Assignment 3

February 17, 2025

1 week-3

1.1 Name: Ayesha Subhashanee Hallala Hewage

1.2 Date:12.02.2025

Question 01: Write a lambda expression to get the product of two numbers.

```
Run test for expression(5,6)
Output:30
```

[134]:
```
x = lambda num1,num2: num1*num2
x(5,6)
```

[134]: 30

Question 02: Write a function to get the area of a circle from the radius.

```
Run test for function (10)
Output: 314.1592653589793
```

[137]:
```
import math

def circle_area(radius):
    return math.pi * radius ** 2

#return test
print(circle_area(10))
```

```
314.1592653589793
```

Question 03: Build a simple calculator which can: add, subtract, multiply, divide.

```
Run test for function(2,5,'d')
Output: 0.4
```

[140]:
```
def calculator(a, b, operation):
    if operation == 'a':  # Addition
        return a + b
    elif operation == 's':  # Subtraction
        return a - b
```

```
    elif operation == 'm':  # Multiplication
        return a * b
    elif operation == 'd':  # Division
        if b != 0:
            return a / b
        else:
            return "Error: Division by zero"
    else:
        return"Invalid operation"
# Run test
print(calculator(2, 5, 'd'))
```

```
0.4
```

Question 04: Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.

```
Run test for r =Rectangle(5,10)
            r.area()
Output: 50
```

[143]:
```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width
    def area(self):
        return self.length * self.width
r = Rectangle(5, 10)
print(r.area())
```

```
50
```

Question 05: Define a class named Shape and its subclass Square.

Shape objects can be constructed by name and lengh has an area function wich return 0

Square subclass has an init function which take a length and name as argument and has an area method and describe method what prints the name of the shape.

Print the are from Square class.

```
Run test for: s = Square('square
print(s.area())
print(s.describe())

Output: The area is:
                    25
```

[146]:
```
class shape:
    def __init__(self, name, length):
        self.name = name
```

```
        self.length = length
    def are(self):
        return 0

class square(shape):
    def __init__(self, name, length):
        super().__init__(name, length)

    def area(self):
        print("The area is:")
        return self.length ** 2
    def describe(self):
        print("this is a:", self.name)
s= square('square', 5)
print(s.area())
print(s.describe())
```

```
The area is:
25
this is a: square
None
```

[ ]: