Question 1: In your own words, add the following terms to your glossary from today: Argument, Parameter, Pass by-Value, Positional Arguments, Keyword Arguments, Default Parameter Variable, Global Variable, Global Constant.

- Parameter → Variables listed in the function header that act as a placeholder for the data the function receives
- Argument → The value passed into a function when called
- Positional Arguments → Arguments that need to be passed in the same order as the functions parameters are created
- Keyword Arguments → Arguments that you can specify using the parameters name to be passed in any order
- Pass-by-Value → This is when the function gets a copy instead of the actual argument's data, making sure changes inside the function don't affect the actual variable
- Default Parameter Variable → Parameter that automatically uses a pre-assigned value when no argument is given when the function is called
- Global Variable → Variables defined outside of any functions, that are able to be accessed anywhere in the program
- Global Constant → Global variables that are made to stay the same and not meant to be changed during the problem (most likely written in all capitals)

Question 2: What is the difference between an argument and a parameter?

Arguments are anything that is passed into a function when the function is called, while a parameter is the actual variable that is given a value from the argument when the function is called. For example, if there was a function called show_score(score), the argument would be the number passed in the brackets, whilst the parameter is the actual score value.

Question 3: What is the scope of a parameter variable? If a parameter value is changed within a function, does this affect the argument that was passed into the function? Explain why or why not.

Parameter variables have a local scope, meaning they are only accessible within a function. If a parameter value is changed within a function, this doesn't change the argument that was passed in the function, as while the argument that was passed in was a global variable, it becomes a local variable when being processed in the function.

Question 6: Without running this program in the Python interpreter, what should it display? Now, run it in the Python interpreter to see if you are right.

I predicted that the output would be 1 3.4, 0 0, and 1 3.4. After running the program in the Python interpreter I learned that my prediction was correct, and that the output was exactly the same as what I guessed.

Question 8: What is the scope of a global variable? How can a function change the value of a global variable?

Global variables have a global scope, and are able to be accessed from anywhere throughout the program. Functions can change the value of a global variable by using the global keyword within the function, then making modifications within the function. This will have a direct effect on the global variable, being altered through the function.

Question 9: Why should global variables be avoided?

Global variables are generally avoided as they make programs harder to debug and understand for developers. This also makes it much more difficult if you have a function that uses a global variable in one file, as you can't move the function to another file without using the variable.This leads to more difficulty with code reusability, and most data that a function needs should be passed through the function.