**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Python Activity 12: Void Functions**

|  |
| --- |
| **Learning Objectives**  Students will be able to:  *Content:*   * Explain the meaning and purpose of a function * Recognize a function definition, function header, and function call in a program * Combine the use of functions with if/else statements * Explain programs that use the same function multiple times * Use good test data for programs that include functions   *Process:*   * Write code that includes function definitions and function calls * Write programs that incorporate functions and if/else statements   **Prior Knowledge**   * Python concepts from Activities 1-11 |

**Critical Thinking Questions:**

|  |
| --- |
| **Python Program** |

|  |
| --- |
| **FYI:** A **function** is a segment of code that performs a single task.  A **function definition** is the segment of code that tells the program what to do when the function is  executed. The first line of a function definition is known as the **function header** |

1. Closely examine the Python program above.

a. What Python **keyword** is used to indicate that a code segment is a **function definition**?

def

b. What are the two **function headers** in the Python code?

Def main()\_ and def printMessage()

c. The name of the function is in the function header. What are the names of the two functions?

Def main()\_ and def printMessage()

d. Enter and execute the Python program. What is the output?

Hello programmer!

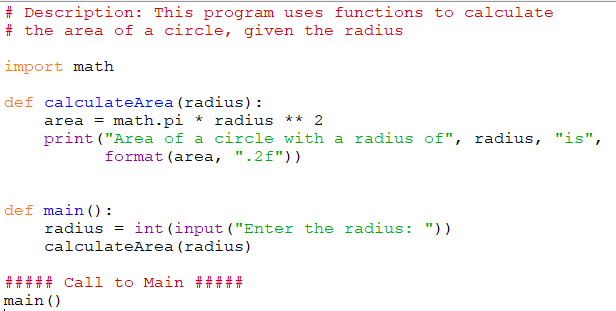
Welcome to python.

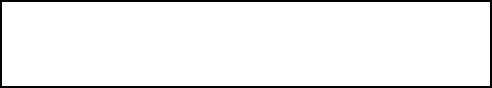
Learn the power of functions!

e. What line of code would you add to the program to print the last two lines twice? Where would you add the code?

PrintMessage(), it goes on the line below PrintMessage in def main()

2. Examine the following program





a. Label the **function definitions** and the **function calls**.

b. The function call and the function definition for *calculateArea* each include a variable within the parentheses. The variable in the function call is known as an **argument**. The variable in the function definition is called a **parameter.** What is the parameter in the function definition? What is its purpose?

The parameter is radius. It sends a copy with the value of radius to be used by calculateArea

c. In this example the parameter in the *function definition* and the argument in the *function call* have the same name. Is this required? No it isn’t it assigns the given variables value to the neew varible

d. Enter and execute the program. Verify your answer to question ‘c’ by changing the variable name in the main function from **radius** to **number**. Do not change the parameter variable name in the function definition. Does the program still work?

Yes, it does still work

e. Write a line of code that calls the **calculateArea** function and sends the value “6” as the argument. Add the line of code to the main program and execute it to be sure it works properly. calculateArea(6)

f. Add another function to the program that calculates and prints the *diameter* of a circle, given the radius as the parameter. Place the function definition above call to the main function of the program. Write the function below.

def diameter(radius):

print("The diameter of the circle is:", radius\*2)

g. Add another line of code to the main function of the program to call the function that was created in part ‘f’. Send the radius entered by the user as the argument to the function. def

calculateArea(radius):

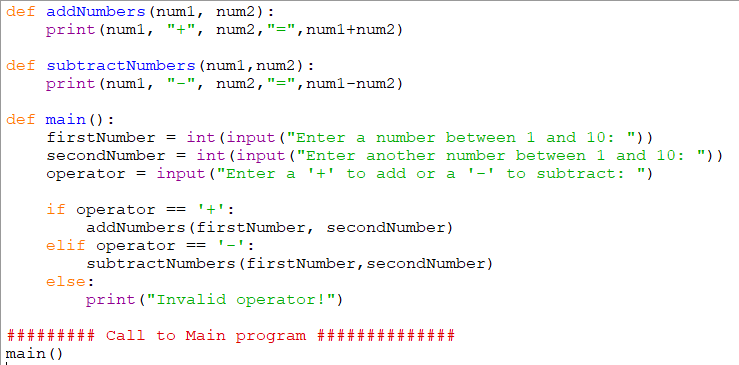
area = math.pi \* radius \*\* 2

print("area of a circle with a radius of", radius, "is",

format(area, ".2f"))

diameter(radius)

3**.** Carefully examine the following program.



a. Place an “\*” next to the first line of code executed by the Python interpreter.

b. Enter and execute the program using the data below. State the output for each set of data.

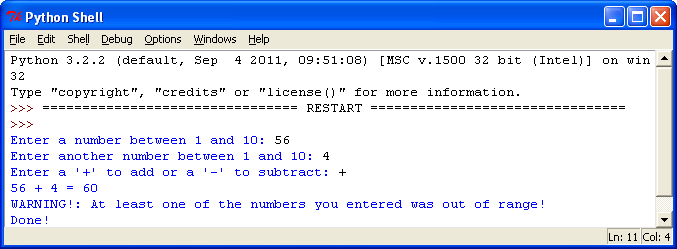
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Data Set** | **Operand 1** | **Operand 2** | **Operator** | **Result** |
| **1** | 2 | 6 | + | 8 |
| **2** | 3 | 8 | - | -5 |
| **3** | 34 | 23 | + | 57 |
| **4** | 4 | 5 | / | Invalid operator |

c. What problems did you notice when you entered Data Sets 3 and 4?

