BRAC UNIVERSITY

Department of Computer Science and Engineering

Examination: Midterm
Duration: 60 Minutes
No. of Questions: 3

CSE 111: Programming Language II

Semester: Summer 2022
Full Marks: 20
No. of Pages: 2

Name: (Please write in CAPITAL LETTERS)	ID:	Section:

A

- ✓ Use the back **part** of the answer script for rough work. **No washroom breaks.**
- ✓ At the end of the exam, put the question **paper** inside the answer script and **return both**.

Question 1: CO2, CO4 [4 Points]

Design the **Exam** class with necessary properties so that the given output is produced:

#Write your code here

```
e1 = Exam('Midterm', 2, 10)
print(e1.detail())
print("===========================")
e2 = Exam('Final', 3, 10)
print(e2.detail())
```

Output:

```
Exam Type: Midterm
Number of questions: 2
Marks per questions: 10
Total Marks: 20
============================
Exam Type: Final
Number of questions: 3
Marks per questions: 10
Total Marks: 30
```

Question 2: CO4 [6 Points]

```
class MidA:
    def __init__(self):
        self.x = 3
        self.y = 7
        self.sum = 0
    def methodA(self, x):
        self.y = x + self.sum + self.x
        self.sum = x + self.y
        z = MidA()
        z.sum = self.sum + self.y
        self.methodB(z)
        print(self.x, self.y, self.sum)
    def methodB(self, a):
        y = 3
        a.x = self.x + self.sum;
        self.sum = a.x + a.y + y
        print(a.x, a.y, a.sum)
```

Illustrate the output of the following statements:

```
a = MidA()
a.methodA(5)
```

Output [Answer on question paper]		

Question 3: CO2, CO4 [10 Points]

Suppose you are making a program for a Television remote control named “RickMote”. The TV channel provider has **provided only 6 channels** where the corresponding channel numbers are **0,2,3,6,7,9**. This is a vital information as you might need to store the list of channel numbers in an instance variable. Now your task is to **design** the “**RickMote**” class in such a way that the expected output is produced for the given code below: [Hint:

- The channel numbers are not circular. So, channel number 0 is the first channel while 9 is the last. You cannot go below channel number 0 and beyond channel number 9.
- If power is turned off, there is no point in changing channel and volume.
- Increasing channel number means going to the next channel in the channel list and decreasing it means going to the previous channel in the channel list.]

```
#Write your code here
oTV = RickMote()
oTV.power()
print("1.############################")
oTV.showInfo()
print("2.############################")
oTV.changeChannel()
oTV.changeVolumeLevel()
oTV.showInfo()
print("3.############################")
oTV.power()
oTV.showInfo()
print("4.############################")
oTV.power()
oTV.changeVolumeLevel(4)
oTV.changeChannel(3)
oTV.showInfo()
print("5.############################")
oTV.changeVolumeLevel(-2)
oTV.showInfo()
print("6.############################")
oTV.power()
oTV.changeChannel(9)
oTV.changeVolumeLevel(-1)
oTV.showInfo()
print("7.############################")
oTV.power()
oTV.changeChannel(11)
oTV.showInfo()
```

Output:

```
1.############################
ID Cable Box Status:
Cable Box is: ON
Channel:0
Volume:3
2.############################
ID Cable Box Status:
Cable Box is: ON
Channel:2
Volume:4
3.############################
ID Cable Box Status:
Cable Box is: OFF
4.############################
ID Cable Box Status:
Cable Box is: ON
Channel:3
Volume:8
5.############################
ID Cable Box Status:
Cable Box is: ON
Channel:3
Volume:6
6.############################
Power is turned off. Cannot change channel.
Power is turned off. Cannot change volume.
ID Cable Box Status:
Cable Box is: OFF
7.############################
TV channel does not exist.
ID Cable Box Status:
Cable Box is: ON
Channel:3
Volume:6
```

BRAC UNIVERSITY

Department of Computer Science and Engineering

Examination: Midterm
Duration: 60 Minutes
No. of Questions: 3

CSE 111: Programming Language II

Semester: Summer 2022
Full Marks: 20
No. of Pages: 2

Name: (Please write in CAPITAL LETTERS)	ID:	Section:

B

✓ Use the back **part** of the answer script for rough work. **No washroom breaks.**

✓ At the end of the exam, put the question **paper** inside the answer script and **return both**.

