fizz"]

Example 2: Input: n = 5 Output: ["1","2","Fizz","4","Buzz"]

Constraints: $1 \leq n \leq 10^4$

✓ FizzBuzz solidifies understanding of loop constructs and nested conditional statements. It encourages students to think about operator precedence and condition ordering, which are critical debugging skills for any Java programmer.

9) **POLYMORPHISM AND INHERITANCE**

A geometry application needs to calculate areas of different shapes. Design a Java program demonstrating method overloading to compute the area of a circle, rectangle, and triangle using a single class.

Program Specifications:

Create a class **AreaCalculator** with the following overloaded methods:

double calculateArea(double radius): area of a circle using $\text{Math.PI} * \text{radius} * \text{radius}$.

double calculateArea(double length, double breadth): area of a rectangle.

double calculateArea(double base, double height, String shape): when shape is "triangle", returns $0.5 * \text{base} * \text{height}$.

Create a **Main** class that reads inputs from the user for each shape and displays the calculated area to two decimal places.

Sample Input / Output:

Enter radius: 7.0

Area of Circle: 153.94

Enter length and breadth: 4.0 5.0

Area of Rectangle: 20.00

Enter base and height: 6.0 8.0

Area of Triangle: 24.00

✓ This problem demonstrates compile-time polymorphism through method overloading. Students learn how the Java compiler resolves method calls based on the number and type of arguments, and how overloading makes APIs cleaner and more intuitive.

10) **POLYMORPHISM AND INHERITANCE**

A transportation company wants to model different types of vehicles. Design a Java program using an abstract class and an interface to demonstrate abstraction and runtime polymorphism.

Program Specifications:

Define an interface **Serviceable** with a method void scheduleService().

Create an abstract class **Vehicle** implementing Serviceable with: String vehicleName,

int vehicleId; abstract method void displayInfo(); concrete method void startEngine() printing "[vehicleName] engine started".

Create two concrete classes **Car** and **Truck** that extend Vehicle and implement displayInfo() and scheduleService().

In a **Main** class, create Car and Truck objects using Vehicle references and demonstrate polymorphic method calls.

Sample Output:

Car [ID: 201, Name: Sedan] - Fuel Type: Petrol Sedan engine started. Service scheduled for Sedan at nearest service center.

✓ This problem illustrates runtime polymorphism using abstract classes and interfaces. Students gain practical experience in designing extensible class hierarchies and understand the role of abstraction in building flexible, maintainable software systems.

11) **EXCEPTION HANDLING**

A bank requires a robust account management module that validates deposit and withdrawal transactions. Design a Java program using user-defined exception classes to enforce business rules.

Program Specifications:

Create a custom exception class **InsufficientFundsException** extending Exception. Accept a double shortfall in the constructor and return a meaningful message via getMessage().

Create a custom exception class **InvalidAmountException** extending Exception, thrown when a non-positive amount is provided.

Create a class **BankAccount** with: String accountHolder, double balance; deposit(double amount) throwing InvalidAmountException if amount is 0 or negative; withdraw(double amount) throwing InvalidAmountException for non-positive amounts and InsufficientFundsException for insufficient balance; displayBalance().

In a **Main** class, create a BankAccount, perform transactions based on user input, and handle all exceptions using try-catch-finally.

Sample Output:

Deposited Rs.5000.0 successfully.

Current Balance: Rs.5000.0

Error: Insufficient funds.

Shortfall: Rs.2000.0

✓ This problem gives students practical experience with user-defined exceptions, making them understand how to enforce domain-specific rules through the exception hierarchy. It reinforces the use of try, catch, and finally blocks in building reliable Java applications.

12) **MULTITHREADING**

Leetcode: [#1114. Print in Order](#)

Consider a class with three methods: first(), second(), and third(). The same instance will be passed to three different threads. Thread A calls first(), Thread B calls second(), Thread C calls third(). Design a mechanism to ensure second() executes after first(), and third() executes after second().

Example 1: Input: nums = [1,2,3] Output: firstsecondthird

Example 2: Input: nums = [1,3,2] Output: firstsecondthird

Constraints: nums is a permutation of [1, 2, 3].

✓ This problem introduces students to thread synchronisation and execution order in Java. It bridges classes/objects with concurrent programming and encourages learners to explore Java synchronisation primitives such as Object.wait() and Object.notify().

1) **SAMPLE CONSOLE-BASED MINI PROJECT (PROJECT BASED LEARNING)**

Library Book Management System

Description:

Develop a console-based Library Book Management System in Java that allows a librarian to manage book records. The system should support adding new books, displaying all available books, searching for a book by title or author, issuing a book to a member, and returning an issued book. The system must use object-oriented principles throughout.

Functional Requirements:

1. Add Book: Accept book details (Book ID, Title, Author, Genre, Total Copies) from the console and store them.
2. Display All Books: Show all books with their current available copies.
3. Search Book: Allow searching by title (case-insensitive partial match) or by author name.
4. Issue Book: Reduce available copies by 1 when a member borrows a book; display an error if no copies are available.
5. Return Book: Increase available copies by 1 when a member returns a book.
6. Exit: Terminate the application.

OOP Design Guidelines:

Create a class **Book** with private attributes and public getters/setters to encapsulate book data.

Create a class **Library** that maintains an ArrayList of Book objects and implements all operations as methods.

Create a **Main** class with a menu-driven loop that calls appropriate Library methods based on the user choice.

Use a Scanner for all console input. Apply exception handling for invalid menu choices and input mismatches.

Sample Menu:

1. Add Book
2. Display All Books
3. Search Book
4. Issue Book

Enter your choice:

✓ This project integrates classes, objects, encapsulation, ArrayList-based collections, and menu-driven console I/O into a cohesive application. Students learn to decompose a real-world problem into well-structured classes and gain experience in building an end-to-end Java program with in-session data persistence.

2) **Student Report Card Generator**

Description:

Design and implement a console-based Student Report Card Generator in Java. The system allows a teacher to enter details and marks for multiple students, automatically calculates each student's total, average, percentage, and grade, and prints a formatted report card. The application must be built using object-oriented principles.

Functional Requirements:

1. Input: Accept the student's name, roll number, and marks in five subjects (out of 100) via console.
2. Computation: Calculate total marks, percentage ($\text{total} / 500 * 100$), and assign a grade: A+ (90% and above), A (80% and above), B (70% and above), C (60% and above), D (50% and above), F (below 50%).
3. Report Card: Display a formatted report card showing name, roll number, subject marks, total, percentage, grade, and pass/fail status (pass if percentage is 40% or above).
4. Class Summary: After processing all students, display the class topper (highest percentage) and the class average percentage.
5. The system should support multiple students in a single session using a loop.

OOP Design Guidelines:

Create a class **Student** with private fields for name, rollNumber, and an int[] marks array of size 5; include methods computeTotal(), computePercentage(), and getGrade().

Create a class **ReportCard** with a method generateReport(Student s) that prints the formatted report card.

Create a **Main** class that manages a list of Student objects, collects input, triggers report generation, and computes class-level statistics.

Apply encapsulation (private fields with public methods) and separation of concerns between data handling and report generation.

Sample Output (partial):

===== REPORT CARD =====

Name: Priya Sharma
Roll No: 21CS001
Maths: 88
Science: 79
English: 92
History: 74
CS: 95
Total: 428/500
Percentage: 85.60%
Grade: A
Result: PASS

✓ This project provides a holistic exercise in OOP design: students practice encapsulation using private fields and accessor methods, separation of concerns by splitting data and presentation into different classes, and collection management using ArrayList. The class summary feature introduces basic aggregate computations, preparing students for real-world data-processing tasks.

RESOURCES

TEXT BOOKS:

1. Herbert Schildt, *Java: The Complete Reference*, McGraw-Hill Education, 13th Edition, 2024.
2. Paul Deitel, Harvey Deitel, *Java: How to Program*, Pearson, 11th Edition, 2017.

REFERENCE BOOKS:

1. Cay S. Horstmann, *Core Java Volume I – Fundamentals*, 13th Edition, Pearson, 2025.
2. Kathy Sierra, Bert Bates, *Head First Java*, 2nd Edition, O'Reilly Media, 2005.

SOFTWARE/TOOLS:

1. Java JDK (Java SE 17 or above)
2. IDEs: IntelliJ IDEA / Eclipse / NetBeans

VIDEO LECTURES:

1. <https://nptel.ac.in/courses/106105191>
2. https://www.youtube.com/playlist?list=PLsyebzWxl7pe_IiTfNyr55kwJPWbgxB5
3. https://www.youtube.com/watch?v=xk4_1vDrzzo
4. https://www.youtube.com/playlist?list=PLu0W_9lII9agS67Uits0UnJyrYiXhDS6q
5. <https://www.youtube.com/watch?v=eIrMbaQSU34>

WEB RESOURCES:

1. <https://www.geeksforgeeks.org/java/java/>
2. <https://www.programiz.com/java-programming>
3. https://www.w3schools.com/java/java_oop.asp
4. <https://www.hackerrank.com/domains/java>
5. <https://leetcode.com/problemset/?language=Java>
6. <https://www.codechef.com/practice/java>

Course Code	Course Title	L	T	P	S	C
CA25PG0106L	BUSINESS ANALYTICS	3	-	-	-	3

Pre-Requisite -

Anti-Requisite -

Co-Requisite -

COURSE DESCRIPTION: This course emphasizes on the basic concepts of Business Analytics. It covers the basic excel skills, Excel look up functions for database queries in business analytics. By the end of this course students will acquire basic knowledge to implement statistical methods for performing descriptive, predictive and prescriptive analytics.

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

- CO1.** Understand the basic concepts and models of Business Analytics
- CO2.** Select Suitable basic excel function to perform analytics on spread sheets.
- CO3.** Apply different statistical techniques and distributions for modeling the data
- CO4.** Develop user-friendly Excel applications by using statistical models for effectiveness decision making.
- CO5.** Analyze the performance of different optimization models used in prescriptive analytics on Binary and Categorical data.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	2	1	-	-	-	-	-	-
CO2	2	3	-	-	-	-	-	-
CO3	2	2	-	-	3	-	-	-
CO4	1	1	-	-	-	-	-	-
CO5	-	-	-	-	-	-	-	-
Course Correlation Mapping	2	2	3	-	3	-	-	-

Correlation Level: **3- High** **2-Medium** **1-Low**

COURSE CONTENT

MODULE 1: FOUNDATIONS OF BUSINESS ANALYTICS

(09 Periods)

Introduction, What is Business Analytics, Evolution of Business Analytics, Scope of Business Analytics, Data for Business Analytics, Applications of Business Analytics, Models in Business Analytics, Problem Solving with Analytics.

