



Inspiring Excellence

Course Title: Programming Language II

Course Code: CSE 111

Lab Assignment no: 2

Question 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.

Question 2

Design a class called **Flower** with the instance variables so that after executing the following line of code the desired result shown in the output box will be printed.

[You are not allowed to change the code below]

```

#Write your class code here
flower1 = Flower()
flower1.name="Rose"
flower1.color="Red"
flower1.num_of_petal=6
print("Name of this flower:", flower1.name)
print("Color of this flower:",flower1.color)
print("Number of petal:",flower1.num_of_petal)
print("=====")
flower2 = Flower()

```

```

flower2.name="Orchid"
flower2.color="Purple"
flower2.num_of_petal=4
print("Name of this flower:",flower2.name)
print("Color of this flower:",flower2.color)
print ("Number of petal:",flower2. num_of_petal)

```

#Write the code for subtask 2 and 3 here

Output:

Name of this flower: Rose
Color of this flower: Red
Number of petal: 6
=====

Name of this flower: Orchid
Color of this flower: Purple
Number of petal: 4

Subtask:

1) Design the class Flower with default constructor to get the above output. 2) At the end of the given code, also print the address of flower1 and flower2 objects. 3) Do they point to the same address? Print ('they are same' or 'they are different') at the very end to answer this question.

Question 3

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

```

Subtask:

1) Design the class using a parameterized constructor.

Output:

```

Heath Ledger
Mind Game
False
=====
Joaquin Phoenix
Laughing out Loud
True
=====
different
same

```

- 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.

Question 4

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

```

Question 5

Design the **Player** class so that the code gives the expected output.

[You are not allowed to change the code below]

Write Your Class Code Here

```

player1 = Player()
player1.name = "Ronaldo"

```

```

player1.jersey_number = 9
player1.position = "Striker"
print("Name of the Player:", player1.name)
print("Jersey Number of player:",
player1.jersey_number) print("Position of player:",
player1.position)
print("=====")
player2 = Player()
player2.name = "Neuer"
player2.jersey_number = 1
player2.position = "Goal Keeper"
print("Name of the player:", player2.name)
print("Jersey Number of player:",
player2.jersey_number) print("Position of player:",
player2.position)

```

Output:

```

Name of the Player: Ronaldo
Jersey Number of player: 9
Position of player: Striker
=====
Name of the player: Neuer
Jersey Number of player: 1
Position of player: Goal Keeper

```

Question 6

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

Question 7

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

Question 8

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
```

Question 9

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

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

x CANCEL x

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

=====

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 \$

\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

Question 10

A class has been designed for this question. Solve the questions to get the desired result as shown in the output box.

[You are not allowed to change the code below]

