Course Title: Programming Language II
Course Code: CSE 111
Lab Assignment no: 3

Task 1

Write a class that for running the following codes:
[You are not allowed to change the code below]

```
#Write your class code here
data_type1 = DataType(‘Integer’, 1234)
print(data_type1.name)
print(data_type1.value)
print('=====================')
data_type2 = DataType(‘String’, ‘Hello’)
print(data_type2.name)
print(data_type2.value)
print('=====================')
data_type3 = DataType(‘Float’, 4.0)
print(data_type3.name)
print(data_type3.value)
```

Output:

```
Integer
1234
=====================
String
Hello
=====================
Float
4.0
```

Subtasks:

1. Create a class named **DataType** with the required constructor.
2. Assign name and values in constructor according to the output.

Task 2

Design a class Joker with parameterized constructor so that the following line of code prints the result shown in the output box.

[You are not allowed to change the code below]

```
#Write your class code here

j1 = Joker('Heath Ledger', 'Mind Game', False)
print(j1.name)
print(j1.power)
print(j1.is_he_psycho)
print(“====================”)
j2 = Joker('Joaquin Phoenix', 'Laughing out Loud', True)
print(j2.name)
print(j2.power)
print(j2.is_he_psycho)
print(“====================”)
if j1 == j2:
    print('same')
else:
    print('different')
j2.name = 'Heath Ledger'
if j1.name == j2.name:
    print('same')
else:
    print('different')
#Write your code for 2,3 here
```

Output:

```
Heath Ledger
Mind Game
```

```
False
=====================
Joaquin Phoenix
Laughing out Loud
True
=====================
different
same
```

Subtask:
1) Design the class using a parameterized constructor.
2) The first if/else block prints the output as ‘different’, but why? Explain your answer and print your explanation at the very end.
3) The second if/else block prints the output as ‘same’, but why? Explain your answer and print your explanation at the very end.

Task 3

Design a class called **Pokemon** using a parameterized constructor so that after executing the following line of code the desired result shown in the output box will be printed. First object along with print has been done for you, you also need to create other objects and print accordingly to get the output correctly.

[You are not allowed to change the code below]

```
#Write your code for class here

team_pika = Pokemon('pikachu', 'charmander', 90, 60, 10)
print('=======Team 1=======')
print('Pokemon 1:',team_pika.pokemon1_name,
team_pika.pokemon1_power)
print('Pokemon 2:',team_pika.pokemon2_name,
team_pika.pokemon2_power)
pika_combined_power = (team_pika.pokemon1_power +
team_pika.pokemon2_power) * team_pika.damage_rate
print('Combined Power:', pika_combined_power)
#Write your code for subtask 2,3,4 here
```

Output:

```
=======Team 1=======
Pokemon 1: pikachu 90
Pokemon 2: charmander 60
Combined Power: 1500
=======Team 2=======
Pokemon 1: bulbasaur 80
Pokemon 2: squirtle 70
Combined Power: 1350
```

Subtask:

1) Design the Pokemon class using a parameterized constructor. The 5 values that are being passed through the constructor are pokemon 1 name, pokemon 2 name, pokemon 1 power, pokemon 2 power, damage rate respectively.

After designing the class, if you run the above code the details in Team 1 will be printed.
2) Create an object named team_bulb and pass the value ‘bulbasaur’, ‘squirtle’, 80, 70, 9 respectively.
3) Use print statements accordingly to print the desired result of Team 2.

Note: Power is always being calculated using the instance variables. Example:
```
(team_pika.pokemon1_power + team_pika.pokemon2_power) *
team_pika.damage_rate
```

Task 4

Design the **Country** class so that the code gives the expected output.
[You are not allowed to change the code below]

```
# Write your Class Code here
country = Country()
print('Name:',country.name)
print('Continent:',country.continent)
print('Capital:',country.capital)
print('Fifa Ranking:',country.fifa_ranking)
print('===================')
country.name = “Belgium”
country.continent = “Europe”
country.capital = “Brussels”
country.fifa_ranking = 1
print('Name:',country.name)
print('Continent:',country.continent)
print('Capital:',country.capital)
print('Fifa Ranking:',country.fifa_ranking)
```

Output:

```
Name: Bangladesh
Continent: Asia
Capital: Dhaka
Fifa Ranking: 187
===================
Name: Belgium
Continent: Europe
```

