1. Why are functions advantageous to have in your programs?

Functions reduce the need for duplicate code. This makes programs shorter, easier to read, and easier to update.

1. When does the code in a function run: when it's specified or when it's called?

The code inside a function is run when a function call is made. A function call moves the program execution into the function, and the function call evaluates to the function's return value

1. What statement creates a function?

**def** keyword creates a function in python. The statements in the block of the function must be indented. The def keyword is followed by the function name with round brackets and a colon. example:

**def function1(param):**

**pass**

1. What is the difference between a function and a function call?

A function is a procedure to achieve a particular result while function call is using the function to achieve that task. A function call means invoking or calling that function. Unless a function is called there is no use of the function.

1. How many global scopes are there in a Python program? How many local scopes?

There's only one global Python scope per program execution. From the moment you start a Python program, you’re in the global Python scope. Internally, Python turns your program’s main script into a module called \_\_main\_\_ to hold the main program’s execution. The namespace of this module is the main global scope of your program.

If you operate with the same variable name inside and outside of a function, Python will treat them as two separate variables, one available in the global scope (outside the function) and one available in the local scope (inside the function):

1. What happens to variables in a local scope when the function call returns?

A local variable

* retains its value until the next time the function is called.
* becomes undefined after the function call completes.
* can be used outside the function any time after the function call completes.

**# global & local scope**

**X = 99 # X and func assigned in module: global**

**def func(Y): # Y and Z assigned in function: locals**

**# local scope**

**Z = X + Y # X is not assigned, so it's a global**

**return Z**

1. What is the concept of a return value? Is it possible to have a return value in an expression?

A return statement is used to end the execution of the function call and “returns” the result (value of the expression following the return keyword) to the caller. The statements after the return statements are not executed

1. If a function does not have a return statement, what is the return value of a call to that function?

If a function does not have a return statement, then the special value **None** is returned

1. How do you make a function variable refer to the global variable?

Variables that are created outside of a function are known as global variables. Global variables can be used by everyone, both inside of functions and outside

1. What is the data type of None?

data type of None is NoneType

1. What does the sentence import areallyourpetsnamederic do?

import statement imports a module areallyourpetsnamederic

1. If you had a bacon() feature in a spam module, what would you call it after importing spam? spam.bacon()
2. What can you do to save a programme from crashing if it encounters an error?

Place the line of code that might cause an error in a try clause. Instead of crashing your program right when an exception occurs, you can write the traceback information to a text file and keep your program running. You can look at the text file later, when you’re ready to debug your program.

# example program showcasing the same

import traceback

try:

raise Exception("This is an error")

except:

errorFile = open("errorInfo.txt", 'w')

errorFile.write(traceback.format\_exc())

errorFile.close()

1. What is the purpose of the try clause? What is the purpose of the except clause?

try and except statement is used to handle these errors within our code in Python.

The try: block is used to check some code for errors i.e the code inside the try block will execute when there is no error in the program. The code inside the except: block will execute whenever the program encounters some error in the preceding try block. example like below:

try:

# Some Code

except:

# Executed if error in the

# try block