Name: _____ **Partner:** _____

Python Activity 8: Looping Structures – *WHILE* Loops

Learning Objectives
Students will be able to:
Content:
- Explain the three parts of a **loop**
- Explain the syntax of a **while loop**
- Explain **sentinel-controlled** and **counter controlled** loops
- Explain **short-cut operators**

Process:
- Write code that includes **sentinel-controlled** and **counter controlled** loops
- Write code that uses short-cut operators

Prior Knowledge
- Python concepts from Activities 1-7

Critical Thinking Questions

1. Closely examine the Python program below.

FYI: A **looping structure** allows a block of code to be repeated one or more times. A **while** loop is one of the two looping structures available in Python.

Python Program

```
1  # This program prints a person's name 20 times
2  name = input("Enter your name: ")
3  x = 0
4  while (x < 20):
5      print(name)
6      x += 1
```

a. In the Python code, circle all the code associated with the WHILE loop.

b. Enter and test the code. What does the line of code: **`x+=1`** do?

_____

d. How does the Python interpreter know what lines of code belong to the loop body?

_____

e. **Every loop structure requires three actions.** Identify the line of code in the Python program that corresponds to each of the three actions.
- *Initialize a variable used in the test condition:*

  _____