FACULTY OF COMPUTER APPLICATION

Proposed Course Structure aligned with NEP-2020

BACHELOR OF COMPUTER APPLICATION
&
BACHELOR OF COMPUTER APPLICATION
with specialisation in

Artificial Intelligence
in Association with

IBM

[Academic Session 2025-26 onwards]

PRAYAGRAJ

[Established under the U. P. Private Universities Act. No. 12 of 2019]

FACULTY OF COMPUTER APPLICATION

Bachelor of Computer Application (BCA)
&
Bachelor of Computer Application (BCA)-Artificial Intelligence
In Association with IBM

Program Outcomes

PO1: Computational Knowledge: Understand and apply foundational mathematical concepts, computing principles, and domain-specific knowledge to conceptualize computing models for solving defined problems.

PO2: Problem Analysis: Ability to identify, analyze, and formulate complex computing problems using fundamentals of computer science and application domains relevant to the BCA field.

PO3: Design / Development of Solutions: Transform complex business scenarios and contemporary issues into well-defined problems, and propose integrated solutions using emerging technologies and tools appropriate for the BCA domain.

PO4: Conduct Investigations of Computing Problems: Devise and conduct experiments, interpret data, and draw conclusions to address complex computing problems in the context of BCA applications.

PO5: Modern Tool Usage: Select and utilize modern computing tools, techniques, and skills necessary for developing innovative software solutions in the BCA field.

PO6: Professional Ethics: Apply and commit to professional ethics and cyber regulations relevant to the BCA profession, understanding the importance of integrity, privacy, and security.

PO7: Life-long Learning: Recognize the significance of continuous learning and development, staying updated with advancements in the BCA domain throughout their professional careers.

PO8: Project Management: Understand and apply project management principles with computing knowledge to manage projects related to BCA applications in multidisciplinary environments.

PO9: Communication Efficacy: Communicate effectively with the computing community and society, demonstrating the ability to comprehend and present technical concepts in clear and accessible ways.

PO10: Societal & Environmental Concern: Recognize the impact of computer technology on society, including economic, environmental, and social aspects, and adhere to ethical practices and responsibilities in BCA professional practice.

PO11: Individual & Team Work: Work proficiently as a member or leader in diverse teams, promoting collaborative problem-solving in the context of BCA projects and applications.

PO12: Innovation and Entrepreneurship: Identify opportunities, foster an entrepreneurial vision, and use innovative ideas to create value and contribute to the betterment of individuals and society, leveraging BCA skills and knowledge.

Program Specific Outcomes

PSO1. Explore technical comprehension in varied areas of Computer Applications and experience a conducive environment in cultivating skills for thriving career and higher studies.

PSO2. Comprehend, explore and build up computer programs in the allied areas like Algorithms, System Software, Multimedia, Web Design and Data Analytics for efficient design of computer-based systems of varying complexity.

PSO3: Demonstrate a clear understanding of both conceptual and application-oriented skills in computer applications within a business context, effectively applying these skills to design, develop, and implement practical solutions that meet organizational needs.

FACULTY OF COMPUTER APPLICATION

COURSE STRUCTURE

Bachelor of Computer Application (BCA)

[Academic Session 2025-26 onwards]

Entry Level-1

Sr. No.	Course Code	Course Title	Category	Teaching			Credit
				L	T	P	
Semester- I							
1	CAUCBC106T	Problem Solving Techniques	PC	3	-	-	3
2	CAUCBC114T	Python Programming	PC	3	-	-	3
3	CAUCBC107T	Digital Logic Design	PC	3	-	-	3
4	CMSFPCA11T	Business Communications	HS	3	-	-	3
5	SCSEPCA11T	Environmental Science	HS	2	-	-	2
6	CAUCBC106P	Problem Solving Techniques Lab	PC	-	-	2	1
7	CAUCBC114P	Python Programming Lab	PC	-	-	4	2
8	CAUCBC104P	Digital Logic Design Lab	PC	-	-	2	1
9	PTSPPCA12T	Professional Proficiency	HS	1	-	2	2
10	UUSCVA001P	Yoga/NSS/Sports/Music		-	-	2	0
Total				**15**	**-**	**12**	**20**

Bachelor of Computer Application (BCA)-Artificial Intelligence

In Association with IBM

[Academic Session 2025-26 onwards]