MODULE 2: ANALYTICS ON SPREADSHEETS

(09 Periods)

Basic Excel Skills, Excel Functions, Using Excel Lookup Functions for Database Queries, Spreadsheet Add-Ins for Business Analytics.

Visualizing and Exploring Data: Data Visualization, Creating Charts In Microsoft Excel, Other Excel Data Visualization, Statistical Methods For Summarizing Data, Exploring Data Using Pivot tables.

MODULE 3: DATA MODELING

(09 Periods)

Basic concepts of Probability, Random Variables and Probability Distributions, Continuous Probability Distributions.

Statistical Sampling, Estimation population parameters, Sampling Error, Sampling Distributions, Hypothesis Testing, ANOVA, Chi Square Test.

MODULE 4: Predictive analytics

(09 Periods)

Trend lines And Regression Analysis, Modeling Relationships And Trends In Data, Simple Linear Regression, Multiple Linear Regression, Building Good Regression Models, Strategies for predictive decision modeling, implementing models on spreadsheets, spreadsheet applications in business analytics, developing user-friendly excel applications, analysing uncertainty and model assumptions, model analysis using analytic solver platform

MODULE 5: Prescriptive analytics

(09 Periods)

Linear Models: Building Linear Models, Implementing Linear Optimization Models On Spreadsheets, Graphical Interpretation Of Linear Optimization, Linear Optimization Models for prediction and Insight.

Integer Models: Solving models with Integer Variables, Integer Optimization Models with Binary Numbers

Decision Analysis: Formulating Decision Problems, Decision Strategies Without Outcome Probabilities, Decision Trees With Outcome Probabilities, Decision Trees.

Total Periods: 45

EXPERIENTIAL LEARNING

1. Diabetic Prediction:

The National Institute of Diabetes and Digestive and Kidney Diseases has created a dataset. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage. The datasets consists of several medical predictor variables and one target variable, Outcome. Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on. Build a machine learning model to accurately predict whether or not the patients in the dataset have diabetes or not?

2. Solve the house price prediction problem using **Linear regression analysis** method. Optimize the parameters of the regression function using gradient descent method.

3. Visualize the decision tree built for solving Heart disease prediction problem and measure the impurity of nodes created via **Decision Tree Analysis**.

Dataset:<https://www.kaggle.com/arviinndn/heart-disease-prediction-uci> dataset/data

4. The data set baby boom (Using R) contains data on the births of 44 children in a one-day period at a Brisbane, Australia, hospital. Compute the skew of the wt variable, which records birth weight. Is this variable reasonably symmetric or skewed?

5. Visualize the **Distribution of data** with different feature scaling methods on online news popularity dataset for article word count.

Dataset:<https://www.kaggle.com/datasets/deepakshende/onlinenewspopularity>

6. Human Activity Recognition System:

The human activity recognition system is a classifier model that can identify human fitness activities. To develop this system, you have to use a smart phone dataset, which contains the fitness activity of 30 people which is captured through smart phones. This

system will help you to understand the solving procedure of the **Multi-classification problem**.

RESOURCES

TEXT BOOKS:

1. James Evans, *Business Analytics*, Pearson Education, 2nd Edition, 2017.

REFERENCE BOOKS:

1. Marc J. Schniederjans, *Business Analytics*, Pearson Education, 2015
2. Camm, Cochran, *Essentials of Business Analytics*, Cengage learning, 2015

VIDEO LECTURES:

1. <https://nptel.ac.in/courses/110105089>
2. <https://archive.nptel.ac.in/courses/110/107/110107092/>
3. <https://nptel.ac.in/courses/110106050>

Web Resources:

1. <https://www.proschoolonline.com/certification-business-analytics-course/what-is-ba>
2. https://michael.hahsler.net/SMU/EMIS3309/slides/Evans_Analytics2e_ppt_01.pdf
3. <https://www.guru99.com/business-analyst-tutorial-course.html>

Course Code	Course Title	L	T	P	S	C
CA25PG0107L	INTERNET OF THINGS	3	-	-	-	3
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This course provides a detailed discussion on Internet of Things components, Domain applications, Communication models, Sensors, Connectivity, Prototyping, Hardware, Design Methodology, Development platforms, Data Analytics for IoT, IoT Security and hands-on experience on IoT applications.

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

- CO1.** Understand IoT Architecture, communication technologies and various applications of IoT.
- CO2.** Analyze sensor technologies and IoT related protocols.
- CO3.** Identify suitable hardware platforms and cloud services for developing IoT applications.
- CO4.** Understand generic design methodology for various IoT applications.
- CO5.** Understand data analytics concepts and security issues in the context of IoT.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	-	-	-	-	-	-
CO2	3	2	-	-	-	-	-	-
CO3	2	2	-	-	-	-	-	-
CO4	2	2	3	2	-	-	-	-
CO5	2	2	-	2	3	-	-	-
Course Correlation Mapping	2	3	3	2	2	-	-	-

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

MODULE 1: INTRODUCTION AND DOMAIN APPLICATIONS (09 Periods)

Introduction to Internet of Things: Definition, Conceptual Framework, Architectural View, Technology behind IoT, Communication Technologies, Data Enrichment, Data consolidation and Device management at Gateway.

IoT and M2M: M2M, Difference between IoT and M2M, SDN and NFV for IoT.

Domain Specific IoTs: Home automation, Cities, Environment, Health and Life Style.

MODULE 2: SENSORS AND CONNECTIVITY (09 Periods)

Sensors: Sensor Technology, Actuators, RFID Technology.

Connectivity: Internet Connectivity, Internet-Based Communications, IP Addressing in the IoT, Medium Access Control, Application Layer Protocols.

MODULE 3: PROTOTYPING AND HARDWARE (09 Periods)

Prototyping: Embedded Computing Basics, Embedded platforms for prototyping.

Hardware: Things always connected to the Internet/Cloud, Amazon Web Services for IoT.

MODULE 4: DESIGN METHODOLOGY AND CASE STUDIES (09 Periods)

Design Methodology: Purpose and Requirements specifications, Process Specifications, Domain Model Specification, Information Model Specification, Service Specification, IoT Level Specifications, Functional View Specification, Operational View Specification, Device and Component integration, Application development.

Case Studies Illustrating IoT Design: Home Automation, Cities.

MODULE 5: DATA ANALYTICS AND SECURITY (09 Periods)

Data Analytics for IoT: Apache Hadoop, Using Hadoop MapReduce for Batch Data Analysis. **IoT Security:** Vulnerabilities, Security Requirements and Threat analysis, Security Tomography and Layered Attacker Model, Identity Management and Establishment, Access Control and Secure Message Communication, Security Models, Profiles and Protocols for IoT.

Total Periods: 45

EXPERIENTIAL LEARNING

1. a) Design and Simulate LED 7-Segment Display interfacing with Arduino.
b) Design and Simulate Servo motor interfacing with Arduino.
2. a) Design and Simulate ultrasonic sensor and LCD interfacing with Arduino.
b) Design and Simulate Flame Sensor interfacing with Arduino.
3. Design and Implement to capture Gas Sensor and send sensor data to cloud from your NodeMCU device using Arduino IDE.
4. Design and Implementation of Humidity and Temperature Monitoring Using Arduino and upload data to cloud using MQTT.

RESOURCES

TEXT BOOKS:

1. Arshdeep Bahga, Vijay Madisetti, "*Internet of Things – A hands-on approach*," University Press, 2015.
2. Raj Kamal, "*Internet of Things- Architecture and Design Principles*," Mc Graw Hill, 2017.

REFERENCE BOOKS:

1. Adrian McEwen and Hakim Cassimally, "*Designing the Internet of Things*," Wiley Publishing, 2013.
2. Charles Bell, "*Beginning Sensor Networks with Arduino and Raspberry Pi*," Apress, 2013.
3. Marco Schwartz, "*Internet of Things with the Arduino Yun*," Packt Publishing, 2014.
4. Matt Richardson, Shawn Wallace, "*Getting Started with Raspberry Pi*," Maker Media, Inc, 2012.

SOFTWARE/TOOLS:

1. Software: Arduino IDE/ Raspberry Pi OS

VIDEO LECTURES:

1. <https://nptel.ac.in/courses/106105166>

WEB RESOURCES:

1. https://www.tutorialspoint.com/internet_of_things/index.htm
2. <https://www.javatpoint.com/iot-internet-of-things>

COURSE DESCRIPTION: Distributed Systems, Blockchain, Types of blockchains, Decentralization, Bitcoin, Alternative Coins, Smart Contracts, Ethereum 101, Applications of Ethereum, Scalability Challenges in Privacy and Security, Current Landscape.

- CO1.** Understand the concepts of distributed Systems, decentralization and blockchain technology for the development of an application.
- CO2.** Select and use the Bitcoins, Smart Contracts and Ethereum 101 for the development of distributed systems and decentralized systems.
- CO3.** Analyze the issues and challenges of scalability, privacy and security in monetizing the businesses using blockchain technology.
- CO4.** Develop and deploy the applications using BlockApps and Eris platforms.
- CO5.** Commit to ethics and cyber regulations to perform encryption, multiparty computation, smart governance using Smart contract security.

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
C01	3	3	1	-	-	-	-	-
C02	1	3	2	-	2	-	2	-
C03	1	3	-	2	-	-	-	-
C04	1	2	3	2	-	-	-	2
C05	-	-	-	-	-	-	-	-
Course Correlation Mapping	2	3	2	2	2	-	2	2
Correlation Level:		3- High		2-Medium		1-Low		

COURSE CONTENT

MODULE 1: DISTRIBUTED SYSTEMS AND DECENTRALIZATION (10 Periods)

Distributed Systems: CAP Theorem, Byzantine Generals Problem, Consensus, History of blockchain, Generic Elements, Features, Applications, Tiers, Types of blockchain, CAP Theorem & blockchain, Benefits and limitations of blockchain.