Capital: Brussels
Fifa Ranking: 1

Task 5

Write the **DemonSlayer** class so that the code gives the expected output.
[You are not allowed to change the code below]

```
# Write your Class Code here
tanjiro = DemonSlayer("Tanjiro", "Water Breathing", 10, 10)
print('Name:',tanjiro.name)
print('Fighting Style:',tanjiro.style)
print(f'Knows {tanjiro.number_of_technique} technique(s) and has
killed {tanjiro.kill} demon(s)')
print('===================')
zenitsu = DemonSlayer("Zenitsu", "Thunder Breathing", 1, 4)
print('Name:',zenitsu.name)
print('Fighting Style:',zenitsu.style)
print(f'Knows {zenitsu.number_of_technique} technique(s) and has
killed {zenitsu.kill} demon(s)')
print('===================')
inosuke = DemonSlayer("Inosuke", "Beast Breathing", 5, 7)
print('Name:',inosuke.name)
print('Fighting Style:',inosuke.style)
print(f'Knows {inosuke.number_of_technique} technique(s) and has
killed {inosuke.kill} demon(s)')
print('===================')
print(f'{tanjiro.name}, {zenitsu.name}, {inosuke.name} knows
total {tanjiro.number_of_technique + zenitsu.number_of_technique
+ inosuke.number_of_technique} techniques')
print(f'They have killed total {tanjiro.kill + zenitsu.kill +
inosuke.kill} demons')
```

Output:

```
Name: Tanjiro
Fighting Style: Water Breathing
Knows 10 technique(s) and has killed 10 demon(s)
===================
Name: Zenitsu
Fighting Style: Thunder Breathing
```

```
Knows 1 technique(s) and has killed 4 demon(s)
===================
Name: Inosuke
Fighting Style: Beast Breathing
Knows 5 technique(s) and has killed 7 demon(s)
===================
Tanjiro, Zenitsu, Inosuke knows total 16 techniques
They have killed total 21 demons
```

Task 6

Write the **box** class so that the code gives the expected output.

#Write your class code here

```
print("Box 1")
b1 = box([10,10,10])
print("=========================")
print("Height:", b1.height)
print("Width:", b1.width)
print("Breadth:", b1.breadth)
print("-------------------------")
print("Box 2")
b2 = box((30,10,10))
print("=========================")
print("Height:", b2.height)
print("Width:", b2.width)
print("Breadth:", b2.breadth)
b2.height = 300
print("Updating Box 2!")
print("Height:", b2.height)
print("Width:", b2.width)
print("Breadth:", b2.breadth)
print("-------------------------")
print("Box 3")
b3 = b2
print("Height:", b3.height)
print("Width:", b3.width)
print("Breadth:", b3.breadth)
```

Output:

```
Box 1
Creating a Box!
Volume of the box is 1000 cubic units.
=========================
Height: 10
Width: 10
Breadth: 10
-------------------------
Box 2
Creating a Box!
Volume of the box is 3000 cubic units.
=========================
Height: 30
Width: 10
Breadth: 10
Updating Box 2!
Height: 300
Width: 10
Breadth: 10
-------------------------
Box 3
Height: 300
Width: 10
Breadth: 10
```

Task 7

Design the required class from the given code and the outputs.
[You are not allowed to change the code below]

Hint:
Number of the border characters for the top and the bottom
= 1
   + Number of spaces between the left side border and the first character of the button name
   + Length of the button name
   + Number of spaces between the right side border and the last character of the button name
   + 1

NOTE: Don't count the space or any character from the button representation to solve this problem.

