Inspiring Excellence

Course Code:	**CSE111**
Course Title:	**Programming Language II**
Homework No:	**04**
Topic:	**OOP (Classes and objects)**
Submission Type:	**Hard Copy (Only submit the part of the code that you have been instructed to write. DO NOT write any given code.)**
Resources:	**1. Class lectures** **2. BuX lectures** **a. English: https://shorturl.at/dhjAZ** **b. Supplementary: https://shorturl.at/wMPRU** **https://shorturl.at/uwENV**

Task 1

Design the **Customer** class with the necessary properties so that the following output is produced.

[**Hint:**

- If the visitor's age is greater than 10, then the ticket price is 100 taka. Otherwise, 50 taka.
- A customer can't buy more than 3 tickets.]

Driver Code	Output

`print('1-------------------------')`
`customer1 = Customer()`
`print('2-------------------------')`
`customer1.buyTicket('Bob', 23)`
`customer1.buyTicket('Henry', 7)`
`customer1.buyTicket('Alexa', 30)`
`customer1.buyTicket('Jonas', 43)`
`print('3-------------------------')`
`customer1.showDetails()`
`print('4-------------------------')`
`customer2 = Customer()`
`print('5-------------------------')`
`customer2.buyTicket('Harry', 60)`
`customer2.buyTicket('Tomas', 28)`
`print('6-------------------------')`
`customer2.showDetails()`
	`1-------------------------`
	`Welcome to ABC Memorial Park`
	`2-------------------------`
	`Successfully purchased a ticket for Bob!`
	`Successfully purchased a ticket for Henry!`
	`Successfully purchased a ticket for Alexa!`
	`You can't buy more than 3 tickets`
	`3-------------------------`
	
	`Total price: 250 Taka`
	`4-------------------------`
	`Welcome to ABC Memorial Park`
	`5-------------------------`
	`Successfully purchased a ticket for Harry!`
	`Successfully purchased a ticket for Tomas!`
	`6-------------------------`
	`Amount of tickets: 2`
	`Total price: 200 Taka`

Task 2

The Giant Panda Protection and Research Center in the Sichuan province of southwest China, actually employs a category of workers known as panda nannies. The primary responsibility is to play with adorable panda cubs and name them, determine gender, keep track of their age and hours they sleep. So being a programmer panda nanny, you will create a code that will do all these works for you.

1. Create a class named **Panda** and also write the constructor.
2. Access the instance attributes and print them in the given format.
3. Call instance methods to keep track of their daily hours of sleep.
4. Suppose consulting with other panda nannies you have set some criteria based on which you will make their diet plans. The criteria are:
   ** Mixed Veggies for pandas having 3 to 5 hours (included) of sleep daily.
   ** Eggplant & Tofu for pandas having 6 to 8 hours (included) of sleep daily.
   ** Broccoli Chicken for pandas having 9 to 11 hours (included) of sleep daily.
   ** Lastly for all other arguments, then just give it bamboo leaves.

Now handle this problem modifying the method designed to keep track of their daily hours of sleep and determine diet plan.

[You are not allowed to change the code below]

#Write your code here for subtasks 1-4. panda1 = Panda("Kunfu", "Male", 5) panda2 = Panda("Pan Pan", "Female",3) panda3 = Panda("Ming Ming", "Female",8) print("{} is a {} Panda Bear who is {} years old".format(panda1.name,panda1.gender,panda1.age)) print("{} is a {} Panda Bear who is {} years old".format(panda2.name,panda2.gender,panda2.age)) print("{} is a {} Panda Bear who is {} years old".format(panda3.name,panda3.gender,panda3.age)) print("==========") print(panda2.sleep(10)) print(panda1.sleep(4)) print(panda3.sleep(13))	***OUTPUT:*** Kunfu is a Male Panda Bear who is 5 years old Pan Pan is a Female Panda Bear who is 3 years old Ming Ming is a Female Panda Bear who is 8 years old ========== Pan Pan sleeps 10 hours daily and should have Broccoli Chicken Kunfu sleeps 4 hours daily and should have Mixed Veggies Ming Ming's duration is 13 thus should have only bamboo leaves

Task 3

Suppose you are the CEO of "Green Phone". After a meeting with the R&D department and sales department, you decided to launch 3 smartphone series, 'A', 'M' and 'U' series. These series will get 2 years, 3 years and 4 years of software update respectively. Now, design a **GreenPhone** class with necessary properties so that it generates the output below for the given driver code.

[**Hint:** updatePhone() method will upgrade the android version of the phone.]

Driver Code	Output

```
print('1======================')
p1 = GreenPhone('A1', 12, 3)
p2 = GreenPhone('M11', 12, 4)
p3 = GreenPhone('U20', 12, 5)
p1.showSpecification()
print('2======================')
p2.showSpecification()
print('3======================')
p1.updatePhone()
print('4======================')
p1.updatePhone()
p2.updatePhone()
p3.updatePhone()
print('5======================')
p1.updatePhone()
p2.updatePhone()
p3.updatePhone()
print('6======================')
p2.updatePhone()
p3.updatePhone()
print('7======================')
p1.showSpecification()
p3.showSpecification()
```

```
1==============================
Phone Company: GreenPhone
Model Name: A1
Android Version: 12
Number of Cameras: 3
2==============================
Phone Company: GreenPhone
Model Name: M11
Android Version: 12
Number of Cameras: 4
3==============================
Your phone Greenphone A1 is upgraded to
Android Version: 13.
4==============================
Your phone Greenphone A1 is upgraded to
Android Version: 14.
Your phone Greenphone M11 is upgraded to
Android Version: 13.
Your phone Greenphone U20 is upgraded to
Android Version: 13.
5==============================
Your phone Greenphone A1 is already up to
date.
Your phone Greenphone M11 is upgraded to
Android Version: 14.
Your phone Greenphone U20 is upgraded to
Android Version: 14.
6==============================
Your phone Greenphone M11 is upgraded to
Android Version: 15.
Your phone Greenphone U20 is upgraded to
Android Version: 15.
7==============================
Phone Company: GreenPhone
Model Name: A1
Android Version: 14
Number of Cameras: 3
Phone Company: GreenPhone
Model Name: U20
Android Version: 15
Number of Cameras: 5
```

