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

- Functions make code reusable, modular, and easier to read, debug, and maintain.

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

- The code in a function runs when it is called, not when it is defined.

3. What statement creates a function?

- The def statement creates a function. Example:

def my\_function():

pass

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

- A function is the block of code that performs a task.

- A function call is when you invoke or execute the function. Example:

def greet(): # Function

print("Hello")

greet() # Function call

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

- There is one global scope per program.

- There can be multiple local scopes, one for each function call.

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

- Variables in a local scope are destroyed when the function call ends.

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

- A return value is the value sent back by a function after execution.

- Yes, a return value can be used in an expression. Example:

result = add(2, 3) + 5

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

- If no return statement exists, the function returns None.

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

- Use the global keyword inside the function. Example:

x = 10

def modify\_global():

global x

x = 20

10. What is the data type of None?

- The data type of None is NoneType.

11. What does the sentence import areallyourpetsnamederic do?

- It attempts to import a module named areallyourpetsnamederic. If the module doesn't exist, it raises an error.

12. If you had a bacon() feature in a spam module, what would you call it after importing spam?

import spam

spam.bacon()

13. What can you do to save a program from crashing if it encounters an error?

- Use a try-except block to handle errors gracefully. Example:

try:

risky\_code()

except Exception as e:

print(f"Error: {e}")

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

- The try clause contains code that might raise an exception.

- The except clause handles the exception if it occurs, preventing the program from crashing.