Python-Functions

Introduction:

This assignment is designed to deepen our understanding of Python functions, a fundamental aspect of programming in Python. Through practical examples, you will explore the use of built-in functions, define and call custom functions, and work with loops and conditional statements. You will also learn to differentiate between local and global variables, and how to use default argument values in functions. Each question aims to enhance your coding skills and problem-solving abilities, providing you with a solid foundation in Python functions.

Functions in Python:

In Python, a function is a reusable block of code designed to perform a specific task. Functions help to organize code, making it more readable and efficient by avoiding repetition.

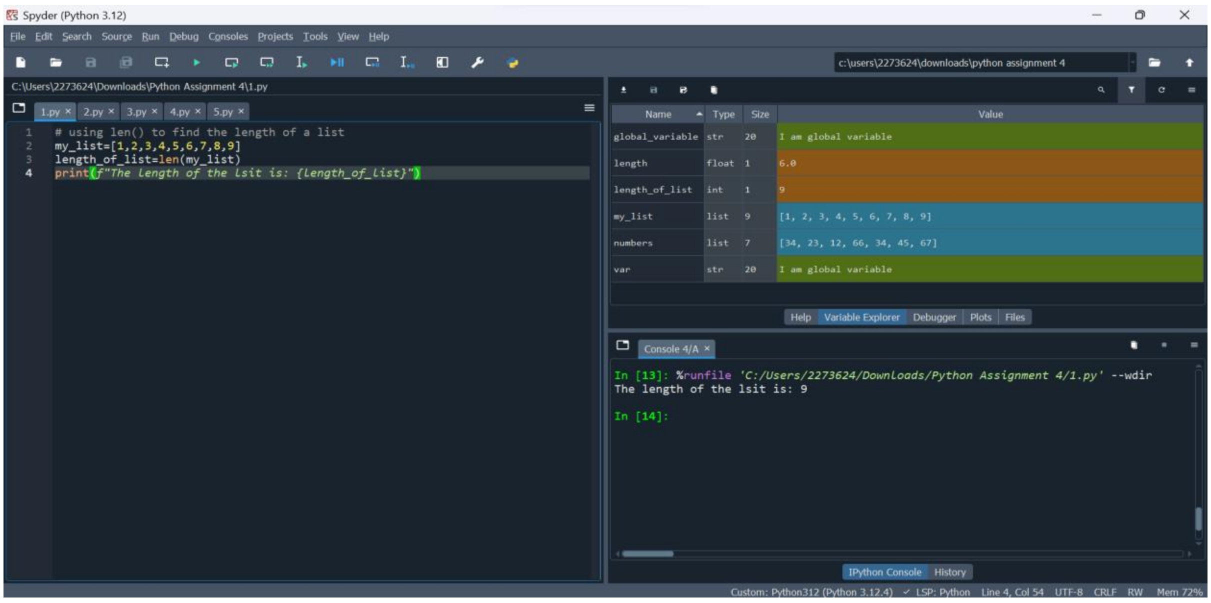
Defining a Function:

A function is defined using the def keyword, followed by the function name, parentheses (), and a colon ‘:’. The code block within the function is indented.

Exercises

1. **What does the len() function do in Python? Write a code example using len() to find the length of a list.**
   a. Description: The len() function returns the number of items in an object. It is commonly used with lists, strings, and dictionaries.
   b. Advantage: It provides a simple and efficient way to determine the size of collections, which is often necessary for iteration, condition checking, and more.
   c. Code:

```
# using len() to find the length of a list.
my_list=[1,2,3,4,5,6,7,8,9]
length_of_list=len(my_list)
print(f"The length of the list is: {length_of_list}")
```


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
c:\users\2273624\downloads\python assignment 4
1.py 2.py 3.py 4.py 5.py
Help Variable Explorer Debugger Plots Files
Console 4/A
IPython Console History
Custom: Python312 (Python 3.12.4) LSP: Python Line 4, Col 54 UTF-8 CRLF RW Mem 72%

C:\Users\2273624\Downloads\Python Assignment 4\1.py

```python
# using len() to find the length of a list
my_list=[1,2,3,4,5,6,7,8,9]
length_of_list=len(my_list)
print(f"The length of the lsit is: {length_of_list}")
```