Decentralization: Decentralization using blockchain, Methods of decentralization, Routes to Decentralization, blockchain and full ecosystem decentralization, Smart contract, Decentralized organizations, Decentralized autonomous organizations, Decentralized autonomous corporations, Decentralized autonomous societies, Decentralized applications, Platforms for decentralization.

MODULE 2: DIGITAL CURRENCY (10 Periods)

Bitcoin: Concepts, Transactions, Blockchain, Bitcoin Payments.

Alternative Coins: Theoretical foundations, Bitcoin limitations, Namecoin, Litecoin, Primecoin, Zcash.

MODULE 3: SMART CONTRACTS AND ALTERNATIVE BLOCKCHAINS (08 Periods)

Smart Contracts: History, Ricardian contracts, Smart Contract Templates, Oracles, Smart Oracles, Deployment of Smart Contracts on Blockchains.

Alternative Blockchains: Kadena; Platforms: BlockApps, Eris

MODULE 4: ETHEREUM 101 (08 Periods)

Ethereum: Ethereum blockchain, Elements of Ethereum blockchain, Precompiled contracts, Accounts, Block, Ether, Messages, Mining, Clients and wallets, Ethereum network, Applications developed on Ethereum, Scalability and Security Issues.

MODULE 5: CHALLENGES AND CURRENT LANDSCAPE (09 Periods)

Scalability and Other Challenges: **Scalability:** Block size increase, Block internal reduction, Invertible Bloom lookup tables, Sharding, State channels, Private blockchain, Proof of Stake.

Privacy: Indistinguishability obfuscation, Homomorphic encryption, Zero knowledge proofs, State channels, Secure multiparty computation, Usage of hardware to provide confidentiality, Coinjoin, Confidential transactions, MumbleWimble;

Security: Smart contract security.

Current Landscape: Emerging Trends, Improvement proposals; Blockchain research: Smart contracts, Centralization issues, Limitations in cryptographic functions, Consensus Algorithms, Scalability, Code Obfuscation.

Total Periods: 45

EXPERIENTIAL LEARNING

1. Use of blockchain technology in different sectors.
2. Application of blockchain technology in higher education.
3. Impact of blockchain technology on economy.

RESOURCES

TEXT BOOK:

1. Imran Bashir, "Mastering Blockchain," Packt Publishing Ltd., 2017

REFERENCE BOOKS:

1. Arshdeep Bahga, Vijay Madisetti, "Blockchain Applications: A Hands-On Approach," VPT Books, 2017.
2. Josh Thompsons, "Blockchain: The Blockchain for Beginners Guide To Blockchain Technology and Leveraging Blockchain Programming," Create Space Independent Publishing Platform, 2017.

VIDEO LECTURES:

1. <https://nptel.ac.in/courses/106105184/>
2. <https://medium.com/moatcoin/part-6-blockchain-simplified-notes-nptel-892f13875555>

WEB RESOURCES:

1. <http://www.hands-on-books-series.com/assets/Bahga-Madisetti-Blockchain-BookCode.zip>

Course Code	Course Title	L	T	P	S	C
CA25PG110L	INTERNSHIP	-	-	-	-	2
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: Expose students to the industrial environment; Create competent professionals for the industry; sharpen the real time technical / managerial skills required at the job; Gain professional experience and understand engineer's responsibilities and ethics; Familiarize with latest equipment, materials and technologies; Gain exposure to technical report writing; Gain exposure to corporate working culture.

COURSE OUTCOMES: After successful completion of this course, the students will be able to:

- CO1.** Analyze latest equipment, materials and technologies that are used in industry to solve complex engineering problems following relevant standards, codes, policies and regulations.
- CO2.** Analyze safety, health, societal, environmental, sustainability, economical and managerial factors considered in industry in solving complex engineering problems.
- CO3.** Perform individually or in a team besides communicating effectively in written, oral and graphical forms on practicing engineering.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	-	3	3	-	-	3
CO2	-	3	-	-	-	3	3	-
CO3	-	-	-	-	-	-	-	-
Course Correlation Mapping	1	3	-	1	1	1	1	1

Correlation Level: 3-High; 2-Medium; 1-Low

Course Code	Course Title	L	T	P	S	C
CA25PG0109L	OPERATING SYSTEMS	3	-	-	-	3
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This course provides theoretical and practical knowledge on process management, synchronization, handling deadlocks, memory management, storage, File system, I/O System. It also emphasizes the mechanisms for access control and security.

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

- CO1.** Analyze performance of CPU scheduling algorithms.
- CO2.** Design solutions for process synchronization problems by using semaphores and monitors.
- CO3.** Devise solutions for deadlocks using deadlock handling mechanism.
- CO4.** Solve memory management problems, file allocation using page replacement and disk scheduling algorithms.
- CO5.** Analyze services of I/O subsystems and mechanisms of security & protection

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	2	2	2	-	-	-
CO2	3	2	3	-	2	-	-	-
CO3	3	2	3	-	2	-	-	-
CO4	3	2	3	-	2	-	-	-
CO5	3	3	-	3	2	-	-	-
Course Correlation Mapping	3	2.4	3	3	2	-	-	-

Correlation Levels: 1: Low 3: High; 2: Medium

COURSE CONTENT

MODULE 1: INTRODUCTION TO OPERATING SYSTEM (09 Periods) **PROCESS MANAGEMENT**

Introduction: Definition, Operating System Structure and Services, System Calls.

Process Management: Process Scheduling, Process Control Block, Inter Process Communication, Threads, Multithreading Models, CPU Scheduling Criteria, Scheduling Algorithms, Multiprocessor Scheduling.

MODULE 2: PROCESS SYNCHRONIZATION AND DEADLOCKS (09 Periods)

Process Synchronization: Critical Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Synchronization Problems, Monitors.

Deadlocks: System Model, Deadlock characterization, Methods for handling deadlocks, Prevention, Detection, Avoidance, Recovery from deadlock.

MODULE 3: MEMORY MANAGEMENT AND SECONDARY STORAGE (09 Periods)

Memory Management: Contiguous Allocation, Paging, Segmentation, Segmentation with Paging

Virtual Memory: Demand Paging, Page Replacement Algorithms, Copy-on-Write, Thrashing.

Secondary Storage Structure: Overview of Mass Storage Structure, Disk Structure, Disk Scheduling.

MODULE 4: FILE AND I/O SYSTEMS (09 Periods)

File System: File concept, Access Methods, Directory Structure, File System Structure, i- node, File System Implementation, Directory Implementation, Allocation Methods.

I/O System: I/O Hardware, Application I/O Interface, Kernel I/O subsystem.

MODULE 5: PROTECTION AND SECURITY (09 Periods)

Protection: Goals, Principles, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights.

Security: Program Threats, System and Network Threats, User Authentication, Implementing Security Defenses, Firewalling to Protect Systems and Networks, Computer-Security Classifications.

Total Periods: 45

EXPERIENTIAL LEARNING

1. Write a program to implement Process System Calls and also Write a program to implement I/O System Calls.
2. Write a program to simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS b) SJF c) Round Robin d) Priority
3. Write a program to simulate multi-level queue scheduling algorithm considering the following scenario. All the processes in the system are divided into two categories – system processes and user processes. System processes are to be given higher priority than user processes. Use FCFS scheduling for the processes in each queue
4. Write a program to simulate producer-consumer problem using semaphores
5. Design Banker's Algorithm for Deadlock Avoidance. Find the safe sequence. If Maximum request of any one process is changed, detect whether a deadlock has occurred or not. Consider the number of resources are three and Jobs are five.
6. Write a program to simulate the MVT and MFT memory management techniques
7. Write a program to simulate the following contiguous memory allocation techniques Worst-fit b) Best-fit c) First-fit
8. Implement the following Page Replacement Algorithms
a) FIFO b) LFU c) LRU d) Optimal
9. Write a program to simulate disk scheduling algorithms
a) FCFS b) SSTF c) SCAN d) CSCAN
10. Write a program to simulate the following file allocation strategies.
a) Sequential b) Indexed c) Linked

RESOURCES

TEXT BOOK

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, Operating System Concepts, Wiley India Edition, 10th Edition, 2018.

REFERENCE BOOKS:

1. William Stallings, Operating Systems, Internals and Design Principles, Pearson Education, 7th Edition, 2013.
2. Andrew S. Tanenbaum, Modern Operating Systems, PHI, 3rd Edition, 2009.

Software: Dev C++

VIDEO LECTURES:

1. https://onlinecourses.nptel.ac.in/noc21_cs44/preview
2. <https://www.edx.org/learn/operating-systems>
3. <https://www.cse.iitb.ac.in/~mythili/virtcc/>
- 4 <https://www.youtube.com/channel/UCCsCO5D7PHqAqVLFM5aKAHA>

WEB RESOURCES:

1. http://www.sncwgs.ac.in/wp-content/uploads/2015/11/operating_system_tutorial.pdf
2. https://www.tutorialspoint.com/operating_system/index.html

Course Code	Course Title	L	T	P	S	C
CA25PG0206L	DATA STRUCTURES AND ALGORITHMS WITH PYTHON	3	-	2	-	4
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This course offers a comprehensive understanding and hands-on experience of core data structures and algorithms. It covers linked lists, stacks, queues, trees, graphs, searching and sorting methods, heaps, and hashing techniques. The course focuses on implementing these structures, analyzing their performance, and applying them to real-world problems such as expression evaluation, graph traversal, data organization, and efficient data retrieval using hashing.

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

- CO1.** Demonstrate knowledge of basic and advanced data structures such as linked lists, stacks, queues, trees, graphs, and their applications.
- CO2.** Apply searching and sorting algorithms to organize and retrieve data efficiently.
- CO3.** Develop optimized algorithms and solutions using tree and graph data structures to effectively address complex problems in data organization and pathfinding.
- CO4.** Design solutions using heap and hashing to handle prioritized tasks and for quick data access.
- CO5.** Work independently to solve problems with effective communication.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	-	-	3	-	-	-
CO2	3	2	-	-	3	-	-	-
CO3	3	3	3	3	3	-	-	-
CO4	3	2	2	2	3	-	-	-
CO5	-	-	-	-	-	2	2	2
Course Correlation Mapping	3	3	2	2	2	1	1	1

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

Module 1: LINKED LIST, STACK, AND QUEUE (10 Periods)

Singly linked list, doubly linked list, circular linked list, node structure, dynamic memory allocation, Linked list operations: insertion, deletion, traversal, searching in various types; Stack ADT, LIFO principle, push, pop, peek operations, array and linked list implementation, expression evaluation, parentheses matching, function call tracking, Queue ADT, FIFO principle, operations: enqueue, dequeue, front, circular queue, priority queue.

