ID: 20CE123 **Name: Shubham Sareliya**
Subject Code: CE259 – Python **Date: 03-02-22**

Decorator Task 1:

Definition: Decorator with argument

Code:

```python
# Decorator for Upper Case

def make_upper(fn):
    def wrapper(String):
        return string.upper()
    return wrapper

@make_upper
def hello(string):
    return string

string = 'I Love Web Development'
print(hello(string))
```

Output:

```
PS E:\College\Sem 4\CE259_Python> python -u "e:\College\Sem 4\CE259_Python\Programs\03_Feb\decorator_task_1.py"
I LOVE WEB DEVELOPMENT
PS E:\College\Sem 4\CE259_Python>
```

ID: 20CE123 **Name: Shubham Sareliya**
Subject Code: CE259 – Python **Date: 03-02-22**

Map Task (Lambda):

Definition: Use Map with Lambda Function

Code:

```
item_list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

square = list(map(lambda x: x**2, item_list))

print(square)

for i in item_list:
    print(list(map(lambda function:function(i), [lambda x: x+x, lambda x: x**2, lambda x: x**3, lambda x: x**4])))
```

Output:

```
PS E:\College\Sem 4\CE259_Python> python -u "e:\College\Sem 4\CE259_Python\Programs\03_Feb\mapLambda.py"
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
[2, 1, 1, 1]
[4, 4, 8, 16]
[6, 9, 27, 81]
[8, 16, 64, 256]
[10, 25, 125, 625]
[12, 36, 216, 1296]
[14, 49, 343, 2401]
[16, 64, 512, 4096]
[18, 81, 729, 6561]
[20, 100, 1000, 10000]
PS E:\College\Sem 4\CE259_Python>
```

ID: 20CE123 **Name: Shubham Sareliya**
Subject Code: CE259 – Python **Date: 03-02-22**

Map Task 1:

Definition: Given a list of fruits, use map function and make two lists (FruitsStartingWithA, FruitsStartingWithE), and append the corresponding items into both lists.

Code:

```
fruits = ['apple', 'banana', 'cherry', 'durian', 'fig',
'grape', 'honeydew', 'jackfruit', 'kiwi', 'lemon', 'mango',
'nectarine', 'orange', 'papaya', 'peach', 'pear',
'pineapple', 'plum', 'pomegranate', 'quince', 'raspberry',
'strawberry', 'tangerine', 'watermelon']

fruits_starting_with_a = []
fruits_ending_with_e = []

def distribute_fruits(fruit):
        if fruit[0] == 'a':
            fruits_starting_with_a.append(fruit)
        if fruit[-1] == 'e':
            fruits_ending_with_e.append(fruit)

list(map(distribute_fruits, fruits))
print(f"Fruits name starting with 'a':
{fruits_starting_with_a}")
print(f"Fruits name ending with 'e':
{fruits_ending_with_e}")
```

Output:

```
PS E:\College\Sem 4\CE259_Python> python -u "e:\College\Sem 4\CE259_Python\Programs\03_Feb\tempCodeRunnerFile.py"
Fruits name starting with 'a': ['apple']
Fruits name ending with 'e': ['apple', 'grape', 'nectarine', 'orange', 'pineapple', 'pomegranate', 'quince', 'tangerine']
PS E:\College\Sem 4\CE259_Python>
```

ID: 20CE123 **Name: Shubham Sareliya**
Subject Code: CE259 – Python **Date: 03-02-22**

Map Task 2:

Definition: Make round_off function using map function, also give attributes(upper and lower bound of the list of rounding off).

Code:

```
circle_area = [23.145, 25.894, 63.857, 42.857, 12.567,
18.857, 23.145, 25.894, 63.857, 42.857, 12.567, 18.857]

a = int(input("Enter the lower bound of the list: "))
b = int(input("Enter the upper bound of the list: "))
c = int(input("Enter the number of decimal places: "))

def round_off(x):
    return round(x, c)

print(list(map(round_off, circle_area[a:b])))
```

Output:

```
PS E:\College\Sem 4\CE259_Python> python -u "e:\College\Sem 4\CE259_Python\Programs\03_Feb\tempCodeRunnerFile.py"
Enter the lower bound of the list: 2
Enter the upper bound of the list: 5
Enter the number of decimal places: 2
[63.86, 42.86, 12.57]
PS E:\College\Sem 4\CE259_Python>
```