# Assignment 05 Lab Report

Student Name: Pablo Sabater
Northeastern University
College of Engineering
Department of Electrical and Computer Engineering

Course Title: EECE2140 Computing Fundamentals for Engineers
Instructor: Fatema Nafa
GitHub Repository: https://github.com/pablosabaterlp/Assignment05_Sabater.git

March 7, 2024

# Student Information

Assignment: *Assignment 05*
Student Name: *Pablo Sabater*
Date: *March 7, 2024*

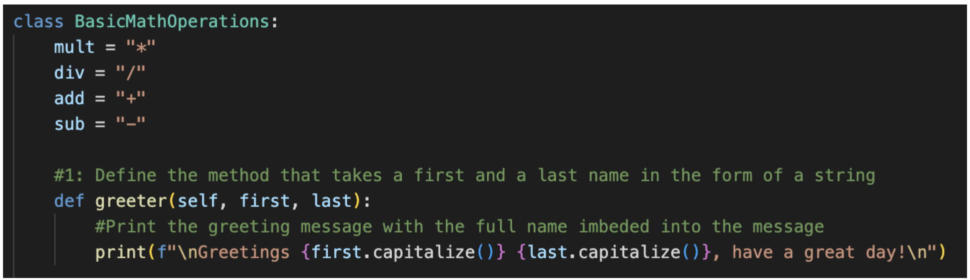
# 1 Pseudocode for Problem 1

## Function: Greeter

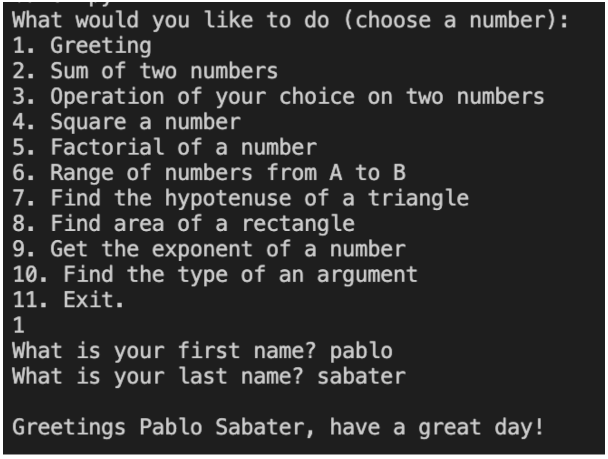
```
Function greeter
    Input: first name, last name
    Output: greeting message with users first and last name

    Begin
        input both names
        print message "Greetings (first name) (last name)"
    End
```

Explanation: The greeting is pretty simple to do, as it just takes the first and last name of the user and appends them in the string for the greeting message using any method.

Code:



```
class BasicMathOperations:
    mult = "*"
    div = "/"
    add = "+"
    sub = "-"

    #1: Define the method that takes a first and a last name in the form of a string
    def greeter(self, first, last):
        #Print the greeting message with the full name imbeded into the message
        print(f"\nGreetings {first.capitalize()} {last.capitalize()}, have a great day!\n")
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
1
What is your first name? pablo
What is your last name? sabater

Greetings Pablo Sabater, have a great day!
```

