Take Test: Quiz 5

Test Information

Description	Definitely watch the videos and read the assigned textbook sections before attempting the quiz
Instructions	
Multiple Attempts	This test allows multiple attempts.
Force Completion	This test can be saved and resumed later.

QUESTION 1 **1 points** Saved

Which of the following are a benefit of using functions in your program?

- 1. Easier to re-use code
- 2. Facilitates teamwork by dividing the code up
- 3. Breaks the program into simpler to solve sub-problems
- 4. Makes your code easier to test and debug
- 5. All of the above
- 6. Only 1, 2 and 3

QUESTION 2 **1 points** Saved

After reaching the end of a function, where in the code will control return in order to continue with the program?

Question Completion Status:

- ○ The control will return to wherever it was when the function was called
- ◉ The control will return to the next line after the function was called
- ○ The program will return to the first line of code in the function

---

QUESTION 3 **1 points** Saved

Consider the following Python script

```
def my_function(num1, num2):
    answer = (num1 + num2) * (num1 - num2)
    print("The answer is:", answer)

x = 42
my_function(x, 37)
```

Which of the following are *parameters* of `my_function`?

- ☑ 37
- ☑ x
- ☐ num2
- ☐ num1

QUESTION 4 **1 points** Saved

Consider the following Python script

```
def my_function(num1, num2):
    answer = (num1 + num2) * (num1 - num2)
    print("The answer is:", answer)
x = 42
my_function(x, 37)
```

Which of the following are *arguments* of `my_function`?

- ☑ num2
- ☐ x

Question Completion Status:

num1

QUESTION 5

1 points Saved

Consider the following code:

```
def add_and_print(num1):
    num1 = num1 + 1
    print(num1)

num1 = 17
num2 = 42
add_and_print(num2)
print(num1)
print(num2)
```

What will be printed first? 43

What will be printed second? 17

What will be printed third? 42

QUESTION 6

1 points Saved

```
#Author: CS021 Instructor
#Introduction to arguments and parameters

#define main function
def main():
    x = 5
    y = 3
    #Calling the add function with x and y as arguments
    add(x,y)
    #Calling the subtract function with 2x and 2y as arguments
    subtract(2*x, 2*y)

#define add function
def add(num1,num2):
    val = num1 + num2
    print(num1, '+', num2, 'is', val)

#define subtract function
def subtract(num1,num2):
    val = num1 - num2
    print(num1, '-', num2, 'is', val)

#calling the main function
main()
```

1
2
3

According to your understanding, pick the best term that represents the blocks highlighted in the picture

1. function call

2. function header

Question Completion Status:

3. function block

QUESTION 7 **1 points** Saved

```
#Author: CS021 Instructor
#Introduction to local variables
#Global variables - DO NOT USE THEM
num1 = 5
num2 = 3

#define main function
def main():
    #Calling the add function
    add()
    #Calling the subtract function
    subtract()

#define add function
def add():
    num1 = 7
    val = num1 + num2
    print(num1, '+', num2, 'is', val)

#define subtract function
def subtract():
    val = num1 - num2
    print(num1, '-', num2, 'is', val)

#calling the main function
main()
```

What is the output of this program

5 + 3 = 8

5 - 3 = 2

7 + 3 = 10
5 - 3 = 2

prints error message
7 + 3 = 10

7 - 3 = 4

QUESTION 8 **1 points** Saved

Question Completion Status:

```
#Author: CS021 Instructor
#Introduction to local variables
#Global variables - DO NOT USE THEM
num1 = 5
num2 = 3

#define main function
def main():
    #Calling the add function
    add()
    #Calling the subtract function
    subtract()

#define add function
def add():
    global num1
    num1 = 7
    val = num1 + num2
    print(num1, '+', num2, 'is', val)

#define subtract function
def subtract():
    val = num1 - num2
    print(num1, '-', num2, 'is', val)

#calling the main function
main()
```

What is the output of this program

- prints error message
- 7 + 3 = 10
  5 - 3 = 2
  5 + 3 = 8
- 5 - 3 = 2
  7 + 3 = 10
- 7 - 3 = 4

QUESTION 9 **1 points** Saved

Which of the following are reasons why we would want to avoid using global variables?

1. Scripts with global variables are more difficult to debug as the value of the global variable could be changed in any function.
2. Functions that rely on the presence of a global variable are harder to re-use in future python programs.
3. There is no need for using global variables in a program. A good program design with functions can avoid the

Question Completion Status:

- (●) 4. All of the above
- ( ) 5. Only 1 and 2

QUESTION 10 **1 points** Saved

Which of these entities can be updated/re-assigned inside a funtion

- ( ) 1. local variables
- ( ) 2. global variables
- ( ) 3. global constants
- (●) 4. all of the above
- ( ) 5. only 1 and 2

Click Save and Submit to save and submit. Click Save All Answers to save all answers.

Save All Answers | Save and Submit