Name: ……………………………………………….. ( ) Class: ……… Date: …………………..

|  |  |  |
| --- | --- | --- |
| **4.1** | **Program Development** | **User-defined functions** |

Besides Python’s built-in functions, you can also write your own functions. These are called user-defined functions (UDFs).

**Activity 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **print\_hello.py** |  | **#** | **return\_hello.py** |
| 1  2  3  4  5 | def hello():  # print hello  print('Hello, World!')  print(hello()) |  | 1  2  3  4  5 | def hello():  # return hello  return 'Hello, World!'  print(hello()) |

Read the 2 pieces of code above carefully. Predict the output for each piece of code. Run the code to verify your prediction.

1. *print\_hello.py*

|  |  |
| --- | --- |
| **Predicted Output** | **Actual Output** |
|  |  |

2. *return\_hello.py*

|  |  |
| --- | --- |
| **Predicted Output** | **Actual Output** |
|  |  |

3. Explain the outputs for the 2 pieces of code.

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

**Activity 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **print\_outside.py** |  | **#** | **print\_within.py** |
| 1  2  3  4  5  6  7  8  9 | def reassign\_var():  num1 = 20  num2 = 67    num1 = 19  num2 = 65  reassign\_var()  print(num1)  print(num2) |  | 1  2  3  4  5  6  7  8  9 | def reassign\_var():  num1 = 20  num2 = 67  print(num1)  print(num2)    num1 = 19  num2 = 65  reassign\_var() |

Read the 2 pieces of code above carefully. Predict the output for each piece of code. Run the code to verify your prediction.

4. *print\_outside.py*

|  |  |
| --- | --- |
| **Predicted Output** | **Actual Output** |
|  |  |

5. *print\_within.py*

|  |  |
| --- | --- |
| **Predicted Output** | **Actual Output** |
|  |  |

6. Explain the difference in the outputs for the 2 pieces of code using the concept of global and local variables.

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

**Activity 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **sample1.py** |  | **#** | **sample2.py** |
| 1  2  3  4  5  6  7 | def multiply():  x = num1 \* num2  return x    num1 = 19  num2 = 65  print(multiply()) |  | 1  2  3  4  5  6  7 | def multiply():  num1 = num1 \* num2  return num1    num1 = 19  num2 = 65  print(multiply()) |

Read the 2 pieces of code above carefully. Predict the output for each piece of code. Run the code to verify your prediction.

7. *sample1.py*

|  |  |
| --- | --- |
| **Predicted Output** | **Actual Output** |
|  |  |

8. *sample2.py*

|  |  |
| --- | --- |
| **Predicted Output** | **Actual Output** |
|  |  |

9. Explain why sample2.py incurs an error.

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

10. Alex claims that the programs are bad examples of UDFs as UDFs should request for all the inputs they need as arguments instead of using global variables. Rewrite the multiply() function properly by passing in the required values as arguments.

|  |  |
| --- | --- |
| **#** | **corrected\_sample2.py** |
| 1  2  3  4  5  6  7 |  |