**Practice Quiz: Expressions and Variables**

Question 1

In this scenario, two friends are eating dinner at a restaurant. The bill comes in the amount of 47.28 dollars. The friends decide to split the bill evenly between them, after adding 15% tip for the service. Calculate the tip, the total amount to pay, and each friend's share, then output a message saying "Each person needs to pay: " followed by the resulting number.



1

2

3

4

5

bill = 47.28

tip = bill \* 0.15

total = bill + tip

share = (total/2)

print("Each person needs to pay: "+str(share))

**Correct**

Great work! You’ve given the computer correct instructions

at each step, paying attention to every detail, and it paid

off!

**1 / 1 point**

Question 2

This code is supposed to take two numbers, divide one by another so that the result is equal to 1, and display the result on the screen. Unfortunately, there is an error in the code. Find the error and fix it, so that the output is correct.



1

2

3

4

numerator = 10

denominator = 10

result =int( numerator / denominator)

print(result)

**Correct**

You got it!. You caught the error and fixed it! Way to go!

**1 / 1 point**

Question 3

Combine the variables to display the sentence "How do you like Python so far?"



1

2

3

4

5

6

7

8

9

word1 = "How"

word2 = "do"

word3 = "you"

word4 = "like"

word5 = "Python"

word6 = "so"

word7 = "far?"

print(word1+" "+ word2+" "+word3+" "+ word4+" "+word5+" "+word6+" "+word7)

**Correct**

Nice job! You caught the error and fixed it! Way to go!

**1 / 1 point**

4.Question 4

This code is supposed to display "2 + 2 = 4" on the screen, but there is an error. Find the error in the code and fix it, so that the output is correct.



1

print("2 + 2 = " + str(2 + 2))

**Correct**

Way to go! You noticed the missing conversion into string,

and fixed it!

**1 / 1 point**

Question 5

What do you call a combination of numbers, symbols, or other values that produce a result when evaluated?



An explicit conversion



**An expression**



A variable

An implicit conversion

**Correct**

Right on! An expression is a combination of values, variables, operators, and calls to functions.

**1 / 1 point**

## Practice Quiz: Functions

Question 1

This function converts miles to kilometers (km).

1. Complete the function to return the result of the conversion
2. Call the function to convert the trip distance from miles to kilometers
3. Fill in the blank to print the result of the conversion
4. Calculate the round-trip in kilometers by doubling the result, and fill in the blank to print the result



**# 1) Complete the function to return the result of the conversion**

**def convert\_distance(miles):**

**return (miles \* 1.6) # approximately 1.6 km in 1 mile**

**# 2) Convert my\_trip\_miles to kilometers by calling the function above**

**x=convert\_distance(55)**

**y=2 \* x**

**# 3) Fill in the blank to print the result of the conversion**

**print("The distance in kilometers is " + str(x))**

**# 4) Calculate the round-trip in kilometers by doubling the result,**

**# and fill in the blank to print the result**

**print("The round-trip in kilometers is " + str(y) )**

**Correct**

Woohoo! You’ve figured out how to make the functions do what

they need to do, and remembered some things from the

previous lessons. Way to go!.

**1 / 1 point**

Question 2

This function compares two numbers and returns them in increasing order.

1. Fill in the blanks, so the print statement displays the result of the function call in order.

Hint: if a function returns multiple values, don't forget to store these values in multiple variables



**# This function compares two numbers and returns them**

**# in increasing order.**

**def order\_numbers(number1, number2):**

**if number2 > number1:**

**return number1, number2**

**else:**

**return number2, number1**

**# 1) Fill in the blanks so the print statement displays the result**

**# of the function call**

**smaller,bigger= order\_numbers(100, 99)**

**print(smaller, bigger)**

**Correct**

Nice! You remembered how to accept multiple return values

from a function. You’re ready for the next lesson!

**1 / 1 point**

Question 3

What are the values passed into functions as input called?