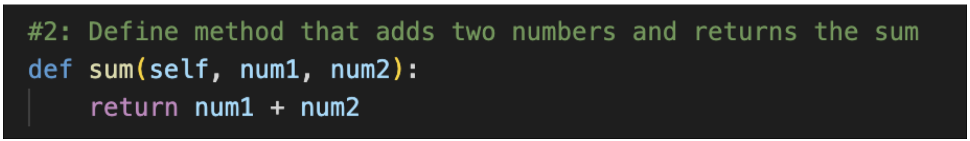
# 2 Pseudocode for Problem 2

## Function: Sum of two numbers

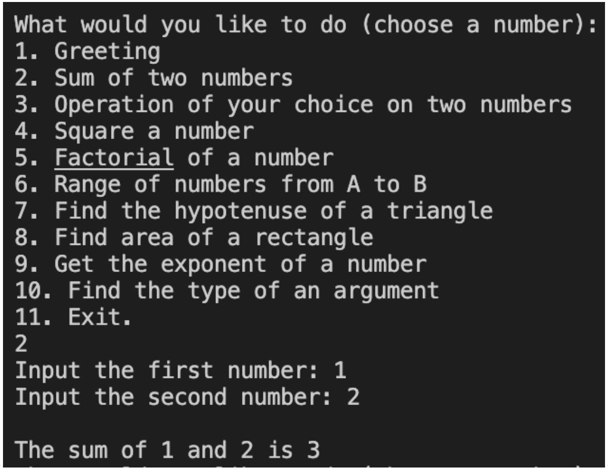
```
Function sum
    Input: number 1, number 2
    Output: sum of the two numbers

    Begin
        input numbers 1 and 2
        return (number 1 + number 2)
        print the return of the function
    End
```

Explanation: This function takes two numbers as input and returns their sum using the '+' operator.

Code:



```python
#2: Define method that adds two numbers and returns the sum
def sum(self, num1, num2):
    return num1 + num2
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
2
Input the first number: 1
Input the second number: 2

The sum of 1 and 2 is 3
```

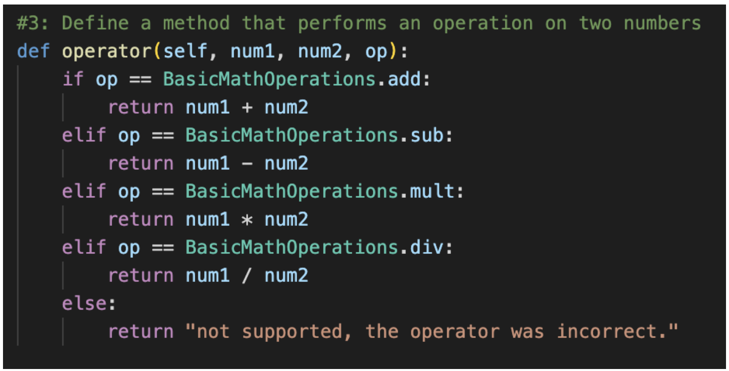
# 3 Pseudocode for Problem 3

## Function: Operation of choice on two numbers

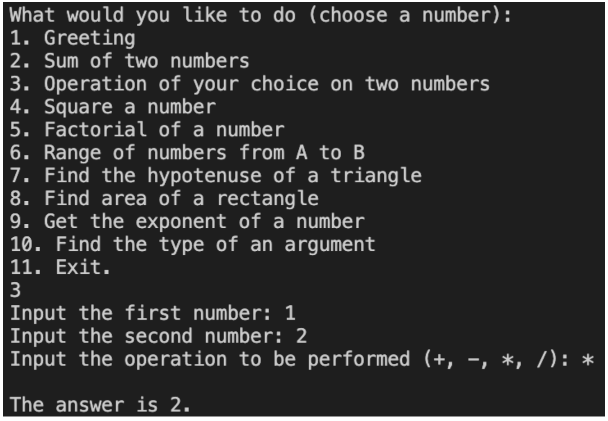
```
Function operator
    Input: number 1, number 2, operator
    Output: result of the operation on the two numbers

    Begin
        input numbers and operator
        check if the operator is +
            if so, perform +
        else check if operator is -
            if so, perform -
        else check if operator is *
            if so, perform *
        else check if operator is /
            if so, perform /
        else tell user operator is not valid
        print the return of the function
    End
```

Explanation: Function takes two numbers and one of the basic math operations. It checks to see which operation was given by the user and returns the result of the operation.

Code:



```
#3: Define a method that performs an operation on two numbers
def operator(self, num1, num2, op):
    if op == BasicMathOperations.add:
        return num1 + num2
    elif op == BasicMathOperations.sub:
        return num1 - num2
    elif op == BasicMathOperations.mult:
        return num1 * num2
    elif op == BasicMathOperations.div:
        return num1 / num2
    else:
        return "not supported, the operator was incorrect."
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
3
Input the first number: 1
Input the second number: 2
Input the operation to be performed (+, -, *, /): *

The answer is 2.
```