Entry Level-1

Sr. No.	Course Code	Course Title	Category	Teaching			Credit
				L	T	P	
Semester- I							
1	CAUCBC106T	Problem Solving Techniques	PC	3	-	-	3
2	CAUIBC106T	Programming in Python and Clean Code Principles	PC	3	-	-	3
3	CAUCBC107T	Digital Logic Design	PC	3	-	-	3
4	CMSFPCA11T	Business Communications	HS	3	-	-	3
5	SCSEPCA11T	Environmental Science	HS	2	-	-	2
6	CAUCBC106P	Problem Solving Techniques Lab	PC	-	-	2	1
7	CAUIBC106P	Programming in Python and Clean Code Principles Lab	PC	-	-	4	2
8	CAUCBC104P	Digital Logic Design Lab	PC	-	-	2	1
9	PTSPPCA12T	Professional Proficiency	HS	1	-	2	2
10	UUSCVA001P	Yoga/NSS/Sports/Music		-	-	2	0
Total				**15**	**-**	**12**	**20**

COURSE CODE & NAME: CAUCBC106T / Problem Solving Techniques

COURSE OUTCOMES

1. Understanding the evolution of programming languages and differentiate between machine level, assembly, and higher-level languages
2. Apply program design techniques using hierarchy charts and express program logic through flowcharts and pseudocode.
3. Analyze algorithms for various problem-solving scenarios, including input-output statements, decision-making, and looping statements.
4. Evaluate the effectiveness of debugging techniques by identifying and categorizing different types of errors (syntax, semantic, and runtime).
5. Create algorithms and implement solutions for complex problems involving arithmetic operations, arrays, and series patterns

UNIT I:

Computer Problem Solving: Evolution of programming, languages – Introduction to machine level language, Assembly language and Higher level languages. Programing Life Cycle , Understanding the Problem Statement, Planning Program design using Hierarchy charts, Expressing Program logic using flowcharts /Pseudocode, Coding using a programing language such as 'C'/'Python', Documenting, Compiling, Debugging and Executing.

UNIT II:

Algorithm Development: Definition, Algorithm: a solution to a problem, Input-Output Statements, Decision Making Statements, Looping Statements, Examples. Flowcharting: Definition, Input-Output Statements, Decision Making Statements, Looping Statements, Module representation, Drawing conventions and standards, Example. Debugging: Bug, errors : syntax ,semantics and runtime, Compilation, Interoperation, Program debugging.

UNIT III:

Logic Building and Problem Solving Based on Number: Addition/ Subtraction/ Multiplication/ Division of two numbers, Power of a Number, Prime Number, Reversing a Number, HCF of two numbers, LCM of two numbers. Perfect No, Factor of a number, Strong number, Perfect number, Auto-morphic number, Harshad number, Abundant number, Friendly pair.

UNIT IV:

Logic Building and Problem Solving Based on Arrays: Largest element in an array, Smallest Element in an Array, Smallest and largest element in an array, Second Smallest Element in an Array, sum of elements in an array, Reverse an Array.

UNIT V:

Logic Building and Problem Solving Based on Series and patterns: Fibonacci Series, Triangular number series, Square number series, Cube number series, Alphabet Triangle, Number Triangle, Fibonacci Triangle

TEXTBOOKS/ REFERENCE BOOKS

1. "Introduction to Programming with C" by David I. Schneider
2. "Python Programming: An Introduction to Computer Science" by John Zelle
3. "Algorithms Unlocked" by Thomas H. Cormen

COURSE CODE & NAME: CAUCBC114T / Python Programming

COURSE OUTCOMES

1. Understanding basic programming skills using Python programming language.
2. Understanding the notion of data types and complex data types such as lists, tuples etc.
3. Understanding the concept of decision making and iterative control structure in python.
4. Understanding the concepts of functions and file handling in Python.

UNIT I:

Introduction to Python Language: Introduction to Python: Python variables, Python basic Operators, Understanding python blocks. Python Data Types, Declaring and using Numeric data types: int, float etc.

UNIT II:

Control Structures: Python Program Flow Control Conditional blocks: if, else and else if, Simple for loops in python, For loop using ranges, string, list and dictionaries. Use of while loops in python, Loop manipulation using pass, continue, break and else. Programming using Python conditional and loop blocks.

