

| **Course Code:** | **CSE111** |
| --- | --- |
| **Course Title:** | **Programming Language II** |
| **Homework No:** | **06** |
| **Topic:** | **Class variable and class method** |
| **Submission Type:** | **Submission Link:** [**https://docs.google.com/forms/d/e/1FAIpQLSd0xIJOv3-Fp9dTtoeu8GOWN2APb0PMV3EVb9Vc1E61jMuyqw/viewform**](https://docs.google.com/forms/d/e/1FAIpQLSd0xIJOv3-Fp9dTtoeu8GOWN2APb0PMV3EVb9Vc1E61jMuyqw/viewform) |
| **Resources:** | 1. **Class lectures** 2. **BuX lectures**    1. **English:**        1. **Class/Static variable:** [**https://shorturl.at/ceBJZ**](https://shorturl.at/ceBJZ)       2. **Class method:** [**https://shorturl.at/clnwy**](https://shorturl.at/clnwy)    2. **Supplementary:**        1. **Class/Static Variable:** [**https://shorturl.at/dLRV6**](https://shorturl.at/dLRV6)       2. **Class method:** [**https://shorturl.at/cdqrY**](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](https://docs.google.com/document/d/1c6MhBat1bBfuvRuFyz_-_ntcU9Y1oRlIWcbxHHJlLfw/edit?usp=sharing) 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**

| **1** | **class FinalT6A:** |
| --- | --- |
| **2** | **temp = 3** |
| **4** | **def \_\_init\_\_(self, x, p):** |
| **5** | **self.sum, self.y = 0, 2** |
| **6** | **FinalT6A.temp += 3** |
| **7** | **self.y = self.temp - p** |
| **8** | **self.sum = self.temp + x** |
| **9** | **print(x, self.y, self.sum)** |
| **11** | **def methodA(self):** |
| **12** | **x, y = 0, 0** |
| **13** | **y = y + self.y** |
| **14** | **x = self.y + 2 + self.temp** |
| **15** | **self.sum = x + y + self.methodB(self.temp, y)** |
| **16** | **print(x, y, self.sum)** |
| **18** | **def methodB(self, temp, n):** |
| **19** | **x = 0** |
| **20** | **FinalT6A.temp += 1** |
| **21** | **self.y = self.y + (FinalT6A.temp)** |
| **22** | **FinalT6A.temp -= 1** |
| **23** | **x = x + 2 + n** |
| **24** | **self.sum = self.sum + x + self.y** |
| **25** | **print(x, self.y, self.sum)** |
| **26** | **return self.sum** |

| **q1 = FinalT6A(2,1)**  **q1.methodA()**  **q1.methodA()** | **x** | **y** | **sum** |
| --- | --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Task 3**

| **1** | **class msgClass:** |
| --- | --- |
| **2** | **def \_\_init\_\_(self):** |
| **3** | **self.content = 0** |
| **4** |  |
| **5** | **class Quiz3:** |
| **6** | **x = 0** |
| **7** | **def \_\_init\_\_(self, k = None):** |
| **8** | **self.sum, self.y = 0, 0** |
| **9** | **if k is None:** |
| **10** | **self.sum = 5** |
| **11** | **Quiz3.x = 2** |
| **12** | **self.y = 2** |
| **13** | **else:** |
| **14** | **self.sum = self.sum + k** |
| **15** | **self.y = 3** |
| **16** | **Quiz3.x += 2** |
| **17** | **def methodA(self):** |
| **18** | **x = 1** |
| **19** | **y = 1** |
| **20** | **msg = [None]** |
| **21** | **myMsg = msgClass()** |
| **22** | **myMsg.content = Quiz3.x** |
| **23** | **msg[0] = myMsg** |
| **24** | **msg[0].content = self.y + myMsg.content** |
| **25** | **self.y = self.y + self.methodB(msg[0])** |
| **26** | **y = self.methodB(msg[0]) + self.y** |
| **27** | **x = y + self.methodB(msg, msg[0])** |
| **28** | **self.sum = x + y + msg[0].content** |
| **29** | **print(x, y, self.sum)** |
| **30** | **def methodB(self, \*args):** |
| **31** | **if len(args) == 2:** |
| **32** | **mg2, mg1 = args** |
| **33** | **x = 2** |
| **34** | **self.y = self.y + mg2[0].content** |
| **35** | **mg2[0].content = self.y + mg1.content** |
| **36** | **x = x + 2 + mg1.content** |
| **37** | **self.sum = self.sum + x + self.y** |
| **38** | **mg1.content = self.sum - mg2[0].content** |
| **39** | **print(Quiz3.x, self.y, self.sum)** |
| **40** | **return self.sum** |
| **41** |  |
| **42** | **elif len(args) == 1:** |
| **43** | **mg1, = args** |
| **44** | **x = 1** |
| **45** | **y = 2** |
| **46** | **y = self.sum + mg1.content** |
| **47** | **self.y = y + mg1.content** |
| **48** | **x = Quiz3.x + 5 + mg1.content** |
| **49** | **self.sum = self.sum + x + y** |
| **50** | **Quiz3.x = mg1.content + x + 3** |
| **51** | **print(x, y, self.sum)** |
| **52** | **return y** |

| **a1 = Quiz3()**  **a2 = Quiz3(5)**  **msg = msgClass()**  **a1.methodA()**  **a2.methodB(msg)** | **x** | **y** | **sum** |
| --- | --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |