

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

In [3]:

```
multi_lambda = lambda num1, num2: num1 * num2
multi_lambda(5,6)
```

Out[3]:

30

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 [5]:

```
import math

def area_circle(num_radius):      #create function
    return math.pi * (num_radius ** 2)

area_circle(10)
```

Out[5]:

314.1592653589793

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 [11]:

```
def calculator(num1, num2, operator):

# create dictionary to transform strings to operators
    ops = {"a": (lambda num1, num2: num1+num2), "s": (lambda num1, num2: num1-num2), \
          "m": (lambda num1, num2: num1*num2), "d": (lambda num1, num2: num1/num2)}

    return ops[operator](num1, num2)

calculator(2, 5, "d")
```

Out[11]:

0.4

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 [12]:

```
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)
r.area()
```

Out[12]:

50

5. Define a class named Shape and its subclass Square. Shape objects can be constructed by name and length has an area function which return 0. Square subclass has an init function which takes a length and name as argument and has an area method and a describe method which 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 [2]:

```
class Shape():
    def __init__(self, name, length):
        self.name = name
        self.length = length

    def area(self):
        return 0

class Square(Shape):
    def __init__(self, name, length):
        super().__init__(name, length)

    def area(self):
        return self.length ** 2

    def describe(self):
        return self.name

s = Square('square', 5)
print(s.area())
print("This is a:", s.describe())
```

25

This is a: square