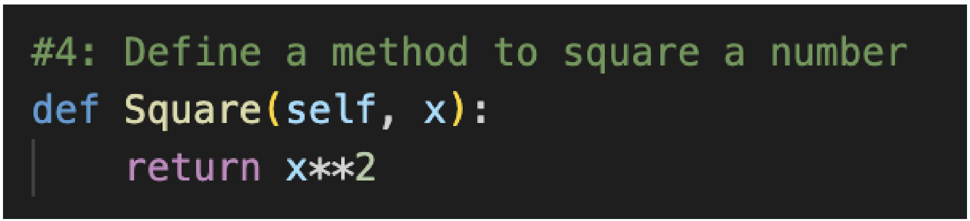
# 4 Pseudocode for Problem 4

## Function: Square of a number

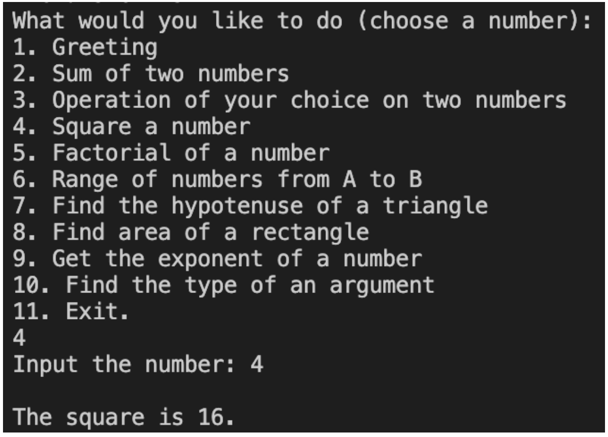
```
Function Square
    Input: number
    Output: square of the number

    Begin
        input number n
        raise number to the power of two
        return number
    End
```

Explanation: Function takes the square of a number by using the ** operator which raises the number to a power, in this case 2.

Code:



```python
#4: Define a method to square a number
def Square(self, x):
    return x**2
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
4
Input the number: 4

The square is 16.
```

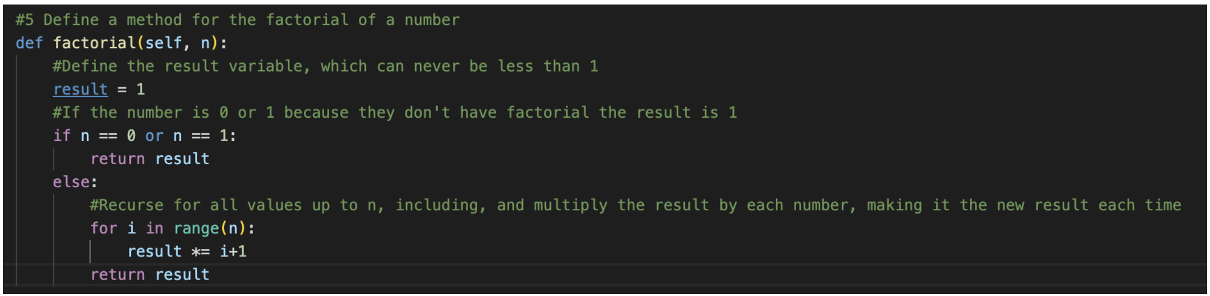
# 5 Pseudocode for Problem 5

## Function: Factorial of a number

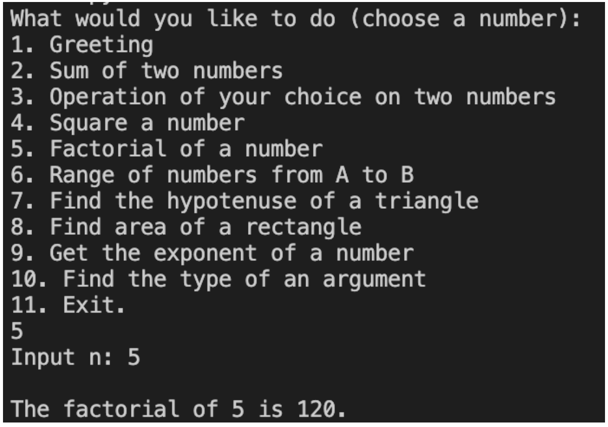
```
Function factorial
    Input: number
    Output: factorial of the number

    Begin
        input number n
        initalize the result which will always start at 1
        check if the number "n" inputted was 1 or 0
            if so produce the default result value
        else multiply result by every number from 1-n
        return result
    End
```

Explanation: Function takes a number n and returns its factorial which is the multiplication of every number from 1 to n. To do this the function uses a for loop for that range, 1-n, where each number is multiplied to the result of the previous multiplications.

