



Inspiring Excellence

Course Code:	CSE111
Course Title:	Programming Language II
Lab No:	06
Topic:	OOP (Class variable and class method)
Number of tasks:	9

Task 1

Write a **Student** class to get the desired output as shown below.

1. Create a Student class and a class variable called ID initialized with 0.
2. Create a constructor that takes 4 parameters: name, department, age and cgpa.
3. Write a **showDetails()** method to represent all the details of a Student
4. Write a *class method* **from_String()** that takes 1 parameter which includes name, department, age and cgpa all four attributes in string.

<p>#Write your code here for subtasks 1-6.</p> <pre>s1 = Student("Samin", "CSE", 21, 3.91) s1.showDetails() print("-----") s2 = Student("Fahim", "ECE", 21, 3.85) s2.showDetails() print("-----") s3 = Student("Tahura", "EEE", 22, 3.01) s3.showDetails() print("-----") s4 = Student.from_String("Sumaiya-BBA-23-3.96") s4.showDetails()</pre> <p># Write the answer of subtask 5 here</p> <p># Write the answer of subtask 6 here</p> <p>#You are not allowed to change the code above</p>	<p>OUTPUT</p> <pre>ID: 1 Name: Samin Department: CSE Age: 21 CGPA: 3.91 ----- ID: 2 Name: Fahim Department: ECE Age: 21 CGPA: 3.85 ----- ID: 3 Name: Tahura Department: EEE Age: 22 CGPA: 3.01 ----- ID: 4 Name: Sumaiya Department: BBA Age: 23 CGPA: 3.96</pre>
---	--

5. Explain the difference between a class variable and an instance variable. Print your answer at the very end of your code.
6. What is the difference between an instance method and class method? Print your answer at the very end

Task 2

Implement the design of the **Passenger** class so that the following output is produced:

The assumption is Bus base-fare is 450 taka. A passenger can carry upto 20 kg for free. 50 taka will be added if bag weight is between 21 and 50 kg. 100 taka will be added if bag weight is greater than 50 kg.

[You are not allowed to change the code below]

<pre># Write your code here print("Total Passenger:", Passenger.count) p1 = Passenger("Jack") p1.set_bag_weight(90) p2 = Passenger("Carol") p2.set_bag_weight(10) p3 = Passenger("Mike") p3.set_bag_weight(25) print("=====") p1.printDetail() print("=====") p2.printDetail() print("=====") p3.printDetail() print("=====") print("Total Passenger:", Passenger.count)</pre>	<p>Output:</p> <pre>Total Passenger: 0 ===== Name: Jack Bus Fare: 550 taka ===== Name: Carol Bus Fare: 450 taka ===== Name: Mike Bus Fare: 500 taka ===== Total Passenger: 3</pre>
---	---

Task 3

Implement the design of the **Travel** class so that the following output is produced:

[You are not allowed to change the code below]

<pre># Write your code here print("No. of Traveller =", Travel.count) print("=====") t1 = Travel("Dhaka","India") print(t1.display_travel_info()) print("=====") t2 = Travel("Kuala Lumpur","Dhaka") t2.set_time(23) print(t2.display_travel_info()) print("=====") t3 = Travel("Dhaka","New_Zealand") t3.set_time(15) t3.set_destination("Germany") print(t3.display_travel_info()) print("=====") t4 = Travel("Dhaka","India") t4.set_time(9) t4.set_source("Malaysia") t4.set_destination("Canada") print(t4.display_travel_info()) print("=====") print("No. of Traveller =", Travel.count)</pre>	<p>Output</p> <p>No. of Traveller = 0</p> <p>=====</p> <p>Source: Dhaka</p> <p>Destination:India</p> <p>Flight Time:1:00</p> <p>=====</p> <p>Source: Kuala Lumpur</p> <p>Destination:Dhaka</p> <p>Flight Time:23:00</p> <p>=====</p> <p>Source: Dhaka</p> <p>Destination:Germany</p> <p>Flight Time:15:00</p> <p>=====</p> <p>Source: Malaysia</p> <p>Destination:Canada</p> <p>Flight Time:9:00</p> <p>=====</p> <p>No of Traveller = 4</p>
--	---

Task 4

We know that Nike is opening their official outlets in Bangladesh. So let's construct a NikeBangladesh class so that they can keep track of their inventory and sales here,

