**Overview**

In this activity, you will test your knowledge and skills related to important function concepts covered so far this semester. Upon completing this activity successfully, you should feel confident in your ability to create simple Python programs that effectively utilize various types of functions.

**Instructions**

Complete each of the following in order:

1. Answer the questions from the Questions and Answers table on your own without utilizing any resources. Do not search the internet, review notes, talk to the TA, use AI, and so on.
   * Type your explanations directly into the document.
   * When you need code for your answer, copy-paste it from VS Code and do not attempt to edit the code in the document.
2. After you have answered each question to the fullest of your ability, you may discuss your answers with teammates that have also completed step 1. You may refine your answers based on the conversation(s) you have with your teammates.
3. Create a new document that contains only your complete Questions and Answers table. **Do not include anything else in this document**. Save the file as:

[last name]-knowledge-check.docx (.pdf, .doc, or .odt is also acceptable)

1. Using Claude AI (or ChatGPT 4 if you have a paid subscription) paste the following into a new prompt and then **attach** (upload) the questions and answers document you created in step 3.

| I was given the attached study guide by my instructor. This study guide is for an introduction to programing with functions course that is taught in Python. Can you please review my answers to the questions and provide helpful feedback to me. Please be positive and encouraging in your feedback. If something is wrong, please explain the correct answer to me. |
| --- |

1. Please continue to have a conversation with the AI until you feel confident you have mastered the skills covered up to this point in the semester. **Keep in mind that this is an introductory course**, you are not expected to be an expert.

**Questions and Answers**

| **Question 1: Functions** |
| --- |
| Explain what a function is in Python and describe the components of a function signature. Provide an example of a simple function that takes two parameters and returns their sum. |
| **Answer 1:** |
| A function is a piece of code defined at the start of a program that can be called throughout the program to execute that piece of code. It can be called any number of times. It has different/limited scope so the only data it can access is what is passed into it. |
| **Question 2: Parameters vs. Arguments** |
| Define "parameter" and "argument" in the context of functions. Provide a code example where you define a function with two parameters and then call this function by passing two arguments. Add comments in your code highlighting this difference. |
| **Answer 2:** |
| Parameters are the names of that data that are defined in the function header and used throughout the function. An argument is the same data before it is sent into the function, the differing names to tell the reader where the text is talking. If it is a parameter then it is inside the function with an argument being passed into the parameter. |
| **Question 3: Default Values** |
| Discuss why and how to use default values in function parameters. Write a function that demonstrates the use of default values, explaining how it affects the function's behavior when arguments are not provided during the call. |
| **Answer 3:** |
| Your answer here… |
| **Question 4: Type Hints and Documentation** |
| Explain the purpose of type hints in Python and how they can be helpful in function definitions. Additionally, illustrate how to write function documentation comments. Provide a code example of a function with both type hints and a docstring. |
| **Answer 4:** |
| Type hints are a simple way to comment important information such as how a function works, they are placed at the top of a function and explain what parameters the function has and what is returned, along with a short description of what the function does. An example from a recent assignment: def compute\_volume(radius, height):  """  Calculate the volume of a right circular cylinder  Args:  radius: float  height: float  Returns:  volume: float  """ |
| **Question 5: Variable Scope** |
| Define what variable scope means in Python and provide examples of local and global scopes. Write a simple function to illustrate how local variables differ from global variables within a function context. |
| **Answer 5:** |
| Scope is the area where a variable is defined. A local scope is where a variable is only accessible inside the function that defines it. A global variable is able to be called anywhere after it has been defined. Below is an example:  globalVariable = "thisisaglobalvariable"  *def* localFunction():  localVariable = "thisisalocalvariable"  #Both statements will print wihtout error as the function is able to see the global variable and its local variable  print(globalVariable)  print(localVariable)  #This will throw an error because localVariable is not defined in the global scope and only in local scope  print(localVariable) |
| **Question 6: Program Structure** |
| Describe how to structure a Python program with a protected main() function that calls other functions. Provide a basic example of such a program where the main() function orchestrates calling at least two other functions. |
| **Answer 6:** |
| You would structure the program with imports at the top, any global variables and the main function defined next, then any other functions. At the end of the program you would place an if statement that calls main if the program is run alone, allowing the other functions to be called if someone imported your program into another. Example:  import random  globalVariable = "thisisaglobalvariable"  *def* main():  print("this is the main function")  #call the local function  localFunction()  #call the global function  globalFunction()  *def* globalFunction():  #change the global variable  globalVariable = "thisisachangedglobalvariable"  print(globalVariable)  *def* localFunction():  localVariable = "thisisalocalvariable"  #Both statements will print wihtout error as the function is able to see the global variable and its local variable  print(globalVariable)  print(localVariable)  if \_\_name\_\_ == "\_\_main\_\_":  main() |
| **Question 7: Return Values and Modifying Data** |
| Discuss the concept of returning values from a function and how functions can modify data. Provide examples of a function that returns a value and another that modifies a list passed to it, highlighting the concept of mutability. |
| **Answer 7:** |
| A function can return a value that can then be assigned to another variable to be used elsewhere. This can be done by having a variable be changed by assigning it a new value by calling a function and passing the variable in as something to be meddled with in the function. This kind of variable is mutable. If we were to define a variable as a constant then this would not be possible. Example:  const = 10  changableVariable = 0  changableVariable = changeVariable(changableVariable)  *def* changeVariable(*variable*):  variable = 20  return variable  The constant cannot be changed but by calling changeVariable and passing in the changeable variable we can set its new value to 20 |
| **Question 8: Mutable vs Immutable Data** |
| Explain the difference between mutable and immutable data types in Python. Provide examples of each and show how these differences affect data handling within functions, especially when data is passed as arguments. |
| **Answer 8:** |
| Mutable data types can be changed while immutable types cannot be. Constants cannot be changed and are the only example I can think of. When the data is passed to a function you have to make sure that you don’t attempt to change the data as doing so would throw an error.  const = 10  changableVariable = 0  changableVariable = changeVariable(changableVariable)  *def* changeVariable(*variable*, *const*):  variable = 20 + const  return variable  In this example the constant is used to add to the variable but it does not change so there is no error thrown. |