Code:



```
#5 Define a method for the factorial of a number
def factorial(self, n):
    #Define the result variable, which can never be less than 1
    result = 1
    #If the number is 0 or 1 because they don't have factorial the result is 1
    if n == 0 or n == 1:
        return result
    else:
        #Recurse for all values up to n, including, and multiply the result by each number, making it the new result each time
        for i in range(n):
            result *= i+1
        return result
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
5
Input n: 5

The factorial of 5 is 120.
```

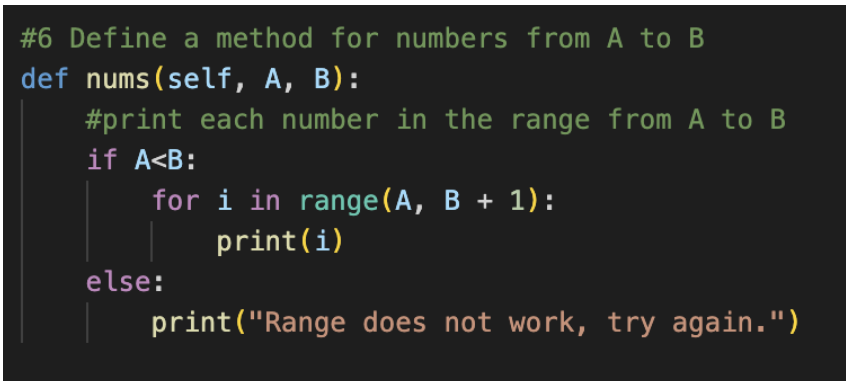
# 6 Pseudocode for Problem 6

## Function: Display every number from A to B

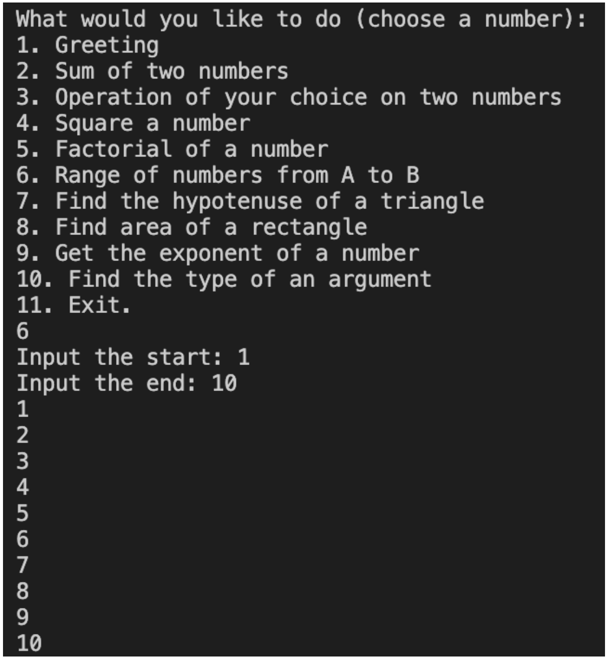
```
Function nums
    Input: start number A, end number B
    Output: list of numbers from A to B, including A and B

    Begin
        input values A and B
        check if B > A
            if so use for loop to print every number from A to B
        else print an error
    End
```

Explanation: Function takes two numbers and prints the list of numbers between them, including the two numbers, using a for loop. Function also produces error if the range is inconsistent.

