1. Create 2 lists L1 and L2 by accepting values from user. Perform following operations on them. Without using built-in any function of set class.
    1. Find L1-L2
    2. L2-L1
    3. Their Symmetric difference
2. Create a menu driven functionality on list with following menus.
   CRUD operations on dictionary.
    1. Create a list
    2. Add an element at the beginning of the list
    3. Add an element at the end of the list
    4. Remove element
    5. Display all the elements of the list.
    6. Exit
3. CRUD operations on dictionary
    1. Add a new record
    2. Update an existing record
    3. Delete a record
    4. Search
    5. Display all records

    You can select any one of case studies given below. Case Studies

    1. Employee
        1. Empid (PK, Numeric, must be unique)
        2. Empname (should be valid name)
        3. Dept (predefined list)
        4. Salary (Numeric)
    2. Book
        1. BookId (PK, Numeric, must be unique)
        2. Title (Only alphabets, digits and space allowed)
        3. Author (Only alphabets, digits)
        4. Type (Predefined list)
        5. Price
    3. Patient
        1. PatientId (PK, Numeric, must be unique)
        2. Patient Name (should be valid name)
        3. Blood group (Predefined List)
        4. Doctor Treating (should be valid name)
    4. Product
        1. Prod_code(Only alphabets and letters. Must be unique)
        2. Stock in hand (numeric only)
        3. Product_Category (Predefined List)

4. Price

5. Food
    1. foodID (PK, Numeric, must be unique)
    2. Food_name
    3. Category (Predefined List)
    4. Price

4. Reports

5. OOP: Write a Python class Account with attributes

1. Account Number

2. Account holder name

3. Balance

4. Type of account

And member functions to 1. Deposit 2. Withdraw 3. Balance Enquiry 4. Display all details.

While opening account, amount should be 3000 or more than 3000. While withdrawing the amount, minimum balance should be checked as 3000. Develop a menu driven functionality.

Use Currency, minimum balance, bank name as Class variables.

6. Define a class Time with data members: 1. Hrs 2. Minutes. The class will be used to operate the time of processes. Class should have a constructor to initialise the values. Overload the following operators,

1. + to add 2 time objects.
   Suppose a process P1 needs 2 hrs. 30 minutes and process p2 need 3 hrs. 40 minutes. So total time for both processes is 6 hrs. 10 minutes.
2. – to find difference of 2 processes.
   Suppose a process P1 needs 2 hrs. 30 minutes and process p2 need 3 hrs. 40 minutes. So, the difference of 2 processes is 1 Hr. 10 Minutes.(Duration should be positive)
3. * to find the number of times of the given time. E.g a process P1 needs 2 hr 10 minutes then, 3 times of the time is, 6 hrs. 30 minutes.

7. Operator Overloading Define a class Date with data members day, month and year. E.g today's date is 26-09-2024 then the data members will hold following values,day = 26 month = 09 year = 2024

Overload the operator + to find tomorrow's date. Consider the given date is valid date.

8. Inheritance: Define a class Quadrilateral which will initialise the length of four sides using constructor. Also will have a method perimeter() which will find the perimeter. Derive 2

classes from Quadrilateral named Rectangle and Square which will use base class's constructor to initialise the sides also will have their own methods to calculate area.

9. Multiple Inheritance:

Write a program in Python as per the information given below.

Create a base class Student with data members registration no. and name. Create another base class Exam with data members examno., pattern, semester. Create a class Result which will inherit the classes Student and Exam and will be having data members as marks of 3 subjects and method to print the result. Use appropriate methods to set and display the data.