Question 1: CO2, CO4 [4 Points]

Design the **Exam** class with necessary properties so that the given output is produced:

#Write your code here

```
e1 = Exam('Midterm', 2, 10)
print(e1.detail())
print("===========================")
e2 = Exam('Final', 3, 10)
print(e2.detail())
```

Output:

```
Exam Type: Midterm
Number of questions: 2
Marks per questions: 10
Total Marks: 20
===========================
Exam Type: Final
Number of questions: 3
Marks per questions: 10
Total Marks: 30
```

Question 2: CO4 [6 Points]

```
class MidA:
    def __init__(self):
        self.x = 5
        self.y = 3
        self.sum = 0
    def methodA(self, x):
        self.y = x + self.sum + self.x
        self.sum = x + self.y
        z = MidA()
        z.sum = self.sum + self.y
        self.methodB(z)
        print(self.x, self.y, self.sum)
    def methodB(self, a):
        y = 4
        a.x = self.x + self.sum;
        self.sum = a.x + a.y + y
        print(a.x, a.y, a.sum)
```

Illustrate the output of the following statements:

```
a = MidA()
a.methodA(6)
```

Output
[Answer on question paper]

Question 3: CO2, CO4 [10 Points]

Suppose you are making a program for a Television remote control named “RickMote”. The TV channel provider has **provided only 6 channels** where the corresponding channel numbers are **0,2,3,6,7,9**. This is a vital information as you might need to store the list of channel numbers in an instance variable. Now your task is to **design** the “**RickMote**” class in such a way that the expected output is produced for the given code below: [Hint:

- The channel numbers are not circular. So, channel number 0 is the first channel while 9 is the last. You cannot go below channel number 0 and beyond channel number 9.
- If power is turned off, there is no point in changing channel and volume.
- Increasing channel number means going to the next channel in the channel list and decreasing it means going to the previous channel in the channel list.]

```
#Write your code here
oTV = RickMote()
oTV.power()
print("1.############################")
oTV.showInfo()
print("2.############################")
oTV.changeChannel()
oTV.changeVolumeLevel()
oTV.showInfo()
print("3.############################")
oTV.power()
oTV.showInfo()
print("4.############################")
oTV.power()
oTV.changeVolumeLevel(4)
oTV.changeChannel(3)
oTV.showInfo()
print("5.############################")
oTV.changeVolumeLevel(-2)
oTV.showInfo()
print("6.############################")
oTV.power()
oTV.changeChannel(9)
oTV.changeVolumeLevel(-1)
oTV.showInfo()
print("7.############################")
oTV.power()
oTV.changeChannel(11)
oTV.showInfo()
```

Output:

```
1.############################
ID Cable Box Status:
Cable Box is: ON
Channel:0
Volume:3
2.############################
ID Cable Box Status:
Cable Box is: ON
Channel:2
Volume:4
3.############################
ID Cable Box Status:
Cable Box is: OFF
4.############################
ID Cable Box Status:
Cable Box is: ON
Channel:3
Volume:8
5.############################
ID Cable Box Status:
Cable Box is: ON
Channel:3
Volume:6
6.############################
Power is turned off. Cannot change channel.
Power is turned off. Cannot change volume.
ID Cable Box Status:
Cable Box is: OFF
7.############################
TV channel does not exist.
ID Cable Box Status:
Cable Box is: ON
Channel:3
Volume:6
```

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Midterm
Duration: 60 Minutes
No. of Questions: 3

CSE 111: Programming Language II

Semester: Fall 2022
Full Marks: 20
No. of Pages: 2

A

Name: (Please write in CAPITAL LETTERS)	ID:	Section:

✔ Use the back **part** of the answer script for rough work. **No washroom breaks.**
✔ At the end of the exam, put the question **paper** inside the answer script and **return both**.

Question 1: CO2 [4 Points]

Design the "**Spotify**" class with required properties to produce the given output for the provided driver code.

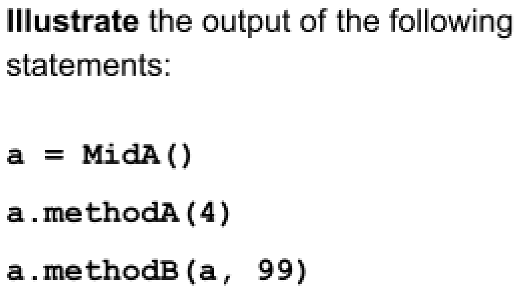
```
#Write your code here
rock= Spotify("Heavy Metal", "This Fire Burns", "Here I Am")
print(rock.show_playlist())
```

