BRAC
UNIVERSITY

Inspiring Excellence

Course Title: Programming Language II
Course Code: CSE 111
Lab Assignment no: 6

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 **get_details()** 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.

```
#Write your code here for subtasks 1-6.

s1 = Student("Samin", "CSE", 21, 3.91)
s1.get_details()
print("-----------------------")
s2 = Student("Fahim", "ECE", 21, 3.85)
s2.get_details()
print("-----------------------")
s3 = Student("Tahura", "EEE", 22, 3.01)
s3.get_details()
print("-----------------------")
s4 = Student.from_String("Sumaiya-BBA-23-3.96")
s4.get_details()


# Write the answer of subtask 5 here

# Write the answer of subtask 6 here


#You are not allowed to change the code above
```

OUTPUT

```
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
```

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

Write a class called **Dates** with the required constructor and methods.

Subtask:

1. Create a **class** called Dates and create the required **constructor**
2. Create a **class method** called toDashDate() to replace the “/” from date to “-”.
3. Create getDate() **method** to access variables.
4. In the conditional statement it prints “Equal”. Explain why.

[You are not allowed to change the code below]

```
# Write your code here for subtasks 1-5

date1 = Dates("05-09-2020")
dateFromDB = "05/09/2020"
date2= Dates.toDashDate(dateFromDB)


if(date1.getDate() == date2.getDate()):
   print("Equal")
else:
   print("Unequal")
```

Task 3

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]

# 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)
```

Output:

```
Total Passenger: 0
=========================
Name: Jack
Bus Fare: 550 taka
=========================
Name: Carol
Bus Fare: 450 taka
=========================
Name: Mike
Bus Fare: 500 taka
=========================
Total Passenger: 3
```

Task 4

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

[You are not allowed to change the code below]

# Write your code here

```
print(“No. of Traveller =”, Travel.count)
print("=======================")
t1 = Travel("Dhaka","India")
print(t1.display_travel_info())
print("=======================")
t2 = Travel("Kuala Lampur","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)
```

Output

```
No. of Traveller = 0
=======================
Source: Dhaka
Destination:India
Flight Time:1:00
=======================
Source: Kuala Lampur
Destination:Dhaka
Flight Time:23:00
=======================
Source: Dhaka
Destination:Germany
Flight Time:15:00
=======================
Source: Malaysia
Destination:Canada
Flight Time:9:00
=======================
No of Traveller = 4
```

Task 5

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

[You are not allowed to change the code below]

# *Write your code here*	***Output***

apples = Fruit("Apple", 3);
pears = Fruit("Pear", 4);
print(apples.name, apples.count)
print(pears.name,pears.count)
print("Total number of fruits", Fruit.fruitCount)
Fruit.saySomethingGood()
Fruit.resetcount()
print("Total number of fruits", Fruit.fruitCount)
	Apple 3
	Pear 4
	Total number of fruits 7
	Fruits are good for health
	Total number of fruits 0

Task 6

Design **Cat** class for the following code to get the output as shown.

You have already solved this problem in assignment 4 using constructor overloading. Now, solve this again but this time DO NOT USE CONSTRUCTOR OVERLOADING.

Hint: You will have to use classmethods.

[You are not allowed to change the code below]

# Write your code here

```
print("Total number of cats:", Cat.Number_of_cats)
c1 = Cat.no_parameter()
c2 = Cat.first_parameter("Black")
c3 = Cat("Brown", "jumping")
c4 = Cat("Red", "purring")
c5 = Cat.second_parameter("playing")
print("======================")
c1.printCat()
c2.printCat()
c3.printCat()
c4.printCat()
c5.printCat()
c1.changeColor("Blue")
c3.changeColor("Purple")
c1.printCat()
c3.printCat()
print("======================")
print("Total number of cats:", Cat.Number_of_cats)
```

Output:

```
Total number of cats: 0
======================
White cat is sitting
Black cat is sitting
Brown cat is jumping
Red cat is purring
Grey cat is playing
Blue cat is sitting
Purple cat is jumping
======================
Total number of cats: 5
```

Task 7

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

1. You will have to create a Cylinder class.
2. You will have to create 2 class variables.
3. Create a required constructor.
4. Write 2 *class methods:*
    - One that takes the height first and then the radius and then swaps
    - One that takes a string where the radius and height values are separated with a hyphen.

   Write 2 *static methods:*
    - One that calculates the area of a whole cylinder *(formula: $2\pi r^2 + 2\pi rh$)*
    - Another that calculates the volume of a cylinder *(formula: $\pi r^2 h$)*

***Observe the output values carefully to understand how the radius and height values are changing.*

[You are not allowed to change the code below]

# Write your code here

```
c1 = Cylinder(0,0)
Cylinder.area(c1.radius,c1.height)
Cylinder.volume(c1.radius,c1.height)
print("================================")
c2 = Cylinder.swap(8,3)
c2.area(c2.radius,c2.height)
c2.volume(c2.radius,c2.height)
print("================================")
c3 = Cylinder.changeFormat("7-13")
c3.area(c3.radius,c3.height)
c3.volume(c3.radius,c3.height)
print("================================")
Cylinder(0.3,5.56).area(Cylinder.radius,Cylinder.height)
print("================================")
Cylinder(3,5).volume(Cylinder.radius,Cylinder.height))
```

Output:

```
Default radius=5 and height=18.
Updated: radius=0 and height=0.
Area:  0.0
Volume:  0.0
================================
Default radius=0 and height=0.
Updated: radius=3 and height=8.
Area:  207.34511513692635
Volume:  226.1946710584651
================================
Default radius=3 and height=8.
Updated: radius=7.0 and height=13.0.
Area:  879.645943005142
Volume:  2001.1945203366981
================================
Default radius=7.0 and height=13.0.
Updated: radius=0.3 and height=5.56.
Area:  11.045839770021713
================================
Default radius=0.3 and height=5.56.
Updated: radius=3 and height=5.
Volume:  141.3716694115407
```