Module 2: BINARY TREE, BINARY SEARCH TREE, AND AVL TREE (10 Periods)

Binary tree structure, types: full, complete, perfect, nodes, height, depth terminology; Tree traversals: preorder, inorder, postorder using recursion and iteration; Binary Search Tree (BST): properties, insertion, deletion, searching; AVL tree basics, balance factor, left and right rotations, self-balancing mechanism; Applications in efficient searching, sorting, dynamic data organization.

Module 3: GRAPHS (10 Periods)

Graphs: vertices, edges, degree, types: directed, undirected, weighted, unweighted; Graph representations: adjacency matrix, adjacency list, trade-offs and use cases; Graph traversal algorithms: BFS, DFS and their applications; Shortest path algorithms: introduction to Dijkstra's, Bellman-Ford, Cycle detection methods, checking connectivity in graphs, Topological sorting.

Module 4: SEARCHING AND SORTING (8 Periods)

Linear search, binary search: working principles, implementation, time complexity analysis, Sorting algorithms : bubble sort, selection sort, insertion sort, merge sort, quick sort, ; Advanced sorting: counting sort, radix sort and when to apply them; Comparison: stability, in-place vs out-of-place, time and space complexity analysis; Applications in real-world data organization and optimization problems.

Module 5: HEAPS AND HASHING (7 Periods)

Heap types: max-heap, min-heap, heap properties, array-based representation; Heap operations: insertion, deletion, heapify, building heap from unsorted array; Priority queues, heap sort algorithm and real-time scheduling examples; Hashing: hash functions, hash tables, use cases for constant-time retrieval; Collision handling: separate chaining, linear probing, quadratic probing, double hashing.

Total Periods: 45

EXPERIENTIAL LEARNING

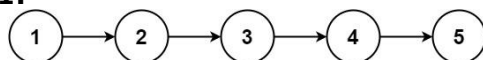
(All source code should be maintained and version-controlled on GitHub)

1) LINKED LIST, STACK, AND QUEUE

Leetcode: [#61. Rotate List](#)

Given the head of a linked list, rotate the list to the right by k places.

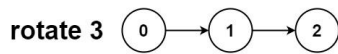
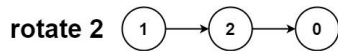
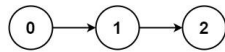
Example 1:



Input: head = [1,2,3,4,5], k = 2

Output: [4,5,1,2,3]

Example 2:



Input: head = [0,1,2], k = 4

Output: [2,0,1]

Constraints:

- The number of nodes in the list is in the range [0, 500].
- $-100 \leq \text{Node.val} \leq 100$
- $0 \leq k \leq 2 * 10^9$

✓ This problem strengthens understanding of linked list manipulation and handling circular rotations. It encourages learners to optimize traversal using modulo operations and manage edge cases like empty lists or large k values. Helps build logic around pointer updates and real-world cyclic data structures.

1) LINKED LIST, STACK, AND QUEUE

LeetCode: [#1249. Minimum Remove to Make Valid Parentheses](#)

Given a string s of '(', ')', and lowercase English characters.

Your task is to remove the minimum number of parentheses ('(' or ')', in any positions) so that the resulting *parentheses string* is valid and return **any** valid string.

Formally, a *parentheses string* is valid if and only if:

- It is the empty string, contains only lowercase characters, or
- It can be written as AB (A concatenated with B), where A and B are valid strings, or
- It can be written as (A), where A is a valid string.

Example 1:

Input: s = "lee(t(c)o)de"

Output: "lee(t(c)o)de"

Explanation: "lee(t(co)de)" , "lee(t(c)ode)" would also be accepted.

Example 2:

Input: s = "a)b(c)d"

Output: "ab(c)d"

Example 3:

Input: s = "))(("

Output: ""

Explanation: An empty string is also valid.

Constraints:

- $1 \leq s.length \leq 10^5$
- $s[i]$ is either '(', ')', or lowercase English letter.

✓ This problem enhances understanding of stack-based validation and greedy algorithms for maintaining syntactic correctness. It helps learners manage open-close matching logic and handle unbalanced expressions efficiently.

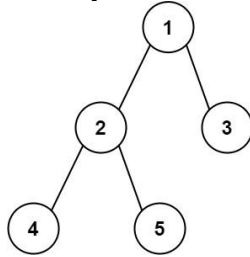
2) BINARY TREE, BINARY SEARCH TREE, AND AVL TREE

Leetcode: #543. Diameter of Binary Tree

Given the root of a binary tree, return *the length of the **diameter** of the tree*.

The **diameter** of a binary tree is the **length** of the longest path between any two nodes in a tree. This path may or may not pass through the root.

The **length** of a path between two nodes is represented by the number of edges between them.

Example 1:

Input: root = [1,2,3,4,5]

Output: 3

Explanation: 3 is the length of the path [4,2,1,3] or [5,2,1,3].

Example 2:

Input: root = [1,2]

Output: 1

Constraints:

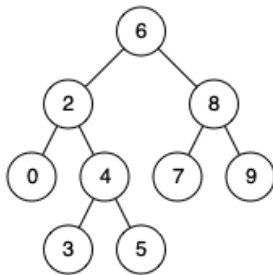
- The number of nodes in the tree is in the range [1, 10⁴].
- -100 ≤ Node.val ≤ 100

✓ This problem helps learners understand tree traversal, particularly depth-first search (DFS). It develops logic to compute height of subtrees and track maximum path length between any two nodes. Ideal for mastering recursive strategies and handling tree-based interview problems.

3) BINARY TREE, BINARY SEARCH TREE, AND AVL TREE**Leetcode: #235. Lowest Common Ancestor of a Binary Search Tree**

Given a binary search tree (BST), find the lowest common ancestor (LCA) node of two given nodes in the BST.

According to the [definition of LCA on Wikipedia](#): "The lowest common ancestor is defined between two nodes p and q as the lowest node in T that has both p and q as descendants (where we allow **a node to be a descendant of itself**)."

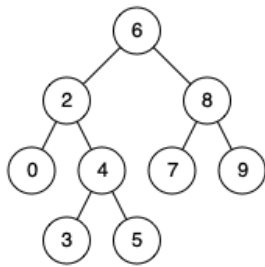
Example 1:

Input: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 8

Output: 6

Explanation: The LCA of nodes 2 and 8 is 6.

Example 2:



Input: root = [6,2,8,0,4,7,9,null,null,3,5], p = 2, q = 4

Output: 2

Explanation: The LCA of nodes 2 and 4 is 2, since a node can be a descendant of itself according to the LCA definition.

Example 3:

Input: root = [2,1], p = 2, q = 1

Output: 2

Constraints:

- The number of nodes in the tree is in the range [2, 10⁵].
- -10⁹ ≤ Node.val ≤ 10⁹
- All Node.val are **unique**.
- p ≠ q
- p and q will exist in the BST.

✓ This problem reinforces BST traversal techniques and deepens understanding of tree properties. It helps learners identify how value-based decisions can guide efficient traversal paths. Useful for building foundational logic for divide-and-conquer problems in trees and graphs.

4) GRAPHS AND GRAPH ALGORITHMS

LeetCode: [#210. Course Schedule II](#)

There are a total of numCourses courses you have to take, labeled from 0 to numCourses - 1. You are given an array prerequisites where prerequisites[i] = [a_i, b_i] indicates that you **must** take course b_i first if you want to take course a_i.

- For example, the pair [0, 1], indicates that to take course 0 you have to first take course 1.

Return *the ordering of courses you should take to finish all courses*. If there are many valid answers, return **any** of them. If it is impossible to finish all courses, return **an empty array**.

Example 1:

Input: numCourses = 2, prerequisites = [[1,0]]

Output: [0,1]

Explanation: There are a total of 2 courses to take. To take course 1 you should have finished course 0. So the correct course order is [0,1].

Example 2:

Input: numCourses = 4, prerequisites = [[1,0],[2,0],[3,1],[3,2]]

Output: [0,2,1,3]

Explanation: There are a total of 4 courses to take. To take course 3 you should have finished both courses 1 and 2. Both courses 1 and 2 should be taken after you finished course 0.

So one correct course order is [0,1,2,3]. Another correct ordering is [0,2,1,3].

Example 3:

Input: numCourses = 1, prerequisites = []

Output: [0]

Constraints:

- $1 \leq \text{numCourses} \leq 2000$
- $0 \leq \text{prerequisites.length} \leq \text{numCourses} * (\text{numCourses} - 1)$
- $\text{prerequisites}[i].\text{length} == 2$
- $0 \leq a_i, b_i < \text{numCourses}$
- $a_i \neq b_i$
- All the pairs $[a_i, b_i]$ are **distinct**.

✓ This problem strengthens your understanding of topological sorting, especially using BFS (Kahn's algorithm) or DFS. It helps learners grasp cycle detection in directed graphs, critical for scheduling and dependency resolution problems.

5) **GRAPHS AND GRAPH ALGORITHMS**

Leetcode: [#200. Number of Islands](#)

Given an $m \times n$ 2D binary grid `grid` which represents a map of '1's (land) and '0's (water), return *the number of islands*.

An **island** is surrounded by water and is formed by connecting adjacent lands horizontally or vertically. You may assume all four edges of the grid are all surrounded by water.

Example 1:

Input: `grid = [`
`["1","1","1","1","0"],`
`["1","1","0","1","0"],`
`["1","1","0","0","0"],`
`["0","0","0","0","0"]`
`]`

Output: 1

Example 2:

Input: `grid = [`
`["1","1","0","0","0"],`
`["1","1","0","0","0"],`
`["0","0","1","0","0"],`
`["0","0","0","1","1"]`
`]`

Output: 3

Constraints:

- $m == \text{grid.length}$
- $n == \text{grid}[i].\text{length}$
- $1 \leq m, n \leq 300$
- $\text{grid}[i][j]$ is '0' or '1'.

✓ This problem enhances understanding of graph traversal techniques like DFS or BFS in 2D matrices. It builds foundational skills for flood fill algorithms, useful in image processing and geographic data analysis.