UNIT III:

Strings, Lists, Tuples and Dictionaries: Python Complex data types: Using string data type and string operations, Defining list and list slicing, Use of Tuple data type. String, List and Dictionary, Manipulations Building blocks of python programs, string

manipulation methods, List manipulation. Dictionary manipulation, Programming using string, list and dictionary in-built functions. Python Functions, Organizing python codes using functions.

UNIT IV:

Functions & Modules: Defining a function, Calling a function, Types of functions, Function Arguments, Anonymous functions, Global and local variables. Importing module, Math module, Packages and their composition

UNIT V:

File Handling: Python File Operations: Reading files, Writing files in python, Understanding read functions, read(), readline(), readlines(). Understanding write functions, write() and writelines() Manipulating file pointer using seek Programming, using file operations.

TEXTBOOKS/ REFERENCE BOOKS

1. R Nageswar Rao, Core Python Programming, 2018.
2. Eric Mathews, Python Crash Course, 2019.
3. Practical Programming: An introduction to Computer Science Using Python, second edition, Paul Gries, Jennifer Campbell, Jason Montojo, The Pragmatic Bookshelf.
4. Exploring Python, Timothy A. Budd, Mc Graw Hill Education

COURSE CODE & NAME: CAUCBC107T / Digital Logic Design

COURSE OUTCOMES

1. Understand and apply number systems and logic gates in digital circuits.
2. Analyze and evaluate Boolean functions using minimization techniques.
3. Design and construct combinational logic circuits with decoders and multiplexers.
4. Synthesize and compare sequential logic circuits using various flip-flops.
5. Understand and differentiate between digital integrated circuit logic families.

UNIT I:

Number system and Logic Gates: Introduction of number systems, Radix, Radix Interco versions. Radix Complement, Diminished radix complement

UNIT II:

Boolean algebra: Basic theorem of Boolean algebra. Boolean function and minimization, Karnaugh map Universal Gates, Realization of Primary gates using Universal gates only. Minterm and Maxterm Realization of Boolean Functions, Gate-level minimization: The map method up to four variables, don't care conditions, SOP and POS simplification, Quine Mc- Cluskey Method.

UNIT III:

Combinational logic circuits: Binary adder and Subtractor circuits, Magnitude comparator, Decoders, Encoders, Multiplexer, and demultiplexer, Realization of switching expressions by decoders, encoders, multiplexer and Demultiplexer.

UNIT IV:

Sequential Logic Circuits: Sequential circuits, latches, and Flip Flops, difference between latch and flip flop, SR flip flop, JK flip flop, Master Slave flip flop, comparison.

UNIT V:

Digital Integrated Circuits: Characteristics of digital ICs, Introduction to logic families-RTL, DTL, TTL, ECL. MOS and CMOS circuits and comparison

TEXTBOOKS/ REFERENCE BOOKS

1. Digital Design: M. Morris Mario (PHI)
2. Digital circuits & logic design: S. C. Lee (PHI)
3. Digital electronics: W. H. Gothmann (PHI)
4. Switching theory: A. K. Gautam (Katsons)
5. R.P. Jain, "Modern Digital Electronics," Tata McGraw Hill, 4th edition, 2009.
6. A. Anand Kumar, "Fundamental of Digital Circuits," PHI 4th edition, 2018.
7. W. H. Gothmann, "Digital Electronics- An Introduction to Theory and practice," PHI, 2Nd edition, 2006.
8. D.V. Hall, "Digital Circuits and Systems," Tata McGraw Hill, 1989.

9. A. K. Singh, “Foundation of Digital Electronics & Logic Design,” New Age Int. Publishers.
10. Subrata Ghosal, “Digital Electronics,” Cengage publication, 2nd edition, 2018

COURSE CODE & NAME: CMSFPCA11T / Business Communications

COURSE OUTCOMES

1. Understanding the evolution of programming languages and differentiate between machine level, assembly, and higher-level languages
2. Apply program design techniques using hierarchy charts and express program logic through flowcharts and pseudocode.
3. Analyze algorithms for various problem-solving scenarios, including input-output statements, decision-making, and looping statements.
4. Evaluate the effectiveness of debugging techniques by identifying and categorizing different types of errors (syntax, semantic, and runtime).
5. Create algorithms and implement solutions for complex problems involving arithmetic operations, arrays, and series patterns

UNIT I:

Business Communication: Introduction, Role of Communication In Business, Definitions of Communication, Purpose Of Communication, Communication Situation, Communication Process, Forms Of Communication (Formal & Grapevine), Barriers Of Communication, Seven Cs Of Communication.