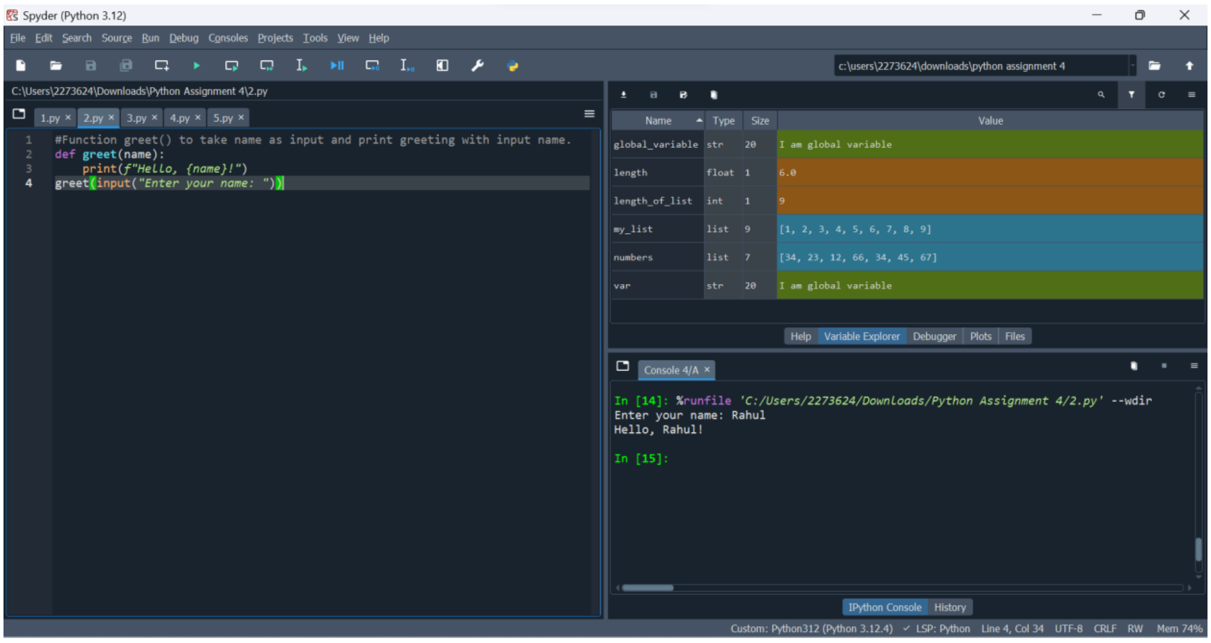
Name	Type	Size	Value
global_variable	str	20	I am global variable
length	float	1	6.0
length_of_list	int	1	9
my_list	list	9	[1, 2, 3, 4, 5, 6, 7, 8, 9]
numbers	list	7	[34, 23, 12, 66, 34, 45, 67]
var	str	20	I am global variable

```
In [13]: %runfile 'C:/Users/2273624/Downloads/Python Assignment 4/1.py' --wdir
The length of the lsit is: 9

In [14]:
```

2. **Write a Python function greet(name) that takes a person's name as input and prints "Hello, [name]!".**
   a. Description: This function takes a single argument name and prints a greeting message.
   b. Advantage: Customizable greeting messages can be created, enhancing user interaction and personalization.
   c. Code:

```
#Function greet() to take name as input and print greeting with input name.
def greet(name):
  print(f"Hello, {name}!")
greet(input("Enter your name: "))
```


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
c:\users\2273624\downloads\python assignment 4
1.py 2.py 3.py 4.py 5.py
Help Variable Explorer Debugger Plots Files
Console 4/A
IPython Console History
Custom: Python312 (Python 3.12.4) LSP: Python Line 4, Col 34 UTF-8 CRLF RW Mem 74%

C:\Users\2273624\Downloads\Python Assignment 4\2.py

```python
#Function greet() to take name as input and print greeting with input name.
def greet(name):
    print(f"Hello, {name}!")
greet(input("Enter your name: "))
```

Name	Type	Size	Value
global_variable	str	20	I am global variable
length	float	1	6.0
length_of_list	int	1	9
my_list	list	9	[1, 2, 3, 4, 5, 6, 7, 8, 9]
numbers	list	7	[34, 23, 12, 66, 34, 45, 67]
var	str	20	I am global variable

