Metropolitan State University

ICS 140 Computational Thinking with Programming

Class Exercise 9

**Lecture Section**

1. How can a global variable be updated from within a function?

Using the GLOBAL keyword

1. Why should we avoid updating global variables from a function?

Makes it harder to debug code

1. What standard library function can be used to generate a random integer?

Random.randint()

1. What function can be used to guarantee a specific set of “random” values are generated?

Random.seed()

1. What statement in a function can be used to return a value to the statement calling the function?

Return statement

1. What math function can be used to take the square root of a number?

Math.sqrt()

1. How can you return multiple values from a function?

Use commas to separate multiple values after a return statement

1. How do you access functions in another file?

Import them

**Using standard python libraries**

This section will have questions about using the math and random libraries. Unless otherwise stated, you can assume that the math and random libraries have already been imported.

1. What line is used to import the math library before any math functions can be used?

Import math

1. What line is used to import the random library before any random functions can be used?

Import random

1. Write a line of code to round a float variable called **x** up.

Math.ceil()

1. Write a line of code to round a float variable called **y** down.

Math.floor()

1. Write a line to generate a random number between 1 and 0 and store it in a variable called **random\_float**.

random\_float = random.random()

1. Write a line to generate a random number from 10 to 100 that is divisible by 10 and store it in a variable called **random\_ten**.

Random\_ten = random.randrange(10,100,10)

**Writing and Calling Functions**

For this section, write functions that meet the description.

1. Define a function called **circle\_measurements**. This function should receive the **radius** as a parameter and return the circumference and area of the circle.
2. def circle\_measurements(radius):
3. area = math.pi\*radius\*\*2
4. circumference = math.pi\*2\*radius
5. return area,circumference
6. Call the **circle\_measurements** function with a **radius** of 5 described above and store the results in variables called **circumference** and **area**.
7. def circle\_measurements(radius):
8. area = math.pi\*radius\*\*2
9. circumference = math.pi\*2\*radius
10. return area,circumference
11. area, circumference = circle\_measurements(5)
12. Define a function called **calculations** that receives 2 number parameters **num1** and **num2**. Return the **sum** and **average** of these numbers.
13. def calculations(num1,num2):
14. sum = num1 + num2
15. average = sum/2
16. return sum,average
17. Call the **calculations** function for numbers **1** and **3**. Store the results in variables **total** and **mean**.
18. total, mean = calculations(1,2)

**Importing Custom Modules**

For each of the following questions, provide 2 lines, one that imports the module and a second that calls the custom function. Assume the module is in the same folder as the file this code would be in.

1. Call a function called **world** contained in a file **hello.py**. This function has no parameters and returns no values.
2. import hello
3. hello.world()
4. Call a function called **circle\_measurements** in a file called **circle.py**. This function receives a parameter for the **radius** of a circle and returns the circumference and the area. Call this function passing an argument of **10** for the **radius** and store the results in variables **circumference** and **area**.
5. import circle
6. circumference, area = circle.circle\_measurements(10)

**Programming Exercise**

For the following exercise, you will use the python files included in this assignment to create functions for basic mathematic operations.

* The add function will take two values and return their sum.
* The subtract function will take two values and return the value of the second value subtracted from the first value.
* The multiply function will take two values and return their product.
* The divide function will take two integers and return two values. The function will use integer division and will return the quotient and remainder as separate values.
* In the main function, prompt the user for input for each function. Then call the function and print the results as shown in the screenshot below.

Text

Description automatically generated

**If using\_automated\_tests:**

Run the automated test scripts before coding the functions and verify that all tests are failing. Once the code has been written, run the tests again and make sure all tests pass. If necessary tweak the functions for any failing tests. Once the code is working, take a screenshot of all tests passing and paste it into the test results section along with the output calling the functions as shown above. Paste the python code into the python code section.

**Else:**

Run tests manually and paste screenshots of the console.

**Python Code**

def add(x,y):

    return x + y

def subtract(x,y):

    return x - y

def multiply(x,y):

    return x \* y

def divide(x,y):

    quotient = x//y

    remainder = x%y

    return quotient,remainder

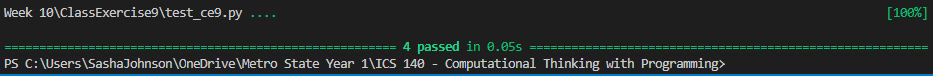
def main():

    pass

if \_\_name\_\_ == '\_\_main\_\_':

    main()

**Test Cases and Results**

****