#Write your class code here

```
word = "CANCEL"
spaces = 10
border = 'x'
b1 = buttons(word, spaces, border)
print("=======================================================")
b2 = buttons("Notify",3, '!')
print("=======================================================")
b3 = buttons('SAVE PROGRESS', 5, '$')
```

Output:

```
CANCEL Button Specifications:
Button name: CANCEL
Number of the border characters for the top and the bottom: 28
Number of spaces between the left side border and the first character of the button
name: 10
Number of spaces between the right side border and the last character of the button
name: 10
Characters representing the borders: x

xxxxxxxxxxxxxxxxxxxxxxxxxxxx
x          CANCEL          x
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
=======================================================
```

Notify Button Specifications:
Button name: Notify
Number of the border characters for the top and the bottom: 14
Number of spaces between the left side border and the first character of the button name: 3
Number of spaces between the right side border and the last character of the button name: 3
Characters representing the borders: !

```
!!!!!!!!!!!!!!
!   Notify   !
!!!!!!!!!!!!!!
===============================================================
```

SAVE PROGRESS Button Specifications:
Button name: SAVE PROGRESS
Number of the border characters for the top and the bottom: 25
Number of spaces between the left side border and the first character of the button name: 5
Number of spaces between the right side border and the last character of the button name: 5
Characters representing the borders: $

```
$$$$$$$$$$$$$$$$$$$$$$$$$
$     SAVE PROGRESS     $
$$$$$$$$$$$$$$$$$$$$$$$$$
```

Task 8

Suppose your little sibling wants your help to check his math homework. He is done with his homework but wants you to see if all his results are correct. Since the student with all correct results gets 3 stars. However, you want your brother to check this on his own. So, you design a calculator for him in python. You could have given your scientific calculator but you wanted to give him a basic calculator and also wanted to see if you can even design one.

Subtasks:

1. Create a class called Calculator.
2. Your class shall have 1 constructor and 4 methods, namely add, subtract, multiply and divide.
3. Now, create an object of your class. After creating an object, it should print “Let’s Calculate!”
4. Then take 3 inputs from the user: first value, operator, second value
5. Now based on the given operator, call the required method and print the result.

Sample Input:

```
1
+
2
```

Sample Output:

```
Let’s Calculate!
Value 1: 1
Operator: +
Value 2: 2
Result:  3
```

Task 9

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

[For BMI, the formula is BMI = weight/height^2, where weight is in kg and height in meters]

Driver Code	Output

# ***Write your code here*** 	
 	
p1 = Patient("A", 55, 63.0, 158.0)
p1.printDetails()
print("====================")
p2 = Patient("B", 53, 61.0, 149.0)
p2.printDetails()
	Name: A
	Age: 55
	Weight: 63.0 kg
	Height: 158.0 cm
	BMI: 25.236340330075304
	====================
	Name: B
	Age: 53
	Weight: 61.0 kg
	Height: 149.0 cm
	BMI: 27.476239809017613

Task 10

Design a “**Vehicle**” class. A vehicle assumes that the whole world is a 2-dimensional graph paper. It maintains its x and y coordinates (both are integers). Any new object created of the Vehicle class will always start at the coordinates (0,0).

It must have methods to move up, down, left, right and a print_position() method for printing the current coordinate.

Note: All moves are 1 step. That means a single call to any move method changes the value of either x or y or both by 1.

[You are not allowed to change the code below]

# ***Write your class here***	***OUTPUT***

```
car = Vehicle()
car.print_position()
car.moveUp()
car.print_position()
car.moveLeft()
car.print_position()
car.moveDown()
car.print_position()
car.moveRight()
```
	(0,0)
	(0,1)
	(-1,1)
	(-1,0)

Task 11

Design a class Shape for the given code below.

• Write a class Shape.

• Write the required constructor that takes 3 parameters and initialize the instance variables accordingly.

• Write a method area() that prints the area.

Hint: the area method can calculate only for the shapes: Triangle, Rectangle, Rhombus, and Square. So, you have to use conditions inside this method

For this task, assume that --

- for a triangle, the arguments passed are the base and height
- for a rhombus, the arguments passed are the diagonals
- for a square or rectangle, the arguments passed are the sides.

Driver Code

```
# Write your code here

triangle = Shape("Triangle",10,25)
triangle.area()
print("==========================")
square = Shape("Square",10,10)
square.area()
print("==========================")
rhombus = Shape("Rhombus",18,25)
rhombus.area()
print("==========================")
rectangle = Shape("Rectangle",15,30)
rectangle.area()
print("==========================")
trapezium = Shape("Trapezium",15,30)
trapezium.area()
```

Output

```
Area: 125.0
==========================
Area: 100
==========================
Area: 225.0
==========================
Area: 450
==========================
Area: Shape unknown
```