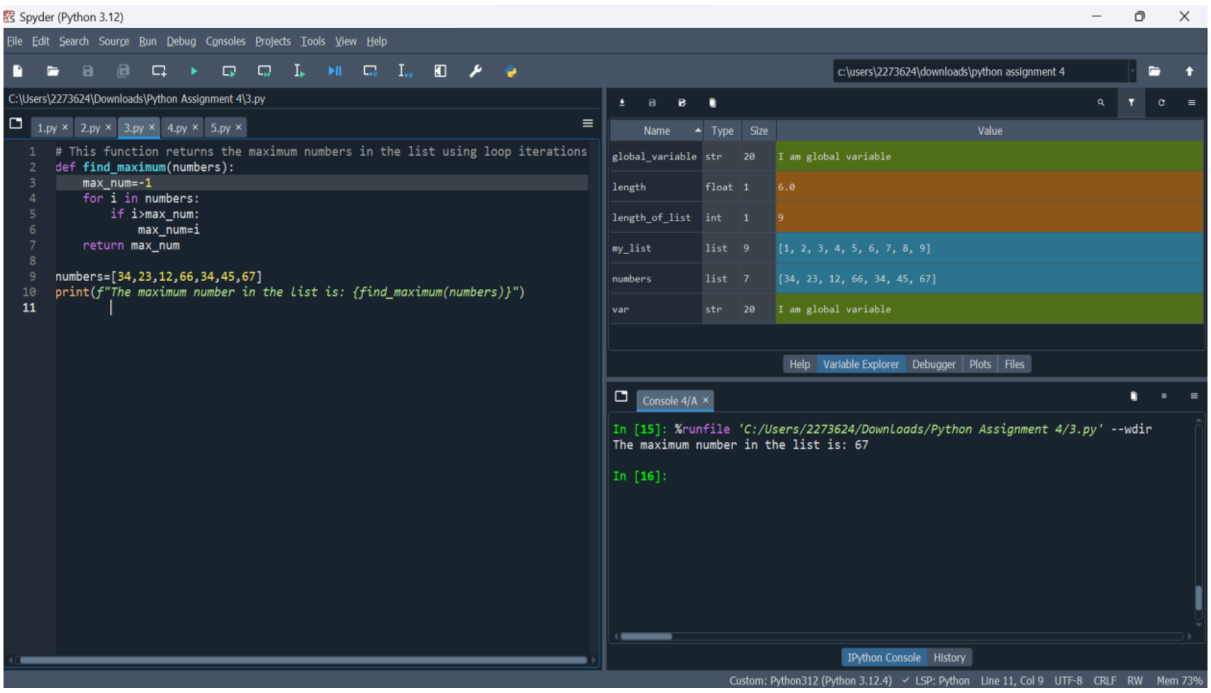
```
In [14]: %runfile 'C:/Users/2273624/Downloads/Python Assignment 4/2.py' --wdir
Enter your name: Rahul
Hello, Rahul!

In [15]:
```

3. **Write a Python function find_maximum(numbers) that takes a list of integers and returns the maximum value without using the built-in max() function. Use a loop to iterate through the list and compare values.**
    a. Description: This function iterates through a list to find the maximum value.
    b. Advantage: By manually finding the maximum value, you gain a better understanding of basic algorithms and loop constructs.
    c. Code:

```
# This function returns the maximum numbers in the list using loop
iterations
def find_maximum(numbers):
    max_num=-1
    for i in numbers:
        if i>max_num:
            max_num=i
    return max_num

numbers=[34,23,12,66,34,45,67]
print(f"The maximum number in the list is: {find_maximum(numbers)}")
```


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
c:\users\2273624\downloads\python assignment 4
C:\Users\2273624\Downloads\Python Assignment 4\3.py
1.py 2.py 3.py 4.py 5.py
Help Variable Explorer Debugger Plots Files
Console 4/A
IPython Console History
Custom: Python312 (Python 3.12.4) ✓ LSP: Python Line 11, Col 9 UTF-8 CRLF RW Mem 73%

```
# This function returns the maximum numbers in the list using loop iterations
def find_maximum(numbers):
    max_num=-1
    for i in numbers:
        if i>max_num:
            max_num=i
    return max_num

numbers=[34,23,12,66,34,45,67]
print(f"The maximum number in the list is: {find_maximum(numbers)}")
```

Name	Type	Size	Value
global_variable	str	20	I am global variable
length	float	1	6.0
length_of_list	int	1	9
my_list	list	9	[1, 2, 3, 4, 5, 6, 7, 8, 9]
numbers	list	7	[34, 23, 12, 66, 34, 45, 67]
var	str	20	I am global variable

```
In [15]: %runfile 'C:/Users/2273624/Downloads/Python Assignment 4/3.py' --wdir
The maximum number in the list is: 67

In [16]:
```

4. Explain the difference between local and global variables in a Python function. Write a program where a global variable and a local variable have the same name and show how Python differentiates between them.