Output:

```
Genre: Heavy Metal
Song list: This Fire Burns, Here I Am
```

Question 2: CO4 [6 Points]

```
class MidA:
  def __init__(self):
      self.x = 4
      self.y = 1
      self.sum = 3
  def methodA(self, x):
self.y = self.sum + self.x + x
      self.sum = x + self.y
      d = MidA()
```



Illustrate the output of the following statements:

```
a = MidA()
a.methodA(4)
a.methodB(a, 99)
```

```
d.sum = self.sum + self.methodB(d)
      print(self.x, self.y, self.sum)
  def methodB(self, t, z=0):
      y = 3
      t.x = self.x + self.sum
      self.sum = t.x + t.y + y
      print(t.x, t.y, t.sum)
      if z == 0:
          return t.sum
```

Question 3: CO2, CO4 [10 Points]

Implement the **“PizzaMachine”** class with necessary properties, so that the given output is produced for the provided driver code.

[Hint:

1. The Pizza machine can only bake pizzas of 3 spice levels: Regular, Hot and Super Naga.
2. Pizza cannot be baked if no toppings are specified.
3. "type(variable)" function returns the data type of the variable passed to it as an argument. You may need it.]

#Write your code here

```
pizza1 = PizzaMachine()
order1 = pizza1.customizePizza(["Cheese", "Pepperoni"],
"Hot")
print("1################################## ")
print(order1)
print("2================================")
pizza2 = PizzaMachine("Vege")
order2 = pizza2.customizePizza("Super Naga")
print("3##################################")
print(order2)
print("4================================")
pizza3 = PizzaMachine("Chicken Blast",12)
order3 = pizza3.customizePizza(["Mushroom"])
print("5##################################")
print(order3)
print("6================================")
pizza4 = PizzaMachine("Beef Bonanza",16)
order4 = pizza4.customizePizza(["Cheese","Beef kala
bhuna"],"Mild")
print("7##################################")
print(order4)
print("8================================")
```

Output:

```
1##################################
Your 6-inch Hot spicy Regular Pizza is ready with
Cheese,Pepperoni toppings. Enjoy!
2================================
3##################################
No toppings specified! Can't bake pizza.
4================================
5##################################
Your 12-inch Regular spicy Chicken Blast Pizza is ready
with Mushroom toppings. Enjoy!
6================================
7##################################
Sorry! Spice level not allowed. Can't bake pizza.
8================================
```

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Midterm
Duration: 60 Minutes
No. of Questions: 3

CSE 111: Programming Language II

Semester: Fall 2022
Full Marks: 20
No. of Pages: 2

B

Name: (Please write in CAPITAL LETTERS)	ID:	Section:

✔ Use the back **part** of the answer script for rough work. **No washroom breaks.**
✔ At the end of the exam, put the question **paper** inside the answer script and **return both**.

Question 1: CO2 [4 Points]

Design the "**Spotify**" class with required properties to produce the given output for the provided driver code.

#Write your code here

```
rock= Spotify("Heavy Metal", "This Fire Burns", "Here I Am")
print(rock.show_playlist())
```

Output:

```
Genre: Heavy Metal
Song list: This Fire Burns, Here I Am
```

Question 2: CO4 [6 Points]

```
class MidB:
  def __init__(self):
      self.x = 3
      self.y = 0
      self.sum = 5
  def methodA(self, x):

self.y = self.sum + self.x + x
      self.sum = x + self.y
      d = MidB()
```

Illustrate the output of the following statements:

```
b = MidB()
b.methodA(6)
b.methodB(b, 99)
```

```

d.sum = self.sum + self.methodB(d)
        print(self.x, self.y, self.sum)
    def methodB(self, t, z=0):
        y = 3
        t.x = self.x + self.sum
        self.sum = t.x + t.y + y
        print(t.x, t.y, t.sum)
        if z == 0:
            return t.sum
```

Question 3: CO2, CO4 [10 Points]

Implement the **“PizzaMachine”** class with necessary properties, so that the given output is produced for the provided driver code.