Hint:

productSold()/restockProducts(): takes in a dictionary with product name and quantity, and updates the instance and class variables accordingly

Driver Code	Output
<pre>print("xxxxxxxxxxxxxxxx1xxxxxxxxxxxxxxxx") NikeBangladesh.status() dhaka = NikeBangladesh("Dhaka Banani") chittagong = NikeBangladesh("Chittagong GEC") print("xxxxxxxxxxxxxxxx2xxxxxxxxxxxxxxxx") dhaka.details() print("xxxxxxxxxxxxxxxx3xxxxxxxxxxxxxxxx") chittagong.details() print("xxxxxxxxxxxxxxxx4xxxxxxxxxxxxxxxx") dhaka.restockProducts({"Air Jordan":1200,"Cortez":200,"Zoom Kobe":200}) chittagong.restockProducts({"Air Jordan":1000,"Cortez":250,"Zoom Kobe":100}) print("xxxxxxxxxxxxxxxx5xxxxxxxxxxxxxxxx") NikeBangladesh.status() print("xxxxxxxxxxxxxxxx6xxxxxxxxxxxxxxxx") dhaka.productSold({"Air Jordan":760,"Cortez":90}) chittagong.productSold({"Air Jordan":520,"Zoom Kobe":70}) print("xxxxxxxxxxxxxxxx7xxxxxxxxxxxxxxxx") NikeBangladesh.status()</pre>	<pre>xxxxxxxxxxxxxxxx1xxxxxxxxxxxxxxxx Nike Bangladesh Status: Branches Opened: [] Currently Stocked {'Air Jordan': 0, 'Cortez': 0, 'Zoom Kobe': 0} Sold: 0 xxxxxxxxxxxxxxxx2xxxxxxxxxxxxxxxx Nike Dhaka Banani outlet: Products Currently Stocked: {'Air Jordan': 0, 'Cortez': 0, 'Zoom Kobe': 0} Sold: 0 xxxxxxxxxxxxxxxx3xxxxxxxxxxxxxxxx Nike Chittagong GEC outlet: Products Currently Stocked: {'Air Jordan': 0, 'Cortez': 0, 'Zoom Kobe': 0} Sold: 0 xxxxxxxxxxxxxxxx4xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx5xxxxxxxxxxxxxxxx Nike Bangladesh Status: Branches Opened: ['Dhaka Banani', 'Chittagong GEC'] Currently Stocked {'Air Jordan': 2200, 'Cortez': 450, 'Zoom Kobe': 300} Sold: 0 xxxxxxxxxxxxxxxx6xxxxxxxxxxxxxxxx xxxxxxxxxxxxxxxx7xxxxxxxxxxxxxxxx Nike Bangladesh Status: Branches Opened: ['Dhaka Banani', 'Chittagong GEC'] Currently Stocked {'Air Jordan': 920, 'Cortez': 360, 'Zoom Kobe': 230} Sold: 1440</pre>

Task 5

Write the **Student** class so that the given code provides the expected output.

1. Create **Student** class
2. Create 3 class variable
3. Create 1 class method for object creation
4. Create 1 class method for printing

[You are not allowed to change the code below]

Write your code here

```
Student.printDetails()
print('#####')

mikasa = Student('Mikasa Ackerman', "CSE")
mikasa.individualDetail()
print('-----')
Student.printDetails()

print('=====')

harry = Student.createStudent('Harry Potter', "Defence Against Dark
Arts", "Hogwarts School")
harry.individualDetail()
print('-----')
Student.printDetails()

print('=====')

levi = Student.createStudent("Levi Ackerman", "CSE")
levi.individualDetail()
print('-----')
Student.printDetails()
```

Output:

```
Total Student(s): 0
BRAC University Student(s): 0
Other Institution Student(s): 0
#####
Name: Mikasa Ackerman
Department: CSE
Institution: BRAC University
-----
Total Student(s): 1
BRAC University Student(s): 1
Other Institution Student(s): 0
=====
Name: Harry Potter
Department: Defence Against Dark Arts
Institution: Hogwarts School
-----
Total Student(s): 2
BRAC University Student(s): 1
Other Institution Student(s): 1
=====
Name: Levi Ackerman
Department: CSE
Institution: BRAC University
-----
Total Student(s): 3
BRAC University Student(s): 2
Other Institution Student(s): 1
```

Task 6

Write the **SultansDine** class so that the given code provides the expected output.

