TW1

1. Write a Python program to read two lists and find the union and intersection of the two lists.
2. Write a function in Python which will return a list of all possible sub lists.

For example :

Input : [1, 2, 3, 4]

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

1. Develop a program using Sieve of Eratosthenes method for computing primes upto a specified number.

In mathematics, the Sieve of Eratosthenes is a simple, ancient algorithm for finding all prime numbers up to any given limit. It does so by iteratively marking as composite (i.e., not prime) the multiples of each prime, starting with the first prime number, 2. The multiples of a given prime are generated as a sequence of numbers starting from that prime, with constant difference between them that is equal to that prime.

1. Develop a menu driven program to implement a queue.

The operations would be

* 1. Add an item to the queue
  2. Delete an item from queue
  3. Display the queue

TW2

1. Develop a menu driven Python program to implement the mapping:

{USN1: [m1,m2,m3], USN2: [m1,m2,m3]....} and perform the following using functions:

1. Add an entry
2. Delete an entry
3. Display all entries
4. Compute and display average of best two marks for a specific USN
5. Store the following information in a dictionary:

Course Code : Course Name, Faculty, Number of registrations.

Perform the following operations using functions:

1. Find Course Name that has highest number of registrations.
2. Given the Course Code, display the associated details.
3. Display details of all courses.
4. Given a list of valid words and a list of characters, print all valid words that are possible using characters from the list. Repetitions of characters is not allowed.

Examples:

Input : validWords = ["go","bat","me","eat","goal","boy", "run"]

char\_list = ['e','o','b', 'a','m','g', 'l']

Output : go, me, goal.

1. Create a dictionary to store player information.

playerName :numMatches, numCenturies, numWickets

Write functions that solve the following queries:

1. Player with highest number of centuries
2. Player with highest number of wickets
3. Number of Players who have scored at least 10 centuries and taken at least 100 wickets
4. Number of players who have taken played at least 50 matches
5. Initialize a Dictionary List with StudentName and Date of Birth
6. Sort the Dictionary

i. In ascending order of StudentName

ii. In ascending order of Date Of Birth (if StudentName is same).

Input:

{'name': 'Santosh', 'd.o.b': '1997-03-02'}

{'name': 'Atharva', 'd.o.b': '1997-01-04'}

{'name': 'Rajat', 'd.o.b': '1997-09-13'}

{'name': 'Akash', 'd.o.b': '1997-03-01'}

{'name': 'Atharva', 'd.o.b': '1996-02-05'}

Output:

{'name': 'Akash', 'd.o.b': '1997-03-01'}

{'name': 'Atharva', 'd.o.b': '1996-02-05'}

{'name': 'Atharva', 'd.o.b': '1997-01-04'}

{'name': 'Rajat', 'd.o.b': '1997-09-13'}

{'name': 'Santosh', 'd.o.b': '1997-03-02'}

TW3

1.

Write a Python program to read the book information from the user and store in a CSV file containing rows in the following format:

bookNo, title, author, price

Develop a menu-driven program (with functions for each) with the following options:

1:Search Book by author

2:Search Books below specified price (Raise an exception if price entered is <= 0)

3:Search Books where title contains the specified word

4:Exit

2.

Develop a python program which reads details of students who wrote an exam from a file namely in.txt with records: USN, Marks, College and name. Find the names of all students from GIT who scored more than 80 marks and write the names to a file out.txt. Marks should be valid, USN should start with "GI" and college should be GIT. Handle exceptions if any.

3. Develop a Python program with exception handling to read n Meal records into a CSV file. Each meal could be breakfast, lunch or dinner followed by one or more items. For each meal (breakfast, lunch and dinner), find all the items and store them in an output CSV file.

4. Write a Python program to read from a file in.txt and write the contents in reverse order to file out.txt, raise an exception if there is no content in in.txt.

TW 4

1.A specialist (Pediatrician, Cardiologist) is a doctor who treats the patients in his specialty .Develop a python program to demonstrate polymorphism. UML is as shown below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  | | --- | | **Doctor** | | name:String  worksAtHospital:String | | setSalary()  treatsPatient() | | |
| |  | | --- | | **Pediatrician** | | setSalary()  treatsPatient() | | |  | | --- | | **Cardiologist** | | setSalary()  treatsPatient() | |

2.Create a Person class that provides attributes for first name, last name, and email address. This class should provide a property or method that returns the person’s full name.

Create a Customer class that inherits the Person class. This class should add an attribute for a customer number.

Create an Employee class that inherits the Person class. This class should add an attribute for a PAN number.

The program should create a Customer or Employee object from the data entered by the user, and it should use this object to display the data to the user. To do that, the program can use the isinstance() function to check whether an object is a Customer or Employee object.

3.Kindly find the following pbm definition:

Create a class called Manager with attributes: name, ID and basic salary. Demonstrate polymorphism by deriving Classes HR Manager and Sales Manager from Manager and compute gross salary as per the following:

HR Manager -  DA = 70% of basic, HRA = 20% of basic, deductions = 5% of basic

Sales Manager - DA = 70% of basic, HRA = 10% of basic, TA = 5% of basic, deductions = 5% of basic