[Hint:

1. The Pizza machine can only bake pizzas of 3 spice levels: Regular, Hot and Super Naga.
2. Pizza cannot be baked if no toppings are specified.
3. "type(variable)" function returns the data type of the variable passed to it as an argument. You may need it.]

```
#Write your code here

pizza1 = PizzaMachine()
order1 = pizza1.customizePizza(["Cheese", "Pepperoni"],
"Hot")
print("1################################# ")
print(order1)
print("2============================")
pizza2 = PizzaMachine("Vege")
order2 = pizza2.customizePizza("Super Naga")
print("3#################################")
print(order2)
print("4============================")
pizza3 = PizzaMachine("Chicken Blast",12)
order3 = pizza3.customizePizza(["Mushroom"])
print("5#################################")
print(order3)
print("6============================")
pizza4 = PizzaMachine("Beef Bonanza",16)
order4 = pizza4.customizePizza(["Cheese","Beef kala
bhuna"],"Mild")
print("7#################################")
print(order4)
print("8============================")
```

Output:

```
1#################################
Your 6-inch Hot spicy Regular Pizza is ready with
Cheese,Pepperoni toppings. Enjoy!
2============================
3#################################
No toppings specified! Can't bake pizza.
4============================
5#################################
Your 12-inch Regular spicy Chicken Blast Pizza is ready
with Mushroom toppings. Enjoy!
6============================
7#################################
Sorry! Spice level not allowed. Can't bake pizza.
8============================
```

BRAC UNIVERSITY

Department of Computer Science and Engineering

Examination: Midterm
Duration: 60 Minutes
No. of Questions: 3

CSE 111: Programming Language II

Semester: Spring 2023
Full Marks: 20
No. of Pages: 2

Name: (Please write in CAPITAL LETTERS)	ID:	Section:

A

- ✓ Use the back **part** of the answer script for rough work. **No washroom breaks.**
- ✓ At the end of the exam, put the question **paper** inside the answer script and **return both**.

Question 1: CO2 [4 Points]

Design the “**CoffeeMachine**” class with required properties to produce the given output for the provided driver code.

```
#Write your code here

cm1 = CoffeeMachine("Miyako")
cm1.insertIngredients("Coffee beans", "Milk", "Sugar")
print(cm1.getDetails())
```

Output:

```
Brand Name: Miyako
Ingredients: Coffee beans, Milk, Sugar
```

Question 2: CO4 [6 Points]

```
class MidA:
    def __init__(self):
        self.y,self.z,self.sum = 2,3,-1
    def m1(self, mg2, mg1=2):
        x = 0
        self.y = self.y + mg2[0]
        x += 33 + mg1
        self.sum += x + self.y
        mg2[0] = self.y + mg1
        mg1 = mg1 + x + 2
        print(x, self.y, self.sum)
    def m2(self, y=3):
        mid = [0]
        mid[0] = 7
        self.m1(mid, mid[0])
        z = y + mid[0]
        y = self.y + mid[0]
        self.sum = z + y + mid[0]
        print(z, y, self.sum)
```

Illustrate the output of the following statements:

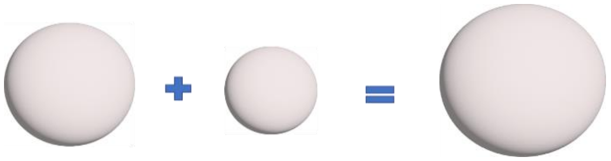
```
a = MidA()
a.m1([6])
a.m2()
```

Output [Answer on question paper]		
		97
25		84

Question 3: CO2, CO4 [10 Points]

Design the **Sphere** class such that the following output is produced. **Hints:**

1. Volume of the sphere = $\frac{4}{3} * \pi * r^3$, where r = radius of the sphere and π = 3.1416.
2. Merging spheres together conserves the total volume. The volume of the bigger sphere can be calculated by adding the volume of the spheres being merged. [see pictures for details]. Pay attention to how the object is updated.
3. When spheres of different colors are merged together then the merged sphere will have '**Mixed Color**' instead of one particular color.
4. Your code should work for any number of Sphere objects passed to the **merge_sphere()** method.
5. You do not need to worry about how many digits should be printed after the decimal point.
6. The default value of the radius r is 1.



r1 = 6 **r2 = 4** **v3 = v1+v2=1172.864**

v1 = 904.7808 **v2 = 268.0832**

#Write your code here

```
sphere1 = Sphere("Sphere 1")
print("1***************")
sphere1.printDetails()
print("2***************")
sphere2 = Sphere("Sphere 2", 3)
print("3***************")
sphere2.printDetails()
print("4***************")
sphere3 = Sphere("Sphere 3", 2)
print("5***************")
sphere3.printDetails()
print("6***************")
sphere3.merge_sphere(sphere1,sphere2)
print("7***************")
sphere3.printDetails()
print("8***************")
sphere4 = Sphere("Sphere 4", 5, "Purple")
print("9***************")
sphere4.merge_sphere(sphere3)
print("10***************")
sphere4.printDetails()
```