Global Variables	Local Variables
Global variables are defined outside of functions and can be accessed throughout the program.	Local variables are defined within a function and can only be accessed within that function.
Can be modified from any part of the program.	Can be modified but limited to the function/block.
It remains in memory for the duration of the program.	It exists only during the function's executions.

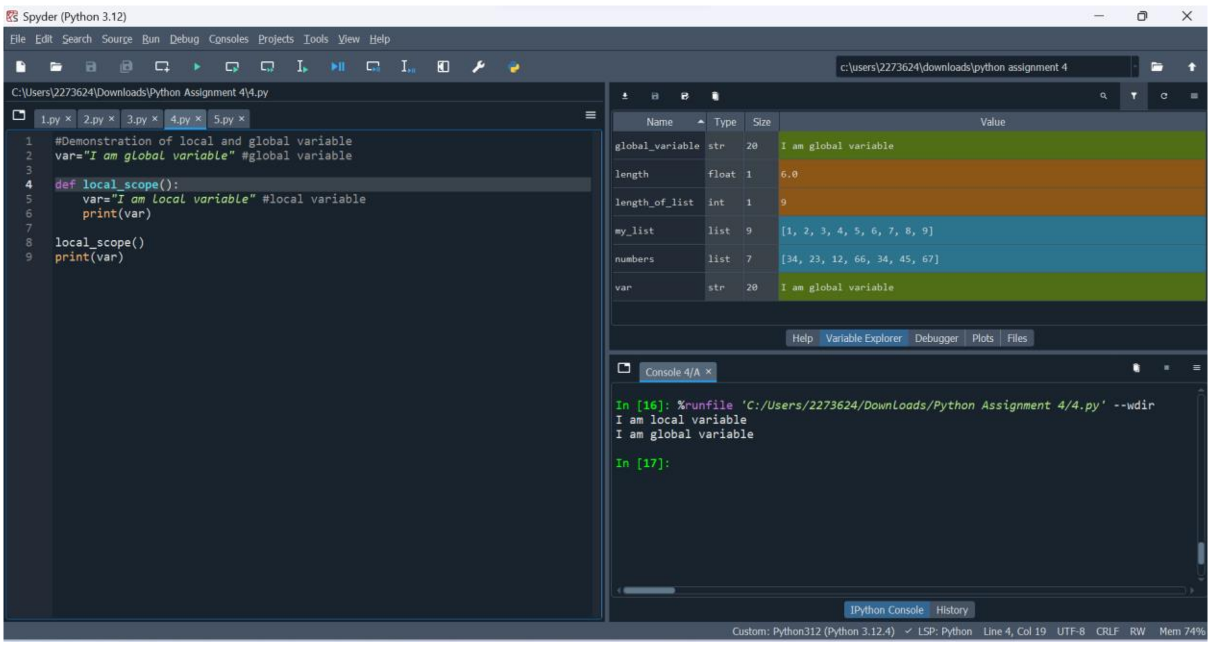
a. Advantage: Understanding the scope of variables helps in managing data and debugging code effectively.

b. Code:

```
#Demonstration of local and global variable

var="I am global variable" #global variable
def local_scope():
   var="I am local variable" #local variable
   print(var)

local_scope()
print(var)
```



Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
c:\users\2273624\downloads\python assignment 4
C:\Users\2273624\Downloads\Python Assignment 4\4.py
1.py 2.py 3.py 4.py 5.py
Help Variable Explorer Debugger Plots Files
Console 4/A
IPython Console History
Custom: Python312 (Python 3.12.4) ✓ LSP: Python Line 4, Col 19 UTF-8 CRLF RW Mem 74%

```
#Demonstration of local and global variable
var="I am global variable" #global variable

def local_scope():
    var="I am local variable" #local variable
    print(var)

local_scope()
print(var)
```