UNIT II:

Oral Communication: Oral Communication, Advantages of Oral Communication, Limitations of Oral Communication, Two Sides of Oral Communication, Principles Of Effective Communication, Effective Listening, Non-verbal Communication

UNIT III:

Written Communication: Written Communication, Purpose of Writing, Principles of Effective Writing, Writing Techniques, Electronic Writing Process

UNIT IV:

Business Letters and Report Writing: Business Letters and Its Need, Types of Letter, Structure of Business Letter, Form of Letters, Report Writing, Types Of Business Reports, Characteristics and Purpose of a Good Report, Guiding Principles of Writing A Report, Preparing A Report, Structure of A Report

UNIT V:

Presentation Skill: Presentation, Elements Of Presentation, Designing A Presentation, Using Visual Aids, Appearance And Posture, Tips For An Effective Presentation Communication and Technology: The Role Of Technological Advancement, Communication Network, Intranet, Internet, E-mails, Teleconferencing, Videoconferencing.

TEXTBOOKS/ REFERENCE BOOKS

1. Vikram Bisen and Priya, “Business Communication”, New Age International Publishers.
2. B. M. Shaikh, “Business Communication”, Vision Publication.

3. Urmila Rai and S. M. Rai, “Business Communications”, Himalaya Publication House.
4. Asha Kaul, “Effective Business Communications”, PHI Learning private Ltd.
5. Dr. Anjali P. Kalkar,” Business Communications”, Success Publications.

COURSE CODE & NAME: SCSEPCA11T/ Environmental Science

COURSE OUTCOMES

1. Comprehend the importance of ecosystem and sustainable
2. Demonstrate interdisciplinary nature of environmental issues
3. Identify different types of environmental pollution and control measures.
4. Adopt cleaner productive technologies
5. Identify the role of non-conventional energy resources in environmental protection.
6. Analyse the impact of human activities on the environment

UNIT I:

Introduction to Environmental Studies: Multidisciplinary nature of environmental studies; Scope and importance; Concept of sustainability and sustainable development. Ecosystems: Structure and function of ecosystem; Energy flow in an ecosystem: food chains, food webs and ecological pyramids. Nutrient cycle (carbon cycle, nitrogen cycle, Sulphur cycle, water cycle, oxygen cycle).

UNIT II:

Renewable and non-renewable energy resources, Land resources and land use change; Land degradation, soil erosion and desertification. Deforestation: Causes and impact due to mining dam building on environment. Flood and drought.

UNIT III:

Environmental Pollution: air pollution, water pollution, thermal pollution, noise pollution, soil pollution; Solid Waste Management; Environmental Impact Assessment.

UNIT IV:

Biodiversity and Conservation: Levels of biological diversity: genetic, species and ecosystem diversity; hot spots; threats to biodiversity; Conservation of biodiversity: in-situ and ex -situ conservation of biodiversity.

UNIT V:

Impact of energy usage on environment: Global warming, Climate change, Depletion of ozone layer, Acid rain. Environmental ethics, Role of NGOs, Environmental Laws: Environment Protection Act. Air (Prevention and Control of Pollution) Act. Water (Prevention and control of Pollution) Act. Wildlife Protection. Act. Forest Conservation Act.

TEXTBOOKS/ REFERENCE BOOKS

1. Environmental and Pollution Awareness by Sharma BR; Satya Prakashan, New Delhi.
2. Environmental Protection Law and Policy in India by Thakur Kailash; Deep and Deep Publications, New Delhi.
3. Environmental Pollution by Dr. RK Khitoliya; S Chand Publishing, New Delhi
4. Environmental Science by Deswal and Deswal; Dhanpat Rai and Co. (P) Ltd. Delhi.

COURSE CODE & NAME: PTSPPCA12T/ Professional Proficiency

COURSE OUTCOMES

1. Learner can demonstrate understanding by their ability to describe various parts of speech like noun, pronoun, verb adverb, adjective, conjunction, interjection and preposition.
2. Learner will be able to understand the articles and use effectively.
3. Learner can demonstrate understanding by their ability to describe various parts of speech like noun, pronoun, verb adverb, adjective, conjunction, interjection and preposition.
4. Learner will be able to understand the articles and use effectively.
5. Students will learn advance tricky approaches for solving Quant.
6. It will enhance student's skill to appear in various aptitude test within limited time constrain.

