BRAC UNIVERSITY

Department of Computer Science and Engineering

Examination: Final
Duration: 90 Minutes
No. of Questions: 3

CSE 111: Programming Language II

Semester: Summer 2022
Full Marks: 30
No. of Pages: 3

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: CO4 [10 Points]

Design the **Monster** class with necessary properties so that the given output is produced.
[**Hint:**

1. **Let's say there are 2 monsters: A and B.**
   -A can attack B if A is alive.
   -If A is alive, it can only attack B if B is alive.
   -If A's power is greater than that of B, only then A can defeat B; Otherwise, A will be defeated by B.
   - Once a monster gets defeated, it is considered to be dead.
2. **The variable "monsterCount" keeps track of the number of monsters alive. So, update it accordingly.]**

```
#Write your code here

monster1 = Monster('Godzilla', 40)
monster2 = Monster('Hydra', 30)
monster3 = Monster('KingKong', 50)
print(f"Number of monsters
alive:{Monster.monsterCount}")
print('1--------------------------------')
print(monster1.get_details())
print('2--------------------------------')
print(monster2.get_details())
print('3--------------------------------')
print(monster3.get_details())
print('4--------------------------------')
monster1.attack()
print('5--------------------------------')
monster1.attack(monster2)
print('6--------------------------------')
monster1.attack(monster2, monster3)
print('7--------------------------------')
print(f"Number of monsters
alive:{Monster.monsterCount}")
print('8--------------------------------')
print(monster2.get_details())
print('9--------------------------------')
monster2.attack(monster1)
```

Output:

```
Number of monsters alive:3
1--------------------------------
Name:Godzilla
Power:40
Alive:True
2--------------------------------
Name:Hydra
Power:30
Alive:True
3--------------------------------
Name:KingKong
Power:50
Alive:True
4--------------------------------
No monsters to attack
5--------------------------------
Attack successful.Godzilla defeated
Hydra.
6--------------------------------
Cannot attack Hydra. It's not alive.
Attack unsuccessful. Godzilla was
defeated by KingKong.
7--------------------------------
Number of monsters alive:1
8--------------------------------
Name:Hydra
Power:30
Alive:False
9--------------------------------
Hydra is not alive to attack.
```

Question 2: CO5 [10 Points]

Implement the **BracbookUser** class that is derived from the **User** class with necessary properties so that the given output is produced.

[**You are not allowed to change the given code under any circumstances.]**

```
class User:
  activities = ["Post", "Like", "Comment"]
  def __init__(self, name,email):
    self.name = name
    self.email = email
  def UserActivity(self, activityType):
    if activityType in User.activities:
      return True
    else:
      return False
  def userDetail(self):
    return f"User Detail:\nName:{self.name}
               \nEmail: {self.email}"

#Write your code here
user1 = BracbookUser("Rakait","xyz@gmail.com")
print("1===========================")
print(user1.userDetail())
print("2===========================")
user2 = BracbookUser("Sazzad","abc@gmail.com",
"01727xxxxxx")
print("3===========================")
print(user2.userDetail())
print("4===========================")
user1.UserActivity("Like")
print("5===========================")
user1.UserActivity("Comment")
print("6===========================")
print(user1.userDetail())
print("7===========================")
user2.UserActivity("Share")
print("8===========================")
user2.UserActivity("Comment")
print("9===========================")
print(user2.userDetail())
```

Output:

```
1===========================
User Detail:
Name: Rakait
Email: xyz@gmail.com
Phone: Not set
Activity Log: No recent activity.
2===========================
3===========================
User Detail:
Name: Sazzad
Email: abc@gmail.com
Phone: 01727xxxxxx
Activity Log: No recent activity.
4===========================
Rakait has Like(d/ed) successfully.
5===========================
Rakait has Comment(d/ed) successfully.
6===========================
User Detail:
Name: Rakait
Email: xyz@gmail.com
Phone: Not set
Activity Log: Like,Comment
7===========================
No activities found like Share
8===========================
Sazzad has Comment(d/ed) successfully.
9===========================
User Detail:
Name: Sazzad
Email: abc@gmail.com
Phone: 01727xxxxxx
Activity Log: Comment
```

Question – 3: CO4 [10 Points]

```
class A:
    temp = 5
    def __init__(self):
        self.y = A.temp - 2
        self.sum = A.temp + 1
        A.temp += 3
    def methodA(self, m,  n):
        x = 0
        self.y = self.y + m + (A.temp)
        x = x + 2 + n
        print(x, self.y, self.sum)
        self.methodB(-2, 6)
        self.sum = self.sum + x + A.temp
        self.methodB(-4, self.sum, 3)
    def methodB(self, m,  n):
        y = 5
        y = y + self.y
        self.sum = B.x + y + n
        print(B.x, y, self.sum)
class B(A):
    x = 1
    def __init__(self, obj=None):
        super().__init__()
        if obj != None:
            obj.sum = 11
        self.y = A.temp + 4
        self.sum = 3 + A.temp + B.x
    def methodB(self, m, n, y=0):
        y = y + self.y + n
        B.x = m + self.y + n
        self.sum = B.x + y + A.temp
        print(B.x, y, self.sum)
```

Illustrate the output of the following statements:

```
b1 = B()
b2 = B(b1)
b1.methodA(-3, -7)
```

Output:

X	y	sum