[You are not allowed to change the code below]

<pre># Write your code here SultansDine.details() print('#####') dhanmondi = SultansDine('Dhanmondi') dhanmondi.sellQuantity(25) dhanmondi.branchInformation() print('-----') SultansDine.details() print('=====') baily_road = SultansDine('Baily Road') baily_road.sellQuantity(15) baily_road.branchInformation() print('-----') SultansDine.details() print('=====') gulshan = SultansDine('Gulshan') gulshan.sellQuantity(9) gulshan.branchInformation() print('-----') SultansDine.details()</pre>	<p>Output:</p> <p>Total Number of branch(s): 0 Total Sell: 0 Taka ##### Branch Name: Dhanmondi Branch Sell: 10000 Taka ----- Total Number of branch(s): 1 Total Sell: 10000 Taka Branch Name: Dhanmondi, Branch Sell: 10000 Taka Branch consists of total sell's: 100.00% =====</p> <p>Branch Name: Baily Road Branch Sell: 5250 Taka ----- Total Number of branch(s): 2 Total Sell: 15250 Taka Branch Name: Dhanmondi, Branch Sell: 10000 Taka Branch consists of total sell's: 65.57% Branch Name: Baily Road, Branch Sell: 5250 Taka Branch consists of total sell's: 34.43% =====</p> <p>Branch Name: Gulshan Branch Sell: 2700 Taka ----- Total Number of branch(s): 3 Total Sell: 17950 Taka Branch Name: Dhanmondi, Branch Sell: 10000 Taka Branch consists of total sell's: 55.71% Branch Name: Baily Road, Branch Sell: 5250 Taka Branch consists of total sell's: 29.25% Branch Name: Gulshan, Branch Sell: 2700 Taka Branch consists of total sell's: 15.04%</p>
---	--

Subtasks:

1. Create **SultansDine** class
2. Create 2 class variable and 1 class list
3. Create 1 class method
4. Calculation of branch sell is given below
 - a. If sellQuantity < 10:

- i. Branch_sell = quantity * 300
 - b. Else if sellQuantity < 20:
 - i. Branch_sell = quantity * 350
 - c. Else
 - i. Branch_sell = quantity * 400
5. Calculation of branch's sell percentage = (branch's sell / total sell) * 100

Task 7

1	<code>class A:</code>
2	<code>temp = 4</code>
3	<code>def __init__(self):</code>
4	<code>self.y = self.temp - 2</code>
5	<code>self.sum = self.temp + 1</code>
6	<code>A.temp -= 2</code>
7	<code>self.methodA(3, 4)</code>
8	<code>def methodA(self, m, n):</code>
9	<code>x = 0</code>
10	<code>self.y = self.y + m + (self.temp)</code>
11	<code>A.temp += 1</code>
12	<code>x = x + 1 + n</code>
13	<code>self.sum = self.sum + x + self.y</code>
14	<code>print(x, self.y, self.sum)</code>
15	
16	<code>class B:</code>
17	<code>x = 0</code>
18	<code>def __init__(self, b = None):</code>

19	<code>self.y, self.temp, self.sum = 5, -5, 2</code>
20	
21	<code>if b == None:</code>
22	<code>self.y = self.temp + 3</code>
23	<code>self.sum = 3 + self.temp + 2</code>
24	<code>self.temp -= 2</code>
25	<code>else:</code>
26	<code>self.sum = b.sum</code>
27	<code>B.x = b.x</code>
28	<code>b.methodB(2, 3)</code>
29	<code>def methodA(self, m, n):</code>
30	<code>x = 2</code>
31	<code>self.y = self.y + m + (self.temp)</code>
32	<code>self.temp += 1</code>
33	<code>x = x + 5 + n</code>
34	<code>self.sum = self.sum + x + self.y</code>
35	<code>print(x, self.y, self.sum)</code>
36	<code>def methodB(self, m, n):</code>
37	<code>y = 0</code>
38	<code>y = y + self.y</code>
39	<code>B.x = self.y + 2 + self.temp</code>
40	<code>self.methodA(self.x, y)</code>
41	<code>self.sum = self.x + y + self.sum</code>