UNIT I:

Basic Grammar-Tenses, Subject-Verb- Agreement ,Article, formation of sentences.Introduction to technical terms.

UNIT II:

Self Introduction , body language, Voice modulation, E-mail writing ,Letter writing,Minutes writing,Resume building

UNIT III:

Simplification & Approximation

Alphanumeric Series & Miscellaneous

Coding-Decoding

Reference Books:

1.Quantitative Aptitude- R.S Agarwal

2.Analytical Reasoning –Peeyush Bhardwaj

3.English Grammar, Composition and Usage by N.K. Agrawal & F.T. Wood, Macmillan India Ltd., New Delhi.

COURSE CODE & NAME: CAUCBC114P / Python Programming Lab

COURSE OUTCOMES

1. Understand and comprehend the basics of python programming.
2. Demonstrate the principles of structured programming and be able to describe, design, implement, and test structured programs using currently accepted methodology.
3. Explain the use of the built-in data structures list, sets, tuples and dictionary.
4. Make use of functions and its applications.

List of Experiments:

1. Write a program to demonstrate different number data types in Python
2. Write a program to compute distance between two points taking input from the user using Pythagorean Theorem.
3. Write a Program for checking whether the given number is a even number or not.
4. Write a Python script that prints prime numbers less than 20.
5. Write a program to create, concatenate and print a string and accessing sub-string from a given string.
6. Write a program to create, append, and remove lists in python.
7. Write a program to demonstrate working with tuples in python.
8. Write a python program to define a module to find Fibonacci Numbers and import the module to another program.
9. Write a python program to define a module and import a specific function in that module to another program
10. Write a script named copyfile.py. This script should prompt the user for the names of two text files. The contents of the first file should be input and written to the second file.
11. Write a Python class to implement pow(x, n).

COURSE CODE & NAME: CAUCBC104P / Digital Logic Design Lab

COURSE OUTCOMES

1. Analyze and verify the truth tables of basic and universal gates to demonstrate their functionality.
2. Apply simplification techniques to logical expressions and implement them using basic and universal gates.
3. Evaluate the significance of NAND and NOR gates as universal gates through practical experiments.
4. Construct and implement Half Adders and Full Adders using both basic and universal gates to understand their operations.
5. Design and realize Half Subtractors, Full Subtractors, and a Binary to Grey code generator using appropriate gate configurations.

List of Experiments:

1. To study and verify the truth table of basic gates.
2. To study and verify the truth table of universal gate.
3. To simplify the given expression and to realize it using basic gates and universal gate.
4. To realize why NAND gate is known as the universal gate.
5. To realize why NOR gate is known as the universal gate.
6. Realization of Half Adder and Full Adder by using Basic gates.
7. Realization of Half Adder and Full Adder by using universal gate.
8. Realization of Half Subtractor and Full Subtractor by using Basic gates.
9. Realization of Half Subtractor and Full Subtractor by using universal gate.
10. Realization of Binary to grey generator.

COURSE CODE & NAME: CAUIBC106T / Programming in Python and Clean Code Principle

COURSE OUTCOMES

1. UUnderstand the principles of clean code and the importance of writing maintainable and readable code.
2. Learn Python-specific best practices for writing efficient and clean code.
3. Acquire skills in refactoring code to improve its structure and readability.
4. Develop proficiency in testing and debugging Python code.
5. Apply clean coding techniques to real-world projects and collaborate effectively using version control.

UNIT I:

Introduction to Clean Code : Introduction to bad and clean code, Purpose of Clean Code, Importance of clean code in software development, Characteristics of clean code, Key principles of clean code, **Python Fundamentals for Clean Coding :** Definition with Real Use Cases, History of Python, How Python is installed, Python IDE, Python Basics: Syntax, variables, data types, Execution of the basic program of the python, Character set, Token, core, input(), eval () & print() function, f-String, Operators and Expressions, Type Conversion

UNIT II:

Conditional & Control Statements : Decision Making statements (if,else,elif statement, its working and execution), Nested-if statement and Elif statement in Python, Conditional Expressions Evaluation & Float Representation, Boolean type, Boolean operators, String Operators, While Loop, For Loop, Nested loop, Break, Continue and Pass statements, range () Function