```
class Wadiya():
    def __init__(self):
        self.name = 'Aladeen'
        self.designation = 'President Prime Minister Admiral General'
        self.num_of_wife = 100
        self.dictator = True
```

#Write your code for subtask 1, 2, 3 and 4 here

Output:

Part 1:

Name of President: Aladeen

Designation: President Prime Minister Admiral General

Number of wife: 100

Is he/she a dictator: True

Part 2:

Name of President: Donald Trump

Designation: President

Number of wife: 1

Is he/she a dictator: False

Subtask:

- 1) Create an object named wadiya.
- 2) Use the object to print the values as shown in part 1 (Also print the sentence Part 1)
- 3) Use the same object to change and print the values in part 2 (Also print the sentence Part 2)
- 4) Did changing the instance variable values using the same object, affect the previous values of Part 1? (Print 'previous information lost' or 'No, changing had no effect on previous value')

Question 11

Write the output of the following code:

| | | Output |
|----|----------------------------------------|--------|
| 1 | <code>class Human:</code> | |
| 2 | <code>def __init__(self):</code> | |
| 3 | <code>self.age = 0</code> | |
| 4 | <code>self.height = 0.0</code> | |
| 5 | | |
| 6 | <code>h1 = Human()</code> | |
| 7 | <code>h2 = Human()</code> | |
| 8 | <code>h1.age = 21</code> | |
| 9 | <code>h1.height = 5.5</code> | |
| 10 | <code>print(h1.age)</code> | |
| 11 | <code>print(h1.height)</code> | |
| 12 | <code>h2.height = h1.height - 3</code> | |
| 13 | <code>print(h2.height)</code> | |
| 14 | <code>h2.age = h1.age</code> | |
| 15 | <code>h1.age += h1.age</code> | |

| | | |
|----|-------------------------------------|--|
| 16 | <code>print(h1.age)</code> | |
| 17 | <code>h2 = h1</code> | |
| 18 | <code>print(h2.age)</code> | |
| 19 | <code>print(h2.height)</code> | |
| 20 | <code>h1.age += h1.age</code> | |
| 21 | <code>h2.height += h2.height</code> | |
| 22 | <code>print(h1.age)</code> | |
| 23 | <code>print(h1.height)</code> | |
| 24 | <code>h2.age += h2.age</code> | |
| 25 | <code>h1.age = h2.age</code> | |
| 26 | <code>print(h2.age)</code> | |

Question 12

| | | |
|---|----------------------------------|--------|
| 1 | <code>class Student:</code> | Output |
| 2 | <code>def __init__(self):</code> | |
| 3 | <code>self.name = None</code> | |
| 4 | <code>self.cgpa = 0.0</code> | |
| 5 | <code>s1 = Student()</code> | |
| 6 | <code>s2 = Student()</code> | |
| 7 | <code>s3 = None</code> | |

| | | |
|----|------------------------------------------------|--|
| 8 | <code>s1.name = "Student One"</code> | |
| 9 | <code>s1.cgpa = 2.3</code> | |
| 10 | <code>s3 = s1</code> | |
| 11 | <code>s2.name = "Student Two"</code> | |
| 12 | <code>s2.cgpa = s3.cgpa + 1</code> | |
| 13 | <code>s3.name = "New Student"</code> | |
| 14 | <code>print(s1.name)</code> | |
| 15 | <code>print(s2.name)</code> | |
| 16 | <code>print(s3.name)</code> | |
| 17 | <code>print(s1.cgpa)</code> | |
| 18 | <code>print(s2.cgpa)</code> | |
| 19 | <code>print(s3.cgpa)</code> | |
| 20 | <code>s3 = s2</code> | |
| 21 | <code>s1.name = "old student"</code> | |
| 22 | <code>s2.name = "older student"</code> | |
| 23 | <code>s3.name = "oldest student"</code> | |
| 24 | <code>s2.cgpa = s1.cgpa - s3.cgpa + 4.5</code> | |
| 25 | <code>print(s1.name)</code> | |
| 26 | <code>print(s2.name)</code> | |
| 27 | <code>print(s3.name)</code> | |

| | | |
|----|-----------------------------|--|
| 28 | <code>print(s1.cgpa)</code> | |
| 29 | <code>print(s2.cgpa)</code> | |
| 30 | <code>print(s3.cgpa)</code> | |

Question 13

Write the output of the following code:

| | | |
|----|--------------------------------------------------------|--------|
| 1 | <code>class Ninja:</code> | Output |
| 2 | <code>def __init__(self):</code> | |
| 3 | <code>self.rank = 0</code> | |
| 4 | <code>self.stamina = 0.0</code> | |
| 5 | | |
| 6 | <code>naruto = Ninja()</code> | |
| 7 | <code>yellow_flash = Ninja()</code> | |
| 8 | <code>naruto.rank = 1</code> | |
| 9 | <code>naruto.stamina = 95.0</code> | |
| 10 | <code>print(naruto.rank)</code> | |
| 11 | <code>print(naruto.stamina)</code> | |
| 12 | <code>yellow_flash.stamina = naruto.stamina - 2</code> | |
| 13 | <code>print(yellow_flash.stamina)</code> | |
| 14 | <code>yellow_flash.rank += (naruto.rank + 1)</code> | |

| | | |
|----|--------------------------------------------------------|--|
| 15 | <code>print(yellow_flash.rank)</code> | |
| 16 | <code>minato = yellow_flash</code> | |
| 17 | <code>print(minato.rank)</code> | |
| 18 | <code>print(minato.stamina)</code> | |
| 19 | <code>naruto.rank = minato.rank - 1</code> | |
| 20 | <code>naruto.stamina = yellow_flash.stamina + 3</code> | |
| 21 | <code>print(naruto.rank)</code> | |
| 22 | <code>print(naruto.stamina)</code> | |
| 23 | <code>naruto.rank = -(-naruto.rank)</code> | |
| 24 | <code>yellow_flash.stamina = -(-minato.stamina)</code> | |
| 25 | <code>print(naruto.rank)</code> | |
| 26 | <code>print(minato.stamina)</code> | |