Data set 3 worked just fine for me there I no input validation for the 2 numbers.

Data set 4 uses an invalid operator. But the way the code is you aren’t prompted to reenter a new, valid one

d. Add code to the program that would warn the user about the problem that could occur when data similar to that in Data Set 3 and 4 is entered. See sample output below. Write the revised program below the sample output.



def addNumbers(num1, num2):

print(num1, "+", num2, "=", num1+num2)

if num1 > 10 or num2 > 10:

print("Warning!: At least one of the numbers you entered was out of range!")

elif num1 <0 or num2 <0:

print("Warning!: At least one of the numbers you entered was out of range!")

def subtractNumbers(num1, num2):

print(num1, "-", num2, "=", num1-num2)

if num1 > 10 or num2 > 10:

print("Warning!: At least one of the numbers you entered was out of range!")

elif num1 <0 or num2 <0:

print("Warning!: At least one of the numbers you entered was out of range!")

def main():

firstNumber = int(input("Enter a number between 1 and 10: "))

secondNumber = int(input("Enter another number between 1 and 10: "))

operator = input("Enter a '+' to add or a '-' to subtract: ")

if operator == '+':

addNumbers(firstNumber, secondNumber)

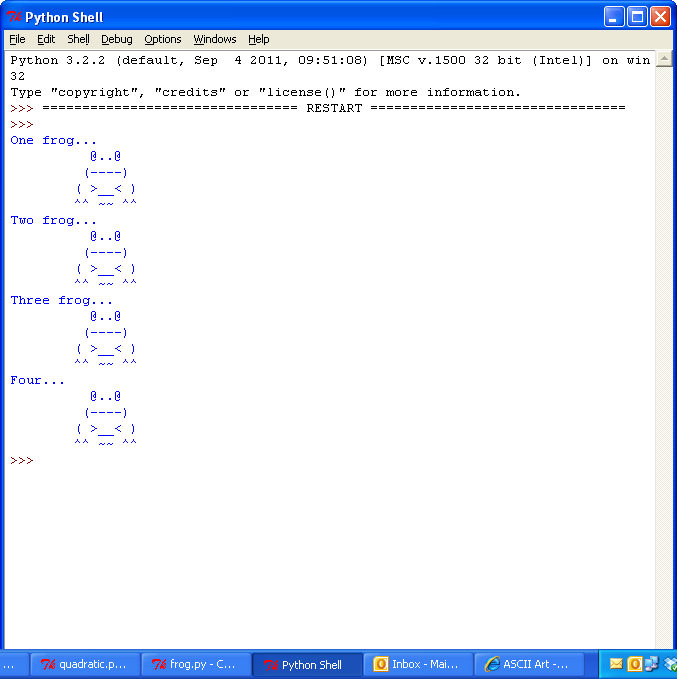
elif operator == '-':

subtractNumbers(firstNumber, secondNumber)

else:

print("Invalid operator")

main()

**Application Questions: Use the Python Interpreter to check your work**

1. Write a function that draws a frog. Call the function to be sure it works. Sample frog:

x=0

def legs():

print("^^ ~~ ^^")

def body():

print("( >\_\_< )")

legs()

def mouth():

print(" (----)")

body()

def eyes():

print(" @..@")

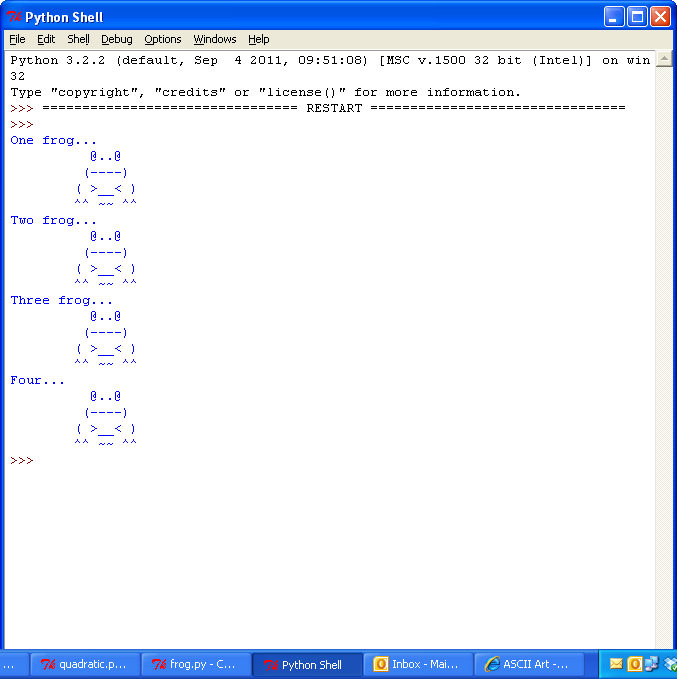
mouth()

def main():

eyes()

main()

2. Expand the program in #1 to produce the following output.

 x=0

def legs():

print("^^ ~~ ^^")

def body():

print("( >\_\_< )")

legs()

def mouth():

print(" (----)")

body()

def eyes():

print(" @..@")

mouth()

def main(x):

if x == 1:

print ("One frog...")

elif x == 2:

print ("Two frog...")

elif x == 3:

print ("Three frog...")

elif x == 4:

print ("Four...")

x=+1

eyes()

while x < 4:

x+=1

main(x)