UNIT III:

Functions and Data Structures : Syntax and Basics of Functions, Use of functions, Parameters and Arguments, local & global Scope of variable, return statement, recursive function, Lambda, Map, Filter, and Reduce, Str class, inbuilt functions of string, traversal of string, string operator & operations, creating a list, Tuple, Dictionaries & sets, In built functions of list, tuple, set & dictionaries, list operators

UNIT IV:

OOP in Python with Clean Coding : Introduction to OOP Concepts: Classes and objects, the concept of 'self', Constructor methods, Encapsulation, Abstraction, Inheritance and Polymorphism, Clean Coding in OOP: Principles of SOLID design, Code refactoring techniques, Writing DRY (Don't Repeat Yourself) code, Avoiding Code smells

UNIT V:

File Handling, Data Handling & Exception Management: Need of File Handling, Different modes of file handling, Read/Write text and numbers to/from a file, Directories on a disk, Error and Exception Handling (Types of errors: syntax, runtime, logical), Try, except, finally blocks, raising custom exceptions, Regular Expression Pattern Matching, working with special characters, date, emails, Parsing Data, Developing a small Python project

TEXTBOOKS

1. S Python Software Foundation, "The Python Standard Library".
2. Miguel Grinberg, "Flask Web Development: Developing Web Applications with Python", 2018.

Further suggested Readings:

1. "Refactoring: Improving the Design of Existing Code" by Martin Fowler.
2. "Effective Python: 90 Specific Ways to Write Better Python" by Brett Slatkin.

COURSE CODE & NAME: CAUIBC106P / Programming in Python and Clean Code Principle Lab

LAB COURSE OUTCOMES

1. Understand and apply the principles of clean and maintainable Python code.
2. Write efficient Python programs using clean coding standards and practices.
3. Refactor, debug, and test Python code to improve structure and readability.
4. Implement object-oriented design patterns following SOLID principles.
5. Apply file handling, data processing, and error handling in real-world projects using version control.

List of Experiments:

1.Write and Compare Clean vs. Messy Python Code
   a) .Objective: Understand clean code principles by comparing readable and unreadable scripts.
   b) Task: Rewrite a messy code snippet using clean coding practices.

2.Python Basics with Clean Syntax
   a) Objective: Practice clean syntax and structure in Python.
   b) Task: Write Python programs using variables, data types, and expressions with meaningful naming and consistent formatting.

3.Implementing Conditional Logic with Clean Code
   a) Objective: Practice clean coding in decision-making blocks.
   b) Task: Create a mini program using if, elif, and else, focusing on indentation, readability, and logic clarity.

4.Mastering Loops with Python Control Statements
   a) Objective: Implement for, while, break, continue, and pass with clarity.
   b) Task: Develop a program to filter and process data using loops and range efficiently.

5.Design and Use Python Functions with Clarity
   a) Objective: Learn modular design using functions.
   b) Task: Create a program using user-defined functions, default arguments, and meaningful docstrings.

6.Lambda, Map, Filter, and Reduce for Clean Functional Style
   a) Objective: Implement Python's functional programming tools.
   b) Task: Use lambda expressions with map/filter/reduce on sample datasets.

7.Work with Strings, Lists, Tuples, Sets, and Dictionaries
   a) Objective: Clean manipulation of built-in data structures.

b) Task: Perform operations on strings and collections, avoiding redundant code and improving clarity.

8.Clean OOP in Python with SOLID Principles
a) Objective: Implement object-oriented principles with clean coding practices.
b) Task: Create a class-based application following encapsulation, inheritance, and SOLID design.

9.Refactoring and Avoiding Code Smells
a) Objective: Identify and refactor "smelly" code.
b) Task: Refactor an existing script by removing duplication, improving naming, and applying DRY principle.

10.File Handling and Clean Data Management
a) Objective: Practice clean file input/output operations.
b) Task: Read from and write to text/CSV files using exception handling and clean path management.

11.Exception Handling with Custom Errors
a) Objective: Write robust code with error management.
b) Task: Build a Python application using try-except-finally, raise custom exceptions, and use meaningful error messages.

12.Mini Project: Clean Coding in a Real-world Python Project
a) Objective: Apply clean coding concepts in a complete project.
b) Task: Design a small application (e.g., contact manager, library system) using all units: functions, OOP, files, error handling, and clean code practices.