42	<code>print(self.x, y, self.sum)</code>
----	---

```

a1 = A()
b1 = B()
b2 = B(b1)
b1.methodA(1, 2)
b2.methodB(3, 2)

```

Task 8

1	<code>class FinalT6A:</code>
2	<code>temp = 3</code>
4	<code>def __init__(self, x, p):</code>
5	<code>self.sum, self.y = 0, 2</code>
6	<code>FinalT6A.temp += 3</code>
7	<code>self.y = self.temp - p</code>
8	<code>self.sum = self.temp + x</code>
9	<code>print(x, self.y, self.sum)</code>
11	<code>def methodA(self):</code>
12	<code>x, y = 0, 0</code>
13	<code>y = y + self.y</code>
14	<code>x = self.y + 2 + self.temp</code>
15	<code>self.sum = x + y + self.methodB(self.temp, y)</code>
16	<code>print(x, y, self.sum)</code>

18	<code>def methodB(self, temp, n):</code>
19	<code> x = 0</code>
20	<code> FinalT6A.temp += 1</code>
21	<code> self.y = self.y + (FinalT6A.temp)</code>
22	<code> FinalT6A.temp -= 1</code>
23	<code> x = x + 2 + n</code>
24	<code> self.sum = self.sum + x + self.y</code>
25	<code> print(x, self.y, self.sum)</code>
26	<code> return self.sum</code>

<pre> q1 = FinalT6A(2,1) q1.methodA() q1.methodA() </pre>	x	y	sum

Task 9

1	<code>class msgClass:</code>
2	<code> def __init__(self):</code>
3	<code> self.content = 0</code>
4	
5	<code>class Quiz3:</code>
6	<code> x = 0</code>
7	<code> def __init__(self, k = None):</code>

8	<code>self.sum, self.y = 0, 0</code>
9	<code>if k is None:</code>
10	<code>self.sum = 5</code>
11	<code>Quiz3.x = 2</code>
12	<code>self.y = 2</code>
13	<code>else:</code>
14	<code>self.sum = self.sum + k</code>
15	<code>self.y = 3</code>
16	<code>Quiz3.x += 2</code>
17	<code>def methodA(self):</code>
18	<code>x = 1</code>
19	<code>y = 1</code>
20	<code>msg = [None]</code>
21	<code>myMsg = msgClass()</code>
22	<code>myMsg.content = Quiz3.x</code>
23	<code>msg[0] = myMsg</code>
24	<code>msg[0].content = self.y + myMsg.content</code>
25	<code>self.y = self.y + self.methodB(msg[0])</code>
26	<code>y = self.methodB(msg[0]) + self.y</code>
27	<code>x = y + self.methodB(msg, msg[0])</code>
28	<code>self.sum = x + y + msg[0].content</code>
29	<code>print(x, y, self.sum)</code>
30	<code>def methodB(self, *args):</code>
31	<code>if len(args) == 2:</code>

32	<code>mg2, mg1 = args</code>
33	<code>x = 2</code>
34	<code>self.y = self.y + mg2[0].content</code>
35	<code>mg2[0].content = self.y + mg1.content</code>
36	<code>x = x + 2 + mg1.content</code>
37	<code>self.sum = self.sum + x + self.y</code>
38	<code>mg1.content = self.sum - mg2[0].content</code>
39	<code>print(Quiz3.x, self.y, self.sum)</code>
40	<code>return self.sum</code>
41	
42	<code>elif len(args) == 1:</code>
43	<code>mg1, = args</code>
44	<code>x = 1</code>
45	<code>y = 2</code>
46	<code>y = self.sum + mg1.content</code>
47	<code>self.y = y + mg1.content</code>
48	<code>x = Quiz3.x + 5 + mg1.content</code>
49	<code>self.sum = self.sum + x + y</code>
50	<code>Quiz3.x = mg1.content + x + 3</code>
51	<code>print(x, y, self.sum)</code>
52	<code>return y</code>

<code>a1 = Quiz3()</code>	x	y	sum
<code>a2 = Quiz3(5)</code>			

<code>msg = msgClass()</code> <code>a1.methodA()</code> <code>a2.methodB(msg)</code>			