Task 12

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

Driver Code

```
# Write your code here

c1 = Calculator()
print("==================")
val = c1.calculate(10, 20, '+')
print("Returned value:", val)
c1.showCalculation()
print("==================")
val = c1.calculate(val, 10, '-')
print("Returned value:", val)
c1.showCalculation()
print("==================")
val = c1.calculate(val, 5, '*')
print("Returned value:", val)
c1.showCalculation()
print("==================")
val = c1.calculate(val, 16, '/')
print("Returned value:", val)
c1.showCalculation()
```

Output

```
Calculator is ready!
==================
Returned value: 30
10 + 20 = 30
==================
Returned value: 20
30 - 10 = 20
==================
Returned value: 100
20 * 5 = 100
==================
Returned value: 6.25
100 / 16 = 6.25
```

Task 13

Design the **Programmer** class such a way so that the following code provides the expected output.

Hint:

- Write the constructor with appropriate printing and multiple arguments.
- Write the addExp() method with appropriate printing and argument.
- Write the prinDetails() method

[You are not allowed to change the code below]

# Write your code here.

```
p1 = Programmer("Ethen Hunt", "Java", 10)
p1.printDetails()
print('-------------------------')
p2 = Programmer("James Bond", "C++", 7)
p2.printDetails()
print('-------------------------')
p3 = Programmer("Jon Snow", "Python", 4)
p3.printDetails()
p3.addExp(5)
p3.printDetails()
```

OUTPUT:

```
Horray! A new programmer is born
Name: Ethen Hunt
Language: Java
Experience: 10 years.
-------------------------
Horray! A new programmer is born
Name: James Bond
Language: C++
Experience: 7 years.
-------------------------
Horray! A new programmer is born
Name: Jon Snow
Language: Python
Experience: 4 years.
Updating experience of Jon Snow
Name: Jon Snow
Language: Python
Experience: 9 years.
```

Task 14

```python
class Test:
    def __init__(self):
        self.sum = 0
        self.y = 0

    def methodA(self):
        x=0
        y =0
        y = y + 7
        x = y + 11
        self.sum = x + y
        print(x , y, self.sum)

    def methodB(self):
        x = 0
        self.y = self.y + 11
        x = x + 33 + self.y
        self.sum = self.sum + x + self.y
        print(x , self.y, self.sum)
```

Write the output of the following code:

```python
t1 = Test()
t1.methodA()
t1.methodA()
t1.methodB()
t1.methodB()
```

x	y	sum

Task 15

```python
class Scope:
    def __init__(self):
        self.x, self.y = 1, 100
    def met1(self):
        x = 3
        x = self.x + 1
        self.y = self.y + self.x + 1
        x = self.y + self.met2() + self.y
        print(x, self.y)
    def met2(self):
        y = 0
        print(self.x, y)
        self.x = self.x + y
        self.y = self.y + 200
        return self.x + y
```

Write the output of the following code:

```python
q2 = Scope()
q2.met1()
q2.met2()
q2.met1()
q2.met2()
```

x	y

Task 16

```python
class Test3:
    def __init__(self):
        self.sum, self.y = 0, 0
    def methodA(self):
        x, y = 2, 3
        msg = [0]
        msg[0] = 3
        y = self.y + msg[0]
        self.methodB(msg, msg[0])
        x = self.y + msg[0]
        self.sum = x + y + msg[0]
        print(x, y, self.sum)
    def methodB(self, mg2, mg1):
        x = 0
        self.y = self.y + mg2[0]
        x = x + 33 + mg1
        self.sum = self.sum + x + self.y
        mg2[0] = self.y + mg1
        mg1 = mg1 + x + 2
        print(x, self.y, self.sum)
```

Write the output of the following code:

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

x	y	sum

Task 17

```python
class Test5:
    def __init__(self):
        self.sum, self.y = 0, 0
    def methodA(self):
        x = 0
        z = 0
        while (z < 5):
            self.y = self.y + self.sum
            x = self.y + 1
            print(x, self.y, self.sum)
            self.sum = self.sum + self.methodB(x, self.y)
            z += 1
    def methodB(self, m, n):
        x = 0
        sum = 0
        self.y = self.y + m
        x = n - 4
        sum = sum + self.y
        print(x, self.y, sum)
        return self.sum
```

Write the output of the following code:
 	
t5 = Test5()
t5.methodA()
	x	y	sum