Code:



```python
#6 Define a method for numbers from A to B
def nums(self, A, B):
    #print each number in the range from A to B
    if A<B:
        for i in range(A, B + 1):
            print(i)
    else:
        print("Range does not work, try again.")
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
6
Input the start: 1
Input the end: 10
1
2
3
4
5
6
7
8
9
10
```

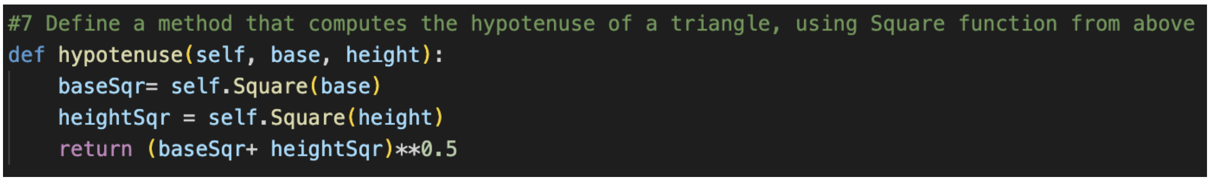
# 7 Pseudocode for Problem 7

## Function: Hypotenuse Finder

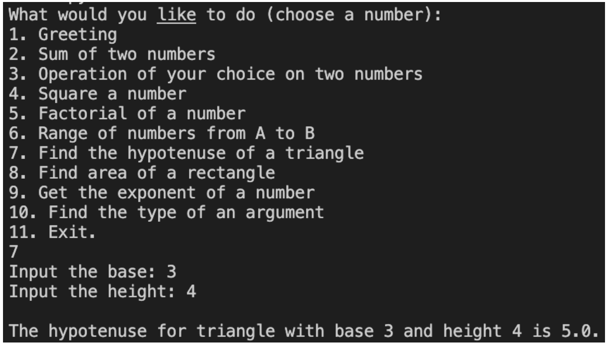
```
Function hypotenuse
    Input: base and height of triangle
    Output: hypotenuse of triangle

    Begin
       input base and height
       square base and height using square function from class
       return the value of the two squares to the power of 1/2
    End
```

Explanation: Function finds the hypotenuse of a right triangle using Pythagorean theorem. To avoid importing libraries, we use **0.5 which is the same as the square root.

Code:



```
#7 Define a method that computes the hypotenuse of a triangle, using Square function from above
def hypotenuse(self, base, height):
    baseSqr= self.Square(base)
    heightSqr = self.Square(height)
    return (baseSqr+ heightSqr)**0.5
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
7
Input the base: 3
Input the height: 4

The hypotenuse for triangle with base 3 and height 4 is 5.0.
```

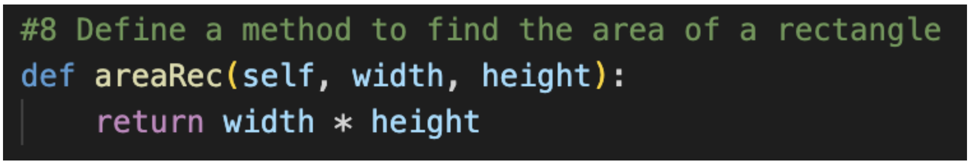
# 8 Pseudocode for Problem 8

## Function: Area of a rectangle

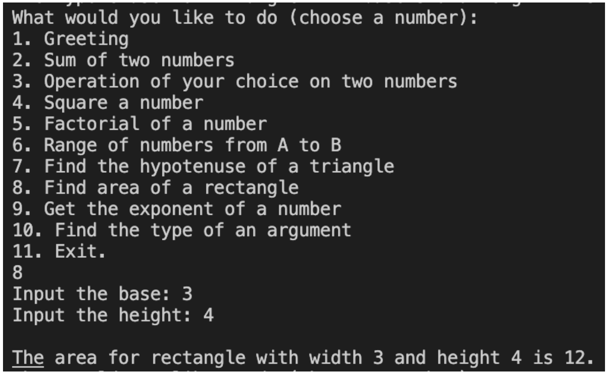
```
Function areaRec
    Input: height and width
    Output: area of rectangle

    Begin
        input width and height
        multiply width and height
        return result
    End
```

Explanation: Function using the formula base*height to compute the area of the rectangle, where base is labeled as width in this case.