6) **SEARCHING AND SORTING**

Leetcode: [#35. Search Insert Position](#)

Given a sorted array of distinct integers and a target value, return the index if the target is found. If not, return the index where it would be if it were inserted in order. You must write an algorithm with $O(\log n)$ runtime complexity.

Example 1:

Input: `nums = [1,3,5,6], target = 5`

Output: 2

Example 2:

Input: nums = [1,3,5,6], target = 2

Output: 1

Example 3:

Input: nums = [1,3,5,6], target = 7

Output: 4

Constraints:

- $1 \leq \text{nums.length} \leq 10^4$
- $-10^4 \leq \text{nums}[i] \leq 10^4$
- nums contains **distinct** values sorted in **ascending** order.
- $-10^4 \leq \text{target} \leq 10^4$

✓ This problem strengthens understanding of binary search, a fundamental divide-and-conquer technique. It trains learners to efficiently handle insert position logic within sorted data structures. The concept is widely applicable in search engines, database indexing, and algorithm optimization tasks.

7) **SEARCHING AND SORTING**

Leetcode: [#324. Wiggle Sort II](#)

Given an integer array nums, reorder it such that $\text{nums}[0] < \text{nums}[1] > \text{nums}[2] < \text{nums}[3] \dots$

You may assume the input array always has a valid answer.

Example 1:

Input: nums = [1,5,1,1,6,4]

Output: [1,6,1,5,1,4]

Explanation: [1,4,1,5,1,6] is also accepted.

Example 2:

Input: nums = [1,3,2,2,3,1]

Output: [2,3,1,3,1,2]

Constraints:

- $1 \leq \text{nums.length} \leq 5 * 10^4$
- $0 \leq \text{nums}[i] \leq 5000$
- It is guaranteed that there will be an answer for the given input nums.

✓ This problem develops an understanding of array rearrangement patterns, particularly wiggle sort. It enhances knowledge of median finding, index mapping, and three-way partitioning techniques. Useful in scenarios where alternating highs and lows are required, such as stock price modeling or load balancing.

8) **HEAPS AND HASHING**

Leetcode: [#242. Valid Anagram](#)

Given two strings s and t, return true if t is an anagram of s, and false otherwise.

Example 1:

Input: s = "anagram", t = "nagaram"

Output: true

Example 2:

Input: s = "rat", t = "car"

Output: false

Constraints:

- $1 \leq \text{s.length}, \text{t.length} \leq 5 * 10^4$
- s and t consist of lowercase English letters.

✓ This problem strengthens skills in string manipulation and frequency counting. It helps learners efficiently compare character distributions using hash maps or arrays. An essential concept for solving problems related to permutations, anagrams, and string matching.

9) HEAPS AND HASHING

Leetcode: [#49. Group Anagrams](#)

Given an array of strings `strs`, group the anagrams together. You can return the answer in **any order**.

Example 1:

Input: `strs = ["eat", "tea", "tan", "ate", "nat", "bat"]`

Output: `[["bat"], ["nat", "tan"], ["ate", "eat", "tea"]]`

Explanation:

- There is no string in `strs` that can be rearranged to form "bat".
- The strings "nat" and "tan" are anagrams as they can be rearranged to form each other.
- The strings "ate", "eat", and "tea" are anagrams as they can be rearranged to form each other.

Example 2:

Input: `strs = [""]`

Output: `[[""]]`

Example 3:

Input: `strs = ["a"]`

Output: `[["a"]]`

Constraints:

- $1 \leq \text{strs.length} \leq 10^4$
- $0 \leq \text{strs}[i].\text{length} \leq 100$
- `strs[i]` consists of lowercase English letters.

✓ This problem enhances skills in hashing and grouping techniques by organizing strings based on sorted keys or frequency maps. It deepens understanding of how to identify anagrams efficiently by leveraging string transformations. Key for mastering problems involving categorization, data grouping, and string manipulation.

RESOURCES

TEXT BOOKS:

1. Data Structures and Algorithms in Python – Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser (2013)
2. Problem Solving with Algorithms and Data Structures Using Python – Bradley N. Miller, David L. Ranum (2011)

REFERENCE BOOKS:

1. Python Algorithms: Mastering Basic Algorithms in the Python Language – Magnus Lie Hetland (2014)
2. Data Structures and Algorithms Made Easy” by Narasimha Karumanchi

SOFTWARE/TOOLS:

1. Python 3.10

2. Eclipse IDE / Jupyter Notebook / Google Colab / VS Code / Online IDE

VIDEO LECTURES:

1. https://www.youtube.com/@abdul_bari
2. https://www.youtube.com/watch?v=AT14ICXuMKI&list=PLdo5W4Nhv31bbKJzrsKfMpo_grxuLI8LU
3. <https://www.youtube.com/watch?v=RBSGKIAvoiM&list=PLqM7aHxFySGg7t9DoeGvZsPmkGz1WfKc>
4. <https://www.youtube.com/c/TakeUForward/playlists>
5. <https://www.youtube.com/watch?v=RBSGKIAvoiM&pp=ygUYZGF0YSBzdHJ1Y3R1cmVzIHR1dG9yaWFs>
6. https://www.youtube.com/watch?v=VTLCohNyACE&list=PLfqMhTWNBTte137I_EPQd34TsgV6IO55pt&pp=0gcJCV8EOCosWNin
7. <https://www.youtube.com/@CodingWithTamil>

WEB RESOURCES:

1. <https://www.geeksforgeeks.org/data-structures/>
2. <https://www.programiz.com/dsa>
3. https://www.tutorialspoint.com/data_structures_algorithms/index.htm
4. <https://visualgo.net/en> (for visualizing trees, graphs, heaps, etc.)
5. <https://www.interviewbit.com/data-structure-interview-questions/>

Course Code	Course Title	L	T	P	S	C
CA25PG0207L	MACHINE LEARNING	3	-	2	-	4
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This course provides a detailed discussion and hands-on experience on Introduction to machine learning, Bayesian concept learning, Supervised learning, Unsupervised learning, Artificial neural networks, Ensemble learning.

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

- CO1.** Analyze the process of machine learning modeling and evaluation to automatically infer a general description for a given learning problem.
- CO2.** Analyze the underlying mathematical models within machine learning algorithms and learning tasks.
- CO3.** Design and implement machine learning solutions for classification, regression, and clustering problems.
- CO4.** Design and implement efficient neural architectures to model patterns for a given learning problem.
- CO5.** Develop intelligent solutions to solve societal problems related to computer vision, information security, healthcare and other areas.
- CO6.** Work Independently to solve problems with effective communication.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	1	3	3	1	3	3	-	-
CO2	2	-	2	2	-	2	-	-
CO3	-	3	2	-	3	2	-	-
CO4	2	-	2	2	-	2	-	-
CO5	1	2	2	1	2	2	-	-
CO6	-	1	-	2	-	-	-	-
Course Correlation Mapping	3	2	2	3	3	2	-	-

Correlation Levels: 3: High; 2: Medium; 1: Low

COURSE CONTENT

MODULE 1: INTRODUCTION TO MACHINE LEARNING

(09 Periods)

Machine Learning: Human learning, Types of human learning, Machine learning, Types of machine learning, Applications of machine learning, Issues in machine learning.

Preparing to Model: Machine learning activities, Types of data, Exploring structure of data, Data quality and remediation.

MODULE 2: MODELING AND EVALUATION, BAYESIAN CONCEPT LEARNING

(09 Periods)

Modeling and Evaluation: Selecting a model, Training a model, Model representation and interpretability, Evaluating performance of a model, Improving performance of a model.

Feature Engineering: Feature transformation, Feature subset Selection.

Bayesian Concept Learning: Introduction, Importance, Bayes' theorem, Bayes' theorem and concept learning, Bayesian belief network.

MODULE 3: SUPERVISED LEARNING

(10 Periods)

Classification: Classification model, Classification learning steps, K-Nearest Neighbor, Support vector machines, Decision Tree - Decision tree representation, Problems for decision tree learning, Decision tree learning algorithm, Hypothesis space search, Inductive bias in decision tree learning, Issues in decision tree learning.

Regression: Introduction, Simple linear regression, Improving accuracy of the linear regression model, Multiple linear regression, Assumptions and problems in regression analysis, Polynomial regression model.

MODULE 4: UNSUPERVISED LEARNING

(07 Periods)

Introduction, Unsupervised vs supervised learning, Applications of unsupervised learning, Clustering as a machine learning task, Types of clustering techniques, Partitioning methods, K-Medoids, Hierarchical clustering.

MODULE 5: ARTIFICIAL NEURAL NETWORKS, ENSEMBLE LEARNING

(10 Periods)

Artificial Neural Networks: Neural network representations, Appropriate problems for neural network learning, Perceptrons, Multilayer networks and Backpropagation algorithm, Convergence and local minima, Representational power of feedforward networks, Hypothesis space search and inductive bias, Hidden layer representations, Generalization, Overfitting.

Ensemble Learning: Bagging, Boosting, Gradient boosting.

Total Periods: 45

Course Code	Course Title	L	T	P	S	C
CA25PG0208L	NATURAL LANGUAGE PROCESSING	3	-	2	-	4

Pre-Requisite -

Anti-Requisite -

Co-Requisite -

COURSE DESCRIPTION: Language Modeling, Regular Expressions, Text Normalization, Word level analysis, PoS Tagging and Entropy models; Context free grammars and Parsing techniques; Semantics and pragmatics, Discourse analysis and lexical resources.

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

- CO1.** Show sensitivity to linguistic phenomena and an ability to model them with formal grammars
- CO2.** Understand and carry out proper experimental methodology for training and evaluating empirical NLP systems
- CO3.** Able to design, implement, and analyze NLP algorithms
- CO4.** Able to design, implement, and analyze NLP algorithms
- CO5.** Able to design different language modeling Techniques.

CO-PO Mapping Table

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	-	-	-	-	-
CO2	3	3	-	-	-	3	3	3
CO3	2	3	3	3	-	2	2	-
CO4	3	-	-	-	3	3	-	-
CO5	2	3	-	-	3	3	-	-
Course Correlation Mapping	3	3	3	3	3	3	3	3

Correlation Level:

3- High

2-Medium

1-Low

COURSE CONTENT

MODULE 1: FINDING THE STRUCTURE OF WORDS

(11 Periods)

Finding the Structure of Words: Words and Their Components, Issues and Challenges, Morphological Models

Finding the Structure of Documents: Introduction, Methods, Complexity of the Approaches, Performances of the Approaches

MODULE 2: SYNTAX ANALYSIS

(09 Periods)

Syntax Analysis: Parsing Natural Language, Treebanks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure, Parsing Algorithms, Models for Ambiguity Resolution in Parsing, Multilingual Issues

MODULE 3: SEMANTIC PARSING

(08 Periods)

Semantic Parsing: Introduction, Semantic Interpretation, System Paradigms, Word Sense Systems, Software.