Output:

```
1***************
Sphere ID: Sphere 1
Color: White
Volume: 4.1888
2***************
3***************
Sphere ID: Sphere 2
Color: White
Volume: 113.09759999999999
4***************
5***************
Sphere ID: Sphere 3
Color: White
Volume: 33.5104
6***************
Spheres are being merged
7***************
Sphere ID: Sphere 3
Color: White
Volume: 150.7968
8***************
9***************
Spheres are being merged
10***************
Sphere ID: Sphere 4
Color: Mixed Color
Volume: 674.3967999999999
```

BRAC UNIVERSITY
Department of Computer Science and Engineering

Examination: Midterm
Duration: 60 Minutes
No. of Questions: 3

CSE 111: Programming Language II

Semester: Spring 2023
Full Marks: 20
No. of Pages: 2

Name: (Please write in CAPITAL LETTERS)	ID:	Section:

B

✓ Use the back **part** of the answer script for rough work. **No washroom breaks.**

✓ At the end of the exam, put the question **paper** inside the answer script and **return both**.

Question 1: CO2 [4 Points]

Design the "**CoffeeMachine**" class with required properties to produce the given output for the provided driver code.

```
#Write your code here

cm1 = CoffeeMachine("Miyako")
cm1.insertIngredients("Coffee beans", "Milk", "Sugar")
print(cm1.getDetails())
```

```
Output:

Brand Name: Miyako
Ingredients: Coffee beans, Milk, Sugar
```

Question 2: CO4 [6 Points]

```
class MidB:
    def __init__(self):
        self.y,self.z,self.sum = 3,2,-1
    def m1(self, mg2, mg1=2):
        x = 0
        self.y = self.y + mg2[0]
        x += 23 + mg1
        self.sum += x + self.y
        mg2[0] = self.y + mg1
        mg1 = mg1 + x + 4
        print(x, self.y, self.sum)
    def m2(self, y=3):
        mid = [0]
        mid[0] = 5
        self.m1(mid, mid[0])
        z = y + mid[0]
        y = self.y + mid[0]
        self.sum = z + y + mid[0]
        print(z, y, self.sum)
```

Illustrate the output of the following statements:

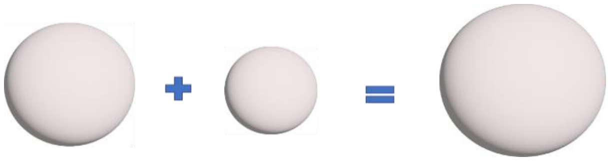
```
a = MidB()
a.m1([8])
a.m2()
```

Output [Answer on question paper]		
		79
24		82

Question 3: CO2, CO4 [10 Points]

Design the **Sphere** class such that the following output is produced. **Hints:**

1. Volume of the sphere = $\frac{4}{3} * \pi * r^3$, where r = radius of the sphere and π = 3.1416.
2. Merging spheres together conserves the total volume. The volume of the bigger sphere can be calculated by adding the volume of the spheres being merged. [see pictures for details]. Pay attention to how the object is updated.
3. When spheres of different colors are merged together then the merged sphere will have '**Mixed Color**' instead of one particular color.
4. Your code should work for any number of Sphere objects passed to the **merge_sphere()** method.
5. You do not need to worry about how many digits should be printed after the decimal point.
6. The default value of the radius r is 1.



r1 = 6 **r2 = 4** **v3 = v1+v2=1172.864**

v1 = 904.7808 **v2 = 268.0832**

#Write your code here

```
sphere1 = Sphere("Sphere 1")
print("1***************")
sphere1.printDetails()
print("2***************")
sphere2 = Sphere("Sphere 2", 3)
print("3***************")
sphere2.printDetails()
print("4***************")
sphere3 = Sphere("Sphere 3", 2)
print("5***************")
sphere3.printDetails()
print("6***************")
sphere3.merge_sphere(sphere1,sphere2)
print("7***************")
sphere3.printDetails()
print("8***************")
sphere4 = Sphere("Sphere 4", 5, "Purple")
print("9***************")
sphere4.merge_sphere(sphere3)
print("10***************")
sphere4.printDetails()
```

Output:

```
1***************
Sphere ID: Sphere 1
Color: White
Volume: 4.1888
2***************
3***************
Sphere ID: Sphere 2
Color: White
Volume: 113.09759999999999
4***************
5***************
Sphere ID: Sphere 3
Color: White
Volume: 33.5104
6***************
Spheres are being merged
7***************
Sphere ID: Sphere 3
Color: White
Volume: 150.7968
8***************
9***************
Spheres are being merged
10***************
Sphere ID: Sphere 4
Color: Mixed Color
Volume: 674.3967999999999
```