Variables



Return values



**Parameters**



Data types

**Correct**

Nice job! A parameter, also sometimes called an argument, is a value passed into a function for use within the function.

**1 / 1 point**

4.Question 4

Let's revisit our lucky\_number function. We want to change it, so that instead of printing the message, it returns the message. This way, the calling line can print the message, or do something else with it if needed. Fill in the blanks to complete the code to make it work.



**def lucky\_number(name):**

**number = len(name) \* 9**

**return ("Hello " + name + ". Your lucky number is " + str(number) )**

**print(lucky\_number("Kay"))**

**print(lucky\_number("Cameron"))**

**Correct**

Way to go! The function now returns the message, for the

calling line to use it in any way it wants to.

**1 / 1 point**

Question 5

What is the purpose of the def keyword?



**Used to define a new function**



Used to define a return value



Used to define a new variable



Used to define a new parameter

**Correct**

Awesome! When defining a new function, we must use the def keyword followed by the function name and properly indented body.

**1 / 1 point**

## Practice Quiz: Conditionals

Question 1

What's the value of this Python expression: (2\*\*2) == 4?



4



2\*\*2



**True**



False

**Correct**

You nailed it! The conditional operator == checks if two values are equal. The result of that operation is a boolean: either True or False.

**1 / 1 point**

2.Question 2

Complete the script by filling in the missing parts. The function receives a name, then returns a greeting based on whether or not that name is "Taylor".

****

**def greeting(name):**

**if name == "Taylor":**

**return ("Welcome back Taylor!")**

**else:**

**return ("Hello there, " + name)**

**print(greeting("Taylor"))**

**print(greeting("John"))**

**Correct**

Great work! You're getting the hang of conditionals in Python.

**1 / 1 point**

Question 3

What’s the output of this code if number equals 10?

****

**if number > 11:**

**print(0)**

**elif number != 10:**

**print(1)**

**elif number >= 20 or number < 12:**

**print(2)**

**else:**

**print(3)**

2

**Correct**

Right on! Our number is 10, which is smaller than 12, so it matches that condition.

**1 / 1 point**

Question 4

Is "A dog" smaller or larger than "A mouse"? Is 9999+8888 smaller or larger than 100\*100? Replace the plus sign in the following code to let Python check it for you and then answer.



1

2

**print("A dog" + "A mouse")**

**print(9999+8888 + 100\*100)**



"A dog" is larger than "A mouse" and 9999+8888 is larger than 100\*100



**"A dog" is smaller than "A mouse" and 9999+8888 is larger than 100\*100**



"A dog" is larger than "A mouse" and 9999+8888 is smaller than 100\*100



"A dog" is smaller than "A mouse" and 9999+8888 is smaller than 100\*100

**Correct**

You got it! Keep getting Python to do the work for you.

**1 / 1 point**

Question 5

If a filesystem has a block size of 4096 bytes, this means that a file comprised of only one byte will still use 4096 bytes of storage. A file made up of 4097 bytes will use 4096\*2=8192 bytes of storage. Knowing this, can you fill in the gaps in the calculate\_storage function below, which calculates the total number of bytes needed to store a file of a given size?



**def calculate\_storage(filesize):**

**block\_size = 4096**

**# Use floor division to calculate how many blocks are fully occupied**

**full\_blocks = filesize//block\_size**

**# Use the modulo operator to check whether there's any remainder**

**partial\_block\_remainder = filesize%block\_size**

**# Depending on whether there's a remainder or not, return**

**# the total number of bytes required to allocate enough blocks**

**# to store your data.**

**if partial\_block\_remainder > 0:**

**return (block\_size \* (full\_blocks + 1) )**

**return (block\_size)**

**print(calculate\_storage(1)) # Should be 4096**

**print(calculate\_storage(4096)) # Should be 4096**

**print(calculate\_storage(4097)) # Should be 8192**

**print(calculate\_storage(6000)) # Should be 8192**

**Correct**

Awesome! Those were some complicated calculations that you

needed to do, but you did it!

**1 / 1 point**