MODULE 4: HADOOP ARCHITECTURE

(07 Periods)

Predicate-Argument Structure, Meaning Representation Systems, Software.

MODULE 5: DATA ANALYTICS WITH R

(10 Periods)

Discourse Processing: Cohesion, Reference Resolution, Discourse Cohension and Structure
Language Modeling: Introduction, N-Gram Models, Language Model Evaluation, Parameter Estimation, Language Model Adaptation, Types of Language Models, Language-Specific Modeling Problems, Multilingual and Cross lingual Language Modeling

Total Periods: 45

EXPERIENTIAL LEARNING

1. To Study Preprocessing of text (Tokenization, Filtration, Script Validation, Stop Word Removal, Stemming).
2. To Study Morphological Analysis.
3. POS Tagging
4. Chunking
5. Named Entity Recognition
6. Virtual Lab on Word Generator
7. Mini Project based on NLP Processor

TEXT BOOKS:

1. Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication.
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary

REFERENCE BOOKS:

1. Speech and Natural Language Processing - Daniel Jurafsky& James H Martin, Pearson Publications.

VIDEO LECTURES:

1. <https://www.youtube.com/watch?v=d4gGtcobq8M>
2. <https://www.youtube.com/watch?v=fNxajSNG3-s>

WEB RESOURCES:

1. <https://www.udemy.com/course/data-science-natural-language-processing-in-python/>
2. https://www.tutorialspoint.com/natural_language_processing/index.htm
3. <https://www.javatpoint.com/nlp>

Course Code	Course Title	L	T	P	S	C
CA25PG0209L	MERN STACK DEVELOPMENT	3	-	2	-	4
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This course provides a detailed discussion and hands-on experience on MERN Stack, Building a Webserver with ExpressJS, Building a Restful API, Building Web Applications with React, MongoDB, NodeJS and Managing state with Redux.

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

- CO1.** Demonstrate knowledge on fundamentals of MongoDB and NodeJS.
- CO2.** Build web server using ExpressJS to configure and customize web applications.
- CO3.** Apply Restful API with ExpressJS and Mongoose.
- CO4.** Apply React to design enterprise web applications.
- CO5.** Develop web applications through MERN stack and Redux store as per societal needs.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	-	-	-	-	-	-
CO2	3	-	-	-	-	-	-	-
CO3	2	3	3	2	-	-	-	-
CO4	1	2	3	-	3	-	-	1
CO5	1	2	3	-	-	1	1	-
Course Correlation Mapping	2	3	3	2	3	1	1	1

Correlation Level:

3- High

2-Medium

1-Low

COURSE CONTENT

MODULE 1: MERN STACK

(08 Periods)

Introduction to MERN STACK, The Model View Controller (MVC) architectural pattern, Installing and configuring MongoDB, Installing Node.js and Installing npm packages.

MODULE 2: BUILDING A WEB SERVER WITH EXPRESSJS

(09 Periods)

Routing in ExpressJS, Modular route handlers, Writing middleware functions, Writing configurable middleware functions, Writing router-level middleware functions, Writing error- handler middleware functions, Using ExpressJS' built-in middleware function for serving static as sets, Parsing the HTTP request body, Compressing HTTP responses, Using an HTTP request logger.

MODULE 3: BUILDING A RESTFUL API

(09 Periods)

CRUD operations using ExpressJS' route methods, CRUD operations with Mongoose, Using Mongoose query builders, Defining document instance methods, Defining static model methods, Writing middleware functions for Mongoose, Writing custom validators for Mongoose's schemas, Building a RESTful API to manage users with ExpressJS and Mongoose.

MODULE 4: BUILDING WEB APPLICATIONS WITH REACT

(10 Periods)

Understanding React elements and React components, Stateful components and life cycle methods, Working with React, Pure Component, React event handlers, Conditional rendering of components, Rendering lists with React, Understanding refs and how to use them, Understanding React portals, Catching errors with error boundary components.

MODULE 5: MANAGING STATE WITH REDUX

(09 Periods)

Defining actions and action creators, Defining reducer functions, Creating a Redux store, Binding action creators to the dispatch method, Splitting and combining reducers, Writing Redux store enhancers, Time traveling with Redux, Understanding Redux middleware, Dealing with asynchronous data flow.

Total Periods: 45

EXPERIENTIAL LEARNING

Front-end Web Application Library

1. Installing Node.js framework and configuring Visual Studio (VS) Code Integrated Development Environment (IDE), and its dependencies.
2. Create and Run —Hello World Application in VS Code.
3. Build two mini applications using Router that will be served in two different sub-domains virtual domains through ExpressJS and Virtual domains.
4. Develop the code for Securing an ExpressJS web application with Helmet.
5. Create a React application that includes simple functional components.
6. Create a React application that includes simple class components.

7. Develop a React application to insert and access props (properties) and state of components.
8. Create a React application to demonstrate event handling.
9. Develop a React application for list rendering.
10. Implement a React application for form handling.

Hosting Web Applications

1. Choosing a hosting server and selecting a plan for web hosting.
2. Choosing and configuring DNS address.
3. Uploading, configuring and running the website over the internet.

RESOURCES

TEXT BOOKS:

1. Eddy Wilson IriarteKoroliova, MERN Quick Start Guide: Build Web Applications with MongoDB, Express.js, React, and Node, Packt, May 2018.
2. Greg Lim, BeginningMERN Stack Development, First Edition, June 2021.

REFERENCE BOOKS:

1. Brad Dayley, Node.js, MongoDB and Angular Web Development, Pearson, 2nd Edition, 2017.
2. Kogent Learning Solutions Inc, HTML 5 Black Book: Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP and JQuery, Dreamtech Press, Second Edition, 2016.

SOFTWARE/TOOLS:

1. Software: Node.js, MongoDB and Visual Studio Code; Express and React modules.

VIDEO LECTURES:

1. <https://www.youtube.com/watch?v=E-GA9GKJWuE>
2. <https://www.youtube.com/watch?v=Oe421EPjeBE>
3. <https://www.youtube.com/watch?v=KNZZZ3pbqco>
4. <https://www.youtube.com/watch?v=bbkBuqC1rU4>

WEB RESOURCES:

1. Node.js Tutorial - <https://www.w3schools.com/nodejs/>
2. Express.js Tutorial - <https://www.javatpoint.com/expressjs-tutorial>
3. MongoDB Tutorial - <https://www.tutorialspoint.com/mongodb/index.htm>
4. Learn ReactJS Tutorial - <https://www.javatpoint.com/reactjs-tutorial>
5. Redux Tutorial - <https://www.tutorialspoint.com/redux/index.htm>

Course Code	Course Title	L	T	P	S	C
CA25PG0210L	DEVOPS	3	-	2	-	4

Pre-Requisite --

Anti-Requisite --

Co-Requisite --

COURSE DESCRIPTION: Introduces the methodologies, tools, and insights of the DevOps process and what it can do for an organization. The course covers development, deployment and operations including infrastructure as code, continuous deployment, testing automation, validation, monitoring and security.

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

- CO1.** Understand different actions performed through Version control tools like Git.
- CO2.** Perform Continuous Integration and Continuous Testing and Continuous Deployment using Jenkins by building and automating test cases using Maven& Gradle.
- CO3.** To Perform Automated Continuous Deployment
- CO4.** To do configuration management using Ansible
- CO5.** Understand to leverage Cloud-based DevOps tools using Azure DevOps

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	2	-	-	-	-
CO2	3	3	3	2	3	3	-	-
CO3	3	3	3	2	-	-	-	-
CO4	3	3	3	2	3	3	-	3
CO5	3	3	3	2	3	3	-	3
Course Correlation Mapping	3	3	3	2	3	3	-	3

Correlation Level: **3- High** **2-Medium** **1-Low**

COURSE CONTENT

MODULE 1: INTRODUCTION TO DEVOPS**(09 Periods)**

Devops Essentials-Introduction To AWS, GCP, Azure-Version control systems: Git and Git hub.

MODULE 2: COMPILE AND BUILD USING MAVEN & GRADLE**(09 Periods)**

Introduction, Installation of Maven, POM files, Maven Build lifecycle, Build phases(compile build, test, package) Maven Profiles, Maven repositories(local, central, global),Maven plugins, Maven create and build Artifacts, Dependency management, Installation of Gradle , Understand build using Gradle.

MODULE 3: CONTINUOUS INTEGRATION USING JENKINS**(09 Periods)**

Install & Configure Jenkins, Jenkins Architecture Overview, Creating a Jenkins Job, Configuring Jenkins job, Introduction to Plugins, Adding Plugins to Jenkins, Commonly used Plugins (GitPlugin, Parameter Plugin, HTML Publisher, Copy Artifact and Extended choice parameters). Configuring Jenkins to work with java, Git and Maven, Creating a Jenkins Build and Jenkins workspace.

MODULE 4: CONFIGURATION MANAGEMENT USING ANSIBLE**(09 Periods)**

Ansible Introduction, Installation, Ansible master /slave configuration,YAML basics, Ansible modules, Ansible Inventory files, Ansible play books, Ansible Roles, adhoc commands in ansible.

MODULE 5: BUILDING DEVOPS PIPELINES USING AZURE**(09 Periods)**

Create Git hub Account, Create Repository, Create Azure Organization, Create a new pipeline, Build a sample code, Modify azure- pipeline. yaml file

Total Periods: 45**EXPERIENTIAL LEARNING**

1. Create Maven Build pipeline in Azure
2. Run regression tests using Maven Build pipeline in Azure
3. Install Jenkins in Cloud
4. Create CI pipeline using Jenkins
5. Create a CD pipeline in Jenkins and deploy in Cloud
6. Create an Ansible play book for a simple web application infrastructure
7. Build a simple application using Gradle

RESOURCES**TEXT BOOKS:**

1. Roberto Vormittag, "A Practical Guide to Git and GitHub for Windows Users: From Beginner to Expert in Easy Step-By-Step Exercises", Second Edition, Kindle Edition, 2016.
2. Jason Cannon, "Linux for Beginners: An Introduction to the Linux Operating System and Command Line", Kindle Edition, 2014

REFERENCE BOOKS:

1. Hands-On Azure DevOps: Cid Implementation For Mobile, Hybrid, And Web Applications Using Azure DevOps And Microsoft Azure: CICD Implementation for DevOps and Microsoft Azure(English Edition) Paperback– 1January2020byMiteshSoni
2. JeffGeerling,“Ansible for DevOps: Server and configuration management for humans”, Fi rstEdition,2015.
3. DavidJohnson, “Ansible for DevOps: Everything You Need to Know to Use Ansible for DevO ps”, SecondEdition,2016.