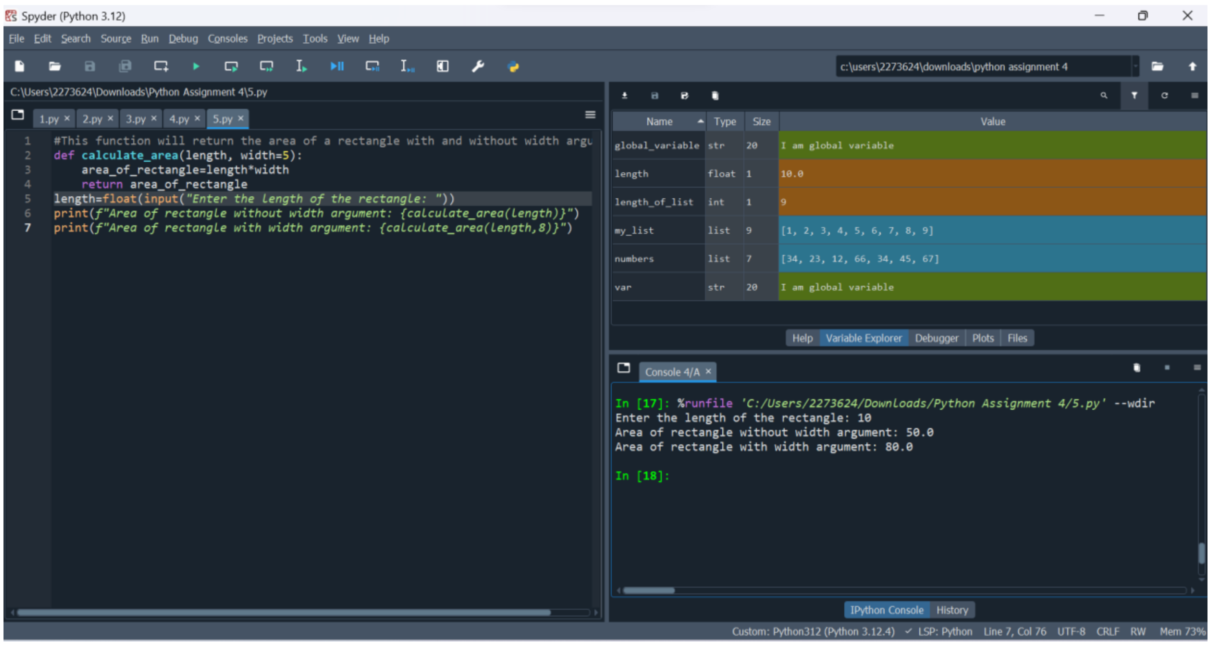
Name	Type	Size	Value
global_variable	str	20	I am global variable
length	float	1	6.0
length_of_list	int	1	9
my_list	list	9	[1, 2, 3, 4, 5, 6, 7, 8, 9]
numbers	list	7	[34, 23, 12, 66, 34, 45, 67]
var	str	20	I am global variable

```
In [16]: %runfile 'C:/Users/2273624/Downloads/Python Assignment 4/4.py' --wdir
I am local variable
I am global variable

In [17]:
```

5. **Create a function calculate_area(length, width=5) that calculates the area of a rectangle. If only the length is provided, the function should assume the width is 5. Show how the function behaves when called with and without the width argument.**
    a. Description: This function calculates the area of a rectangle using length and width, with a width argument and without width argument.
    b. Advantage: Learn about default argument values, which simplify function calls by providing default values for parameters.
    c. Code:

```
#This function will return the area of a rectangle with and without width arguments.
def calculate_area(length, width=5):
  area_of_rectangle=length*width
  return area_of_rectangle
length=float(input("Enter the length of the rectangle: "))
print(f"Area of rectangle without width argument: {calculate_area(length)}")
print(f"Area of rectangle with width argument: {calculate_area(length,8)}")
```


Spyder (Python 3.12)
File Edit Search Source Run Debug Consoles Projects Tools View Help
c:\users\2273624\downloads\python assignment 4
1.py 2.py 3.py 4.py 5.py
Help Variable Explorer Debugger Plots Files
Console 4/A
IPython Console History
Custom: Python312 (Python 3.12.4) LSP: Python Line 7, Col 76 UTF-8 CRLF RW Mem 73%

C:\Users\2273624\Downloads\Python Assignment 4\5.py

```
#This function will return the area of a rectangle with and without width argu
def calculate_area(length, width=5):
    area_of_rectangle=length*width
    return area_of_rectangle
length=float(input("Enter the length of the rectangle: "))
print(f"Area of rectangle without width argument: {calculate_area(length)}")
print(f"Area of rectangle with width argument: {calculate_area(length,8)}")
```

Name	Type	Size	Value
global_variable	str	20	I am global variable
length	float	1	10.0
length_of_list	int	1	9
my_list	list	9	[1, 2, 3, 4, 5, 6, 7, 8, 9]
numbers	list	7	[34, 23, 12, 66, 34, 45, 67]
var	str	20	I am global variable

```
In [17]: %runfile 'C:/Users/2273624/Downloads/Python Assignment 4/5.py' --wdir
Enter the length of the rectangle: 10
Area of rectangle without width argument: 50.0
Area of rectangle with width argument: 80.0

In [18]:
```