Code:



```python
#8 Define a method to find the area of a rectangle
def areaRec(self, width, height):
    return width * height
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
8
Input the base: 3
Input the height: 4

The area for rectangle with width 3 and height 4 is 12.
```

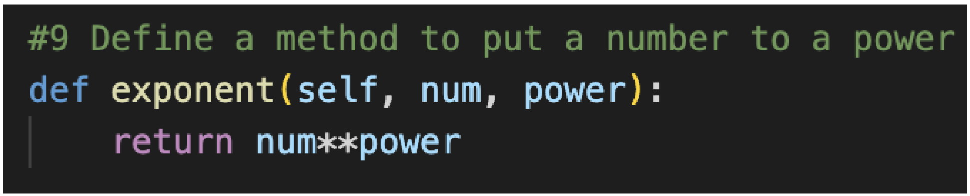
# 9 Pseudocode for Problem 9

## Function: Exponent of a number

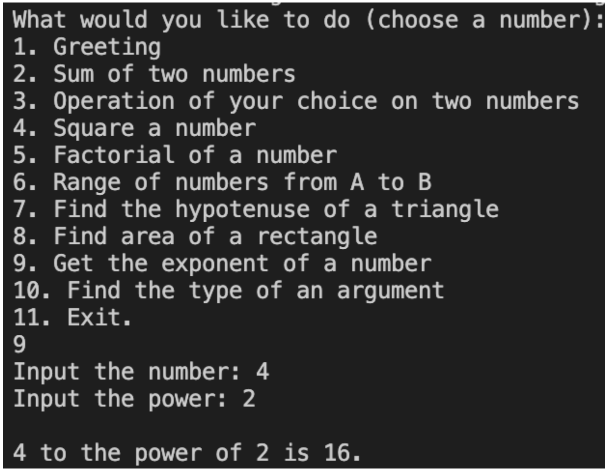
```
Function exponent
    Input: number and power
    Output: number raised to the power

    Begin
        input number and power
        use ** to raise number to power
        return result
    End
```

Explanation: Function uses ** operator to raise number to power.

Code:



```
#9 Define a method to put a number to a power
def exponent(self, num, power):
    return num**power
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
9
Input the number: 4
Input the power: 2

4 to the power of 2 is 16.
```

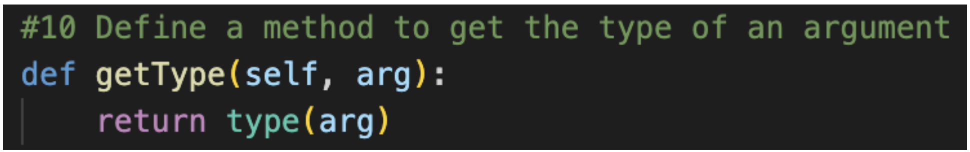
# 10 Pseudocode for Problem 10

## Function: Type checker

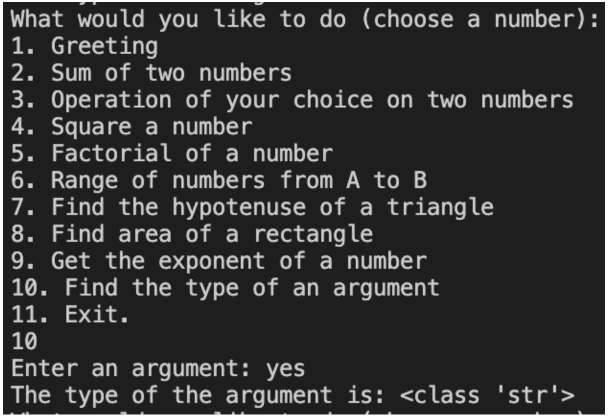
```
Function getType
    Input: argument
    Output: type of argument

    Begin
        input argument
        use function type() of the inputted argument
        return type
    End
```

Explanation: Function uses built in type() function to get the type (int, float, string, etc.) of the inputted argument.

Code:



```
#10 Define a method to get the type of an argument
def getType(self, arg):
    return type(arg)
```



```
What would you like to do (choose a number):
1. Greeting
2. Sum of two numbers
3. Operation of your choice on two numbers
4. Square a number
5. Factorial of a number
6. Range of numbers from A to B
7. Find the hypotenuse of a triangle
8. Find area of a rectangle
9. Get the exponent of a number
10. Find the type of an argument
11. Exit.
10
Enter an argument: yes
The type of the argument is: <class 'str'>
```