SOFTWARE/TOOLS:

GitHub, VMware Workstation, VMware Fusion, or Oracle’s VirtualBox software should meet the following specifications:

1. **CPU Capabilities** – A dual core 64 bit CPU, with four threads (Intel hyper-threading) is required. An Intel i5 or i7 type chip or the AMD equivalent is recommended.
2. **Computer Memory** – 8 GB of RAM required, 16 GB or more recommended.
3. **Disk** – 500 GB of disk space for your coursework is required. A Solid-state disk is recommended.

Course Code	Course Title	L	T	P	S	C
CA25PG0501L	GENERATIVE AI TOOLS AND PROMPT ENGINEERING	-	1	2	-	2

Pre-Requisite -

Anti-Requisite -

Co-Requisite -

COURSE DESCRIPTION: This course introduces students to the practical use of Generative AI tools across a broad spectrum of real-world applications. Beginning with the fundamentals of prompt engineering and progressively apply AI to creative, technical, and productivity tasks. The course covers AI-generated images, videos, and multimedia storytelling, followed by hands-on experience in building websites and web applications using prompt-based development platforms. The course concludes with AI-assisted productivity workflows including automated presentation creation and intelligent data analysis.

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

- CO1.** Design and evaluate instruction-based, zero-shot, and few-shot prompts to generate effective responses from Generative AI models.
- CO2.** Apply AI coding tools to generate, test, refine, and document code through structured prompt-based development workflows.
- CO3.** Generate and transform images, videos, and multimedia content using AI tools through effective prompt design.
- CO4.** Build and deploy websites and web applications using prompt-based AI development platforms.
- CO5.** Apply Generative AI tools to automate presentation creation and perform intelligent data analysis.
- CO1.** Demonstrate effective individual, teamwork skills and communicate experimental results effectively.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	3	3	2	-	-	3	2
CO2	3	3	3	2	-	-	3	2
CO3	3	3	3	2	-	-	3	2
CO4	3	3	3	2	-	-	3	2
CO5	3	3	3	2	-	-	3	2
CO6								2
Course Correlation	3	3	3	2	-	-	3	2

Correlation Levels: 3: High; 2: Medium; 1: Low

EXPERIENTIAL LEARNING

Introduction to Generative AI and Prompt Engineering:

Introduction to Artificial Intelligence and Generative AI – Large Language Models (LLMs) – Fundamentals of Prompt Engineering – Types of Prompts – Instruction-based and Question-based Prompting – Zero-shot and Few-shot Prompt Design – Prompt Evaluation and Response Analysis.

6. Implement the following prompt engineering fundamentals activities:
 - a) Create instruction-based prompts for information retrieval.
 - b) Create zero-shot prompts for general question answering.
 - c) Create few-shot prompts, role-based and contextual prompts using example inputs and outputs.

AI-Assisted Code Generation

Introduction to AI-Assisted Code Generation – Role of AI in Software Development – Prompt Strategies for Code Generation – Iterative Prompt Refinement – Code Testing and Debugging Using AI – Code Comprehension and Explanation Using AI – Bug Identification and Code Improvement Using AI Tools.

7. Develop a simple application entirely through AI-assisted code generation using any AI coding tool.
 - a) Choose a simple application idea (e.g. calculator, to-do list, quiz app, expense tracker) and a programming language (e.g. Python, JavaScript, HTML/CSS).
 - b) Generate the complete application code using an AI tool through a series of structured prompts.
 - c) Test the generated application, identify issues, and refine the code using follow-up prompts until the application works correctly.
8. Use AI tools to understand, explain, and document existing code.
 - a) Paste an unfamiliar or complex code snippet into an AI tool and prompt it to explain the code line by line.
 - b) Generate inline comments and a documentation summary for the code using AI prompts.
 - c) Ask the AI to identify bugs or improvements in the code and apply the suggested refinements.

AI Image Generation and Digital Content Creation

Principles of AI Image Generation – Text-to-Image Models – Image Editing – Ghibli Art, Cartoonization, Poster Design, Branding, and Creative Digital Content Generation.

9. Implement AI image generation activities using text prompts:
 - a) Generate realistic images from textual descriptions.
 - b) Create poster and advertisement designs using AI tools.
 - c) Compare image quality using different prompt styles.
10. Implement AI image transformation activities:
 - a) Generate Ghibli-style artwork from photographs.
 - b) Create cartoon and anime-style image variations.
 - c) Apply artistic filters and style-transfer techniques.
 - d) Design custom avatars and digital characters.

AI Video Generation and Digital Content Creation

Introduction to AI Video Generation – Text-to-Video and Image-to-Video Techniques – Animated Content Creation – Social Media Content Development.

11. Implement AI video generation activities:
 - a) Generate short videos from text prompts.
 - b) Create educational videos using AI tools.
 - c) Convert images into animated videos.

- d) Create social media reels and shorts using AI.

AI-Assisted Website and Application Development

Introduction to AI-Powered Development Platforms – Prompt-Driven Website Design – AI-Assisted Application Development – User Interface Generation – Website Deployment and Rapid Prototyping using Modern AI Development Tools.

12. Implement AI-assisted website development activities:
 - a) Generate a personal portfolio website using Lovable AI or similar platforms.
 - b) Customize layouts, themes, and content using prompts.
 - c) Deploy the generated website.
13. Implement AI application development activities:
 - a) Generate a simple web application using Bolt.new, Lovable, or Firebase Studio.
 - b) Modify features through prompt-based development.
 - c) Test and refine generated applications.

PPT Generation Using AI Tools

AI-Assisted Presentation Generation – Prompt-based Slide Generation – Customizing Layouts, Themes, and Content Using Prompts.

14.
 - a) Generate a presentation on a given topic using AI tools such as Gamma, Beautiful.ai, or Microsoft Copilot.
 - b) Customize slide layouts, themes, and content using prompts.
 - c) Compare AI-generated presentations with manually created ones.

Data Analysis Using AI Tools

Introduction to AI-Assisted Data Analysis – Generating Insights, Summaries, and Visualizations Through Prompts – Interpreting and Documenting AI-Generated Analysis Results.

15.
 - a) Upload a dataset (CSV/Excel) to an AI tool such as ChatGPT, Claude, or Julius AI.
 - b) Generate insights, summaries, and visualizations using natural language prompts.
 - c) Interpret and document the AI-generated analysis results.

SOFTWARE TOOLS

- ✓ AI Tools – LLMs, Image Generation Tools, Video Generation Tools, Presentation Tools, Data Analysis Tools.
- ✓ Web Browser
- ✓ Coding IDE
- ✓ Spreadsheet Tool

RESOURCES

REFERENCE BOOKS:

1. James Phoenix and Mike Taylor, "Prompt Engineering for Generative AI: Future-Proof Inputs for Reliable AI Outputs", O'Reilly Media, 2024.
2. Karen E. Sutherland, "Artificial Intelligence for Strategic Communication", Springer Nature, 2025.
2. Nathan Hunter, "The Art of Prompt Engineering with ChatGPT: A Hands-On Guide", Independently Published, 2023.
3. Oliver Theobald, "Generative AI Art", Packt Publishing, 2026.
4. Lee Boonstra, "Prompt Engineering Guide for Generative AI", Google Cloud Technical Documentation.

VIDEO LECTURES:

1. <https://www.youtube.com/watch?v=8ib4Qnh2HFE>
2. <https://www.youtube.com/watch?v=mEsleV16qdo>
3. <https://www.youtube.com/watch?v=jC4v5AS4RIM>
4. <https://www.youtube.com/watch?v=qrfj32pwGFw>
5. <https://www.youtube.com/watch?v=DOXJ7s1D6iE>

WEB RESOURCES:

1. <https://cloud.google.com/blog/products/ai-machine-learning/ultimate-prompting-guide-for-nano-banana>
2. <https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/overview>
3. <https://www.ibm.com/think/topics/prompt-engineering>
4. <https://mitsloanedtech.mit.edu/ai/basics/effective-prompts/>
5. <https://learn.microsoft.com/en-us/azure/foundry/openai/concepts/prompt-engineering>
6. <https://docs.aws.amazon.com/bedrock/latest/userguide/prompt-engineering-guidelines.html>

Course Code	Course Title	L	T	P	S	C
CA25PG0110L	FULL STACK WEB DEVELOPMENT	3	-	-	-	3
Pre-Requisite	-					
Anti-Requisite	-					
Co-Requisite	-					

COURSE DESCRIPTION: This course provides a detailed discussion and hands-on experience on Hypertext markup language (HTML), Features of HTML5, Cascading style sheets (CSS), JavaScript, JQuery, Bootstrap, AJAX, Hypertext preprocessor (PHP), MySQL.

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

- CO1.** Demonstrate knowledge on web page design elements, dynamic content and database interaction.
- CO2.** Analyze user requirements to develop web applications.
- CO3.** Design client-server applications using web technologies.
- CO4.** Demonstrate problem solving skills to develop enterprise web applications.
- CO5.** Use HTML, CSS, JavaScript, JQuery, Bootstrap and PHP technologies for device independent web application development.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	-	-	-	-	-	-	-
CO2	2	3	-	-	-	-	-	-
CO3	2	2	3	-	-	-	-	-
CO4	2	2	2	3	-	-	-	-
CO5	2	2	3	2	3	-	-	-
Course Correlation Mapping	2	2	3	2	3	-	3	-

Correlation Level: **3- High** **2-Medium** **1-Low**

COURSE CONTENT

MODULE 1: HTML AND CSS

(09 Periods)

HTML5: Introduction, HTML5 document structure, Creating editable content, Checking spelling mistakes, Exploring custom data attributes, Client-Side storage, Drag and drop feature, Offline web applications, Web communications.