Task 4

Design **StudentDatabase** class so that the following output is produced: Calculation of GPA:
GPA = Sum of (Grade Points * Credits)/ Credits attempted

- Each course a student takes is of 3 credits.
- **For example:** Wanda has taken 3 courses in Summer 2022 semester. So her CGPA will be

  [(CSE111 GP × 3) + (CSE260 GP × 3) + (ENG101 GP × 3)] / (3 courses × 3)

  [(3.7 × 3) + (3.7 × 3) + (4.0 × 3)] / (3 × 3) = **3.8**

Driver Code	Output

`# Write your code here`
`s1 = StudentDatabase('Pietro', '10101222')`
`s1.calculateGPA(['CSE230: 4.0', 'CSE220: 4.0', 'MAT110: 4.0'], 'Summer2020')`
`s1.calculateGPA(['CSE250: 3.7', 'CSE330: 4.0'], 'Summer2021')`
`print(f'Grades for {s1.name}\n{s1.grades}')`
`print('---------------------------------')`
`s1.printDetails()`
`s2 = StudentDatabase('Wanda', '10103332')`
`s2.calculateGPA(['CSE111: 3.7', 'CSE260: 3.7', 'ENG101: 4.0'], 'Summer2022')`
`print('---------------------------------')`
`print(f'Grades for {s2.name}\n{s2.grades}')`
`print('---------------------------------')`
`s2.printDetails()`
	`Grades for Pietro`
	`{'Summer2020': {('CSE230', 'CSE220', 'MAT110'): 4.0}, 'Summer2021': {('CSE250', 'CSE330'): 3.85}}`
	`---------------------------------`
	`Name: Pietro`
	`ID: 10101222`
	`Courses taken in Summer2020:`
	`CSE230`
	`CSE220`
	`MAT110`
	`CGPA: 4.0`
	`Courses taken in Summer2021:`
	`CSE250`
	`CSE330`
	`CGPA: 3.85`
	`---------------------------------`
	`Grades for Wanda`
	`{'Summer2022': {('CSE111', 'CSE260', 'ENG101'): 3.8}}`
	`---------------------------------`
	`Name: Wanda`
	`ID: 10103332`
	`Courses taken in Summer2022:`
	`CSE111`
	`CSE260`
	`ENG101`
	`CGPA: 3.8`

Task 5

```
1  class Scope:
2      def __init__(self):
3          self.x, self.y = 1, 100
4      def met1(self):
5          x = 3
6          x = self.x + 1
7          self.y = self.y + self.x + 1
8          x = self.y + self.met2() + self.y
9          print(x, self.y)
10     def met2(self):
11         y = 0
12         print(self.x, y)
13         self.x = self.x + y
14         self.y = self.y + 200
15         return self.x + y
```

Write the output of the following code: q2 = Scope() q2.met1() q2.met2() q2.met1() q2.met2()	x	y

Task 6

```
1  class Test3:
2      def __init__(self):
3          self.sum, self.y = 0, 0
4      def methodA(self):
5          x, y = 2, 3
6          msg = [0]
7          msg[0] = 3
8          y = self.y + msg[0]
9          self.methodB(msg, msg[0])
10         x = self.y + msg[0]
11         self.sum = x + y + msg[0]
12         print(x, y, self.sum)
13     def methodB(self, mg2, mg1):
14         x = 0
15         self.y = self.y + mg2[0]
16         x = x + 33 + mg1
17         self.sum = self.sum + x + self.y
18         mg2[0] = self.y + mg1
19         mg1 = mg1 + x + 2
20         print(x, self.y, self.sum)
```

Write the output of the following code:

```
t3 = Test3()
t3.methodA()
t3.methodA()
t3.methodA()
t3.methodA()
```

x	y	sum

Task 7

```
1   class FinalT6A:
2       def __init__(self, x, p):
3           self.temp, self.sum, self.y = 4, 0, 1
4           self.temp += 1
5           self.y = self.temp - p
6           self.sum = self.temp + x
7           print(x, self.y, self.sum)
8       def methodA(self):
9           x = 0
10          y = 0
11          y = y + self.y
12          x = self.y + 2 + self.temp
13          self.sum = x + y + self.methodB(self.temp, y)
14          print(x, y, self.sum)
15      def methodB(self, temp, n):
16          x = 0
17          temp += 1
18          self.y = self.y + temp
19          x = x + 3 + n
20          self.sum = self.sum + x + self.y
21          print(x, self.y, self.sum)
22          return self.sum
```

What is the output of the following code sequence?

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

x	y	sum