BRAC UNIVERSITY

Inspiring Excellence

Course Code:	**CSE111**
Course Title:	**Programming Language II**
Homework No:	**06**
Topic:	**Class variable and class method**
Submission Type:	**Soft Copy**
Resources:	**1. Class lectures** **2. BuX lectures** **a. English:** **i. Class/Static variable: https://shorturl.at/ceBJZ** **ii. Class method: https://shorturl.at/clnwy** **b. Supplementary:**

	i. **Class/Static Variable:** https://shorturl.at/dLRV6 ii. **Class method:** https://shorturl.at/cdqrY

Task 1

There is a hype in the movie theaters these days where many people are going to watch movies like Barbie, Oppenheimer and many other Bengali movies. Suppose a movie theater called Star Cinema is opening new branches where they are adding movies to each branch to show to the audience. You need to design two classes named StarCinema and Movie with a HAS-A relationship using the concept of class/ static variables and class methods.

Class Name	Instance Variables	Description
Movie	name: string	Name of the movie.
	genre: string	Movie Genre.
	duration: int	Duration of the movie in minutes.

Class Name	Instance Methods	Description
Movie	movieInfo()	returns a string displaying the name, genre and duration of the movie.

Class Name	Class Methods	Description
Movie	createMovie_fromString (string)	takes a single string of name, genre, duration of a movie separated by '-'. Ex: 'Prohelika-Drama-153' This method creates a new Movie object using this string and returns the reference of that object.

Class Name	Instance Variables	Description
StarCinema	branch_name: string	Stores the name of the StarCinema branch.
	movie_list: list	Stores list of Movie objects added to a branch.

Class Name	Class Variables	Description
StarCinema	all_branch_info: dictionary	Stores the branch name of Star Cinema branches as keys and the list of movie objects of each branch as values.

Class Name	Instance Methods	Description
StarCinema	addMovies(*movie_objects)	The method takes an unknown number of objects of Movie class. It adds each movie object to the movie_list list if the movie does not exist in the list yet. all_branch_info dictionary should also be updated accordingly. (This method may be called multiple times to add more movie objects for a StarCinema branch object.) **Ex:** If branch name is Mohakhali, all_branch_info = {‘Mohakhali’: [movie1, movie2, movie3]} If any movie name already exists in the movie_list list, you should not add the movie again to the list nor should you update the dictionary.
StarCinema	removeMovie(Movie Object)	The method takes a Movie object as argument and removes that movie from the movie_list and updates the class variable all_branch_info accordingly.

Class Name	Class Methods	Description
StarCinema	check(string)	Takes a movie name as an argument and shows which branches it is streaming in and the duration and genre of that movie.

		If the movie is not streaming in any branch, it should show that it is not being streamed through a print statement.
StarCinema	showAllBranchInfo()	This method prints the information extracted from the all_branch_info variable in the following manner: Branch Name: ____________ Movie No: 1 Movie Name: _____________ Movie Genre: _____________ Movie Duration: ____________ Movie No: 2 Movie Name: _____________ Movie Genre: _____________ Movie Duration: ____________ ……………………………………………………… …………………………………………………….. Note: Make sure to use moveInfo() method from the Movie class to display the information of each movie.

Task:
Design the two classes and implement the necessary methods mentioned above. You need to write both the class and driver code on your own and check if each method is working properly.

But to help you out, a demo driver code and output is given. Do not copy-paste the same driver code. This is given just to give you an idea.

Note: You can adopt other ways to solve this task as long as the method is giving the desired output.

Task 2

```python
class FinalT6A:
    temp = 3

    def __init__(self, x, p):
        self.sum, self.y = 0, 2
        FinalT6A.temp += 3
        self.y = self.temp - p
        self.sum = self.temp + x
        print(x, self.y, self.sum)

    def methodA(self):
        x, y = 0, 0
        y = y + self.y
        x = self.y + 2 + self.temp
        self.sum = x + y + self.methodB(self.temp, y)
        print(x, y, self.sum)

    def methodB(self, temp, n):
        x = 0
        FinalT6A.temp += 1
        self.y = self.y + (FinalT6A.temp)
        FinalT6A.temp -= 1
        x = x + 2 + n
        self.sum = self.sum + x + self.y
        print(x, self.y, self.sum)
        return self.sum
```

```python
q1 = FinalT6A(2,1)
q1.methodA()
q1.methodA()
```

x	y	sum

Task 3

```python
class msgClass:
    def __init__(self):
        self.content = 0

class Quiz3:
    x = 0
    def __init__(self, k = None):
        self.sum, self.y = 0, 0
        if k is None:
            self.sum = 5
            Quiz3.x = 2
            self.y = 2
        else:
            self.sum = self.sum + k
            self.y = 3
            Quiz3.x += 2
    def methodA(self):
        x = 1
        y = 1
        msg = [None]
        myMsg = msgClass()
        myMsg.content = Quiz3.x
        msg[0] = myMsg
        msg[0].content = self.y + myMsg.content
        self.y = self.y + self.methodB(msg[0])
        y = self.methodB(msg[0]) + self.y
        x = y + self.methodB(msg, msg[0])
        self.sum = x + y + msg[0].content
        print(x, y, self.sum)
    def methodB(self, *args):
        if len(args) == 2:
            mg2, mg1 = args
            x = 2
            self.y = self.y + mg2[0].content
            mg2[0].content = self.y + mg1.content
            x = x + 2 + mg1.content
            self.sum = self.sum + x + self.y
            mg1.content = self.sum - mg2[0].content
            print(Quiz3.x, self.y, self.sum)
            return self.sum

        elif len(args) == 1:
            mg1, = args
            x = 1
            y = 2
            y = self.sum + mg1.content
            self.y = y + mg1.content
            x = Quiz3.x + 5 + mg1.content
            self.sum = self.sum + x + y
            Quiz3.x = mg1.content + x + 3
            print(x, y, self.sum)
            return y
```

```python
a1 = Quiz3()
a2 = Quiz3(5)
msg = msgClass()
a1.methodA()
a2.methodB(msg)
```

x	y	sum