CSS3: Introduction, CSS selectors, Inserting CSS in an HTML document, Backgrounds, Fonts, and Text styles, Creating boxes, Displaying, Positioning and floating elements, Features of CSS3, Media queries.

MODULE 2: JAVASCRIPT AND JQUERY

(10 Periods)

JavaScript: Overview of JavaScript, JavaScript functions, Events, Image maps and animations, JavaScript objects, Working with browser and document objects.

JQuery: Introduction, JQuery Selectors, Events, Methods to access HTML elements and attributes, Introduction to AJAX.

MODULE 3: BOOTSTRAP

(09 Periods)

Getting started with Bootstrap, Creating responsive layouts using Bootstrap CSS - Basic HTML structure for Bootstrap, Responsive classes, Rendering images, The grid system, Constructing data entry forms, Packaged components in Bootstrap - The page header, Glyphicons, The navigation bar, Badges, Alerts, Toolbars and button groups.

MODULE 4: INTRODUCTION TO PHP

(09 Periods)

Introduction, Data types, Variables, Constants, Expressions, String interpolation, Control structures, Functions, Arrays, Embedding PHP code in web pages, Object Oriented PHP.

MODULE 5: PHP WEB FORMS AND MYSQL

(08 Periods)

PHP Web forms: PHP and web forms, Sending form data to a server, Working with cookies and session handlers.

PHP with MySQL: Interacting with the database, Prepared statement, Database transactions.

Total Periods: 45

EXPERIENTIAL LEARNING

1. Explore Basic HTML Tags and Elements.
2. Familiarize with JS, and CSS, Animation using sample webpage
3. Design a Webpage using advance HTML Form tags input-date, time, number, email, HTML5 Header And Footer, spell check and editable areas.
4. Write a JavaScript/JQuery code to validate the following fields of the Registration web page.
 - a. First Name/Last Name - should contain only alphabets and the length should not be less than 8 characters.

- b. Username - It should contain combination of alphabets, numbers and underscore. It should not allow spaces and special symbols.
 - c. Password - It should not less than 8 characters in length and it contains one uppercase letter and one special symbol.
 - d. Date of Birth - It should allow only valid date; otherwise display a message stating that entered date is invalid. Ex. 29 Feb. 2009 is an invalid date.
 - e. Postal Code: It must allow only 6 digit valid number.
 - f. Mobile No. - It should allow only numbers and total number of digits should be equal to 10.
 - g. e-mail id - It should allow the mail id with the following format:
Ex. mailid@domainname.com
5. Design a web page with the following features using HTML5, JavaScript and JQuery
 - a. Displaying of images with Custom animated effects
 - b. Playing of selected video from the list of videos
 - c. Showing the animated text in increasing and decreasing font size
 - d. Changing the size of the area in a web page using DIV tag
 - e. Hiding and Showing elements in a web page.
 6. Design a web page with the following features using Bootstrap and Media Query.
 - a. Components
 - b. Responsive tables
 - c. Responsive images and videos
 - d. Toolbars, Buttons and Lists
 7.
 - a. Deploy and navigate web pages of online book store using WAMP/XAMPP web server.
 - b. Write a PHP program to read user name and favorite color from the HTML form. Display the name of the user in green color and sets user favorite color as a background for the web page.
 8. Change a Content of webpage using AJAX. Perform Different Operations using JQUERY Selectors.
 9. Write a PHP code to read user details entered through the registration web page and store the same into MySQL database.
 10. Write a PHP code for storing books details like Name of the book, author, publisher, edition, price, etc into MySQL database. Embed a PHP code in catalogue page of the online book store to extract books details from the database.

RESOURCES

TEXT BOOKS:

1. Kogent Learning Solutions Inc, "*HTML 5 Black Book: Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP and JQuery*," Dreamtech Press, 2nd Edition, 2016.
2. W. Jason Gilmore, "*Beginning PHP and MySQL*, APress," 4th Edition, 2011.

REFERENCE BOOKS:

1. Snig Bahumik, "*Bootstrap Essentials*," PACKT Publishing, 1st Edition, 2015. (e-book).
2. Thomas A. Powell, "*The Complete Reference: HTML and CSS*," Tata McGraw Hill, 5th Edition, 2010.
3. Andrea Tarr, "*PHP and MySQL*," Willy India, 1st Edition, 2012.

SOFTWARE/TOOLS:

1. Editor: Notepad++

2. Programming Language: Javascript, JQuery, PHP

VIDEO LECTURES:

1. <https://www.youtube.com/watch?v=Q33KBiDriJY>
2. <https://www.coursera.org/learn/html-css-javascript-for-web-developers?=>
3. <https://www.edx.org/course/using-javascript-jquery-and-json-in-django>
4. <https://www.edx.org/course/building-web-applications-in-django>
5. <https://www.udemy.com/course/web-technology-for-entrepreneurs/>
6. <https://www.edx.org/course/html5-apps-and-games>
7. <https://www.classcentral.com/course/openhpi-web-technologies-3517>
8. <https://www.edx.org/course/css-basics>

WEB RESOURCES:

1. <https://www.w3schools.com/php/DEFAULT.asp>
2. <https://www.w3schools.com/js/>
3. <https://www.geeksforgeeks.org/web-technology/>
4. <https://www.smashingmagazine.com/2021/03/complete-guide-accessible-front-end-components/>
5. <https://css-tricks.com/>
6. <https://davidwalsh.name/css-optional>
7. <https://jquery.com/>

Course Code**Course Title****L T P S C****CA25PG0111L****ENTREPRENEURSHIP AND INNOVATION
MANAGEMENT**

3 - - - 3

Pre-Requisite -**Anti-Requisite** -**Co-Requisite** -

COURSE DESCRIPTION: This course aims to provide students with a deep understanding of entrepreneurship and innovation. It explores entrepreneurial processes, opportunity identification, business planning, innovation management, intellectual property rights, and venture growth strategies. Students will develop entrepreneurial thinking, creativity, and problem-solving abilities to create and manage innovative ventures that contribute to economic and societal development.

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

- CO1.** Understand the fundamentals of entrepreneurship and its role in economic development.
- CO2.** Analyze opportunities and prepare business plans for entrepreneurial ventures.
- CO3.** Apply creativity and innovation techniques to business problems.
- CO4.** Demonstrate knowledge of technology management, IPR, and startup ecosystem.
- CO5.** Evaluate financing options, marketing strategies, and growth models for ventures.

CO-PO Mapping Table:

Course Outcomes	Program Outcomes							
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1	3	2	1	1	-	-	-	-
CO2	3	3	2	2	2	-	-	-
CO3	2	3	3	2	2	1	-	-
CO4	2	2	2	2	2	1	-	-
CO5	3	3	2	2	3	1	-	-
Course Correlation Mapping	3	3	2	2	2	1	-	-

Correlation Level: **3- High** **2-Medium** **1-Low**

COURSE CONTENT

MODULE 1: INTRODUCTION TO ENTREPRENEURSHIP (09 Periods)

Concept, meaning and importance of entrepreneurship – Characteristics and skills of entrepreneurs – Types of entrepreneurs: social, women, corporate, rural, technology-based – Intrapreneurship vs. entrepreneurship – Entrepreneurial ecosystem and its components.

MODULE 2: ENTREPRENEURIAL PROCESS AND BUSINESS PLANNING (09 Periods)

Stages of entrepreneurial process: Idea, Feasibility, Startup, Growth – Opportunity identification and evaluation – Market research and environmental scanning – Structure and components of a business plan – Case studies of successful entrepreneurs.

MODULE 3: INNOVATION MANAGEMENT (09 Periods)

Meaning, scope and significance of innovation – Types of innovation: product, process, business model, disruptive, frugal – Creativity techniques: brainstorming, lateral thinking, design thinking, TRIZ – Managing innovation in organizations – Innovation as a competitive advantage.

MODULE 4: TECHNOLOGY, IPR AND STARTUP ECOSYSTEM (09 Periods)

Technology management and commercialization – Intellectual Property Rights (IPR): patents, copyrights, trademarks, designs, trade secrets – Technology transfer and licensing – Startup India, Atal Innovation Mission, MSME policies – Role of incubators, accelerators and innovation hubs.

MODULE 5: FINANCING AND GROWTH OF VENTURES (09 Periods)

Sources of finance: bootstrapping, angel investors, venture capital, crowdfunding, government support – Entrepreneurial marketing strategies – Financial planning for startups – Scaling up ventures: challenges and strategies – Exit strategies: mergers, acquisitions, IPO.

Total Periods: 45

EXPERIENTIAL LEARNING

1. Prepare a mini-business plan for a startup idea.
2. Visit a nearby incubation/innovation center and prepare a report.
3. Conduct a creativity workshop (brainstorming/design thinking session).
4. Prepare a case study presentation on a successful entrepreneur/startup.

RESOURCES

TEXT BOOKS:

1. Hisrich, R.D., Peters, M.P., & Shepherd, D.A. (2017), *Entrepreneurship*, McGraw-Hill Education.
2. Drucker, P. (2007), *Innovation and Entrepreneurship*, Harper Business.
3. Kuratko, D.F. (2020), *Entrepreneurship: Theory, Process and Practice*, Cengage Learning.

REFERENCE BOOKS:

1. Timmons, J.A., & Spinelli, S. (2019), *New Venture Creation: Entrepreneurship for the 21st Century*, McGraw-Hill Education.
2. Schilling, M.A. (2020), *Strategic Management of Technological Innovation*, McGraw-Hill Education.
3. Scarborough, N.M. (2018), *Essentials of Entrepreneurship and Small Business Management*, Pearson.

VIDEO LECTURES:

1. <https://www.youtube.com/watch?v=rA4uKIy5gO0&list=PLsh2FvSr3n7fQlIDbfKutmSL26TsWitGQ>
2. <https://www.youtube.com/watch?v=itRVzjk9mkg>

WEB RESOURCES:

1. https://nacosadsu.org.ng/main/docs/300L/ENT%20301.pdf?utm_source=chatgpt.com
2. https://www.measiim.edu.in/myweb/uploads/2022/05/PMFEA-IE-1.pdf?utm_source=chatgpt.com
3. https://ocw.mit.edu/courses/15-351-managing-innovation-and-entrepreneurship-spring-2008/pages/lecture-notes/?utm_source=chatgpt.com