Muhammad Zain

19F-0228

Lab 2

AI LAB

Task1:

import random  
class LIBRARY:  
 # ==============================================================================  
 # Constructor  
 # ==============================================================================  
 def \_\_init\_\_(self):  
 print("Library Managment System")  
 self.stdName = None  
 self.rollNo = None  
 # ==============================================================================  
 # HardCoding the Books in Library  
 # ==============================================================================  
 Genre = ("reading books", "business", "designers", "machine learning", "coding")  
 readingBooks = ("Rich\_Dad\_Poor\_Dad", "Atomic\_Habits", "Learn\_to\_grow")  
 business = ("Good\_to\_greet", "zero\_to\_one", "The\_4\_Hour\_week")  
 designers = ("logo\_Design\_love", "know\_your\_onions")  
 machineLearning = ("python\_AI\_learning", "machine\_learning", "data\_mining")  
 Coding = ("Clean\_Code", "Design\_Patterns", "Crack\_to\_Code")  
 # ==============================================================================  
 # Getter Functions  
 # ==============================================================================  
  
 def getName(self):  
 return self.stdName  
  
 def getRollNo(self):  
 return self.rollNo  
 # ==============================================================================  
 # Setter Functions  
 # ==============================================================================  
  
 def setName(self, name\_):  
 self.stdName = name\_  
  
 def setRollNo(self, rollName\_):  
 self.rollName = rollName\_  
 # ==============================================================================  
 # Date Function  
 # ==============================================================================  
  
 def Date\_(self):  
 return print(random.randint(1,30), "-", random.randint(3,12), "- 2022")  
 # ==============================================================================  
 # Selecting Categories to get the relevant book  
 # ==============================================================================  
  
 def Category(self):  
 print("Categories of Books")  
 for i in range(len(self.Genre)):  
 print((i+1)," => ", self.Genre[i])  
 select = int(input("Select Book : "))  
 self.selectBook(select)  
 return  
 # ==============================================================================  
 # Add member into the Dictionary  
 # ==============================================================================  
  
 def addMember (self, Dictionary, Name\_, RollNo\_):  
 Dictionary[Name\_] = RollNo\_  
 self.Category()  
 return  
 # ==============================================================================  
 # function to check if the member is present in the list and then greet him/her  
 # ==============================================================================  
  
 def CheckAndGreet (self, BookDictionary, Name\_, RollNo\_):  
 for keys, values in BookDictionary.items():  
 if values == RollNo\_:  
 print("Hi", Name\_)  
 self.Category()  
 return  
 # if not present already just put it inside the dictionary  
 else:  
 self.addMember(BookDictionary, Name\_, RollNo\_)  
 # ==============================================================================  
 # function to print the books after selection of particular genre  
 # ==============================================================================  
  
 def printBooks(self,typeofGenre):  
 print("Books ",typeofGenre)  
 for i in range(len(typeofGenre)):  
 print((i + 1), ")", typeofGenre[i])  
 # ==============================================================================  
 # select the copy of book  
 # ==============================================================================  
  
 def selectBookCopy(self,options,typeofGenre):  
 for i in range(len(typeofGenre)):  
 if i == options:  
 print("Selected => ",typeofGenre[i])  
 print("Issuance till ", self.Date\_())  
 # ==============================================================================  
 # Menu for the selection of the books  
 # ==============================================================================  
  
 def selectBook(self, choice):  
 if choice == 1:  
 self.printBooks(self.readingBooks)  
 choice\_ = int(input("==Select the Book =="))  
 self.selectBookCopy(choice\_, self.readingBooks)  
  
 elif choice == 2:  
 self.printBooks(self.business)  
 choice\_ = int(input("==Select the Book =="))  
 self.selectBookCopy(choice\_,self.business)  
  
 elif choice == 3:  
 self.printBooks(self.designers)  
 choice\_ = int(input("==Select the Book =="))  
 self.selectBookCopy(choice\_,self.designers)  
  
 elif choice == 4:  
 self.printBooks(self.machineLearning)  
 choice\_ = int(input("==Select the Book =="))  
 self.selectBookCopy(choice\_,self.machineLearning)  
  
 elif choice == 5:  
 self.printBooks(self.Coding)  
 choice\_ = int(input("==Select the Book =="))  
 self.selectBookCopy(choice\_,self.Coding)  
  
 else:  
 print("invalid input")  
  
# ==============================================================================  
# Library Members  
# ==============================================================================  
Dictionary = {  
 "Harry": "20F0199",  
 "Kante": "21f0291",  
 "David": "22f0382",  
 "Weise": "24f0473",  
 "Alexa": "24f0564"  
 }  
objLib = LIBRARY()  
Name\_ = input("Input your name : ")  
objLib.setName(Name\_)  
RollNo\_ = input("Input your Roll No: ")  
objLib.setRollNo(RollNo\_)  
objLib.CheckAndGreet(Dictionary, objLib.getName(), objLib.getRollNo())

Screenshot:

Text

Description automatically generated

Task2:

def append\_Vertex(vertex):  
 global graph  
 global vertices\_no  
 if vertex in graph:  
 print("\*\*\*", vertex, " Vertex already exists \*\*\*")  
 else:  
 vertices\_no = vertices\_no + 1  
 graph[vertex] = []  
def append\_edge(vertex1, vertex2):  
 global graph  
 if vertex1 not in graph:  
 print("\*\*\* ", vertex1, " Vertex not exist \*\*\*")  
 elif vertex2 not in graph:  
 print("\*\*\* ", vertex2, " Vertex not exist \*\*\*")  
 else:  
 temp = [vertex2]  
 graph[vertex1].append(temp)  
  
def print\_graph():  
 global graph  
 for vertex in graph:  
 for edges in graph[vertex]:  
 print (vertex, " -> ", edges[0])  
graph = {}  
vertices\_no = 0  
append\_Vertex('A')  
append\_Vertex('B')  
append\_Vertex('C')  
append\_Vertex('D')  
append\_Vertex('E')  
append\_Vertex('F')  
append\_Vertex('G')  
append\_Vertex('H')  
append\_edge('A', 'B')  
append\_edge('A', 'C')  
append\_edge('A', 'D')  
append\_edge('B', 'E')  
append\_edge('B', 'F')  
append\_edge('E', 'G')  
append\_edge('F', 'H')  
print("\*\*\* Ajency List Using Dictionay \*\*\*")  
print("\*\*\* Print Ajency List \*\*\*")  
print\_graph()

Screenshot:

Text

Description automatically generated