Assignment_3

Name: Nandana Sri Dharmarathne Gangana Pahala Hene Gedara

Date : 15-02-2025

Student ID : EN-001101

Excise 01

1. Write a lambda expression to get the product of two numbers. Run test for expression(5,6) Output: 30

In [1]:
```
# Lambda Expression
product = lambda a, b: a * b
print(product(5 , 6))
```

```
30
```

Excise 02

2. Write a function to get the area of a circle from the radius.
   Hint: remember to import the right modul for being able to calculte the area of the circle.

   Run test for function(10)
   Output: 314.1592653589793

In [2]:
```
#Function Definition
import math

def circle_area(radius):
    return math.pi * radius ** 2
# Test the function with radius 10

print(circle_area(10))
```

```
314.1592653589793
```

Excise 03

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

   Hint: solve by writing a function that takes as argument two numbers and the operation and returns the desired output.

   Run test for function(2,5,'d')

   Output: 0.4

In [3]:
```
# Perform basic arithmetic operations: addition, subtraction, multiplication, and division.
def calculator(a, b, operation):
    if operation == 'x': # Addition
        return a + b
    elif operation == 's': # Subtraction
        return a - b
    elif operation == 'm': # Multiplication
        return a * b
    elif operation == 'd': # Division
        return a / b if b != 0 else "Error: Division by zero"
    else:
        return "Invalied operation"
print(calculator(2, 5, 'd'))
```

```
0.4
```

Excise 04

4. 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

In [4]:
```
# Define a class named Rectangle
class rectangle:
# A class to represent a rectangle.
#Attributes:
# length (float or int): The length of the rectangle.
# width (float or int): The width of the rectangle.

# Methods:
#area(): Computes and returns the area of the rectangle.

    def __init__(self, length, width):
        self.length = length
        self.width = width
    def area(self):
        return self.length * self.width
# Test the class functionality
r = rectangle(5, 10) # Create a rectangle object with length=5 and width=10
print(r.area()) # Expected Output: 50
```

```
50
```

Excise 05

5. Define a class named Shape and its subclass Square.
   Shape objects can be consrtucted by name and length 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 a describe method what prints the name of the Shape. Print the area from Square class.

   Run test for: s = Square('square',5)

print(s.area())
   print(s.describe())
Output: The area is:
   25
This is a: square

In [5]:
```
# Define the parent class 'shape'
class shape:
    # Initialize the shape with a name and length
    def __init__(self, name, length):
        self.name = name  # Store the name of the shape
        self.length = length  # Store the length (for square, it could be the side length)

    # Define a method to return a default area (this can be overridden by child classes)
    def area(self):
        return 0  # Default area method that returns 0

# Define the child class 'square' that inherits from 'shape'
class square(shape):
    # Initialize the square with a name and side length
    def __init__(self, name, length):
        super().__init__(name, length)  # Call the parent constructor to initialize name and length

    # Override the area method to calculate the area of a square
    def area(self):
        print("The area is:")  # Print a message indicating that the area will be printed
        return self.length ** 2  # Return the area of the square (side^2)

    # Define a method to describe the shape
    def describe(self):
        print("This is a:", self.name)  # Print the name of the shape

# Create an object 's' of the 'square' class with name 'square' and side length 5
s = square('square', 5)

# Call the area method of the square object and print the result
print(s.area())  # Expected Output: "The area is:" followed by 25

# Call the describe method of the square object
print(s.describe())  # Expected Output: "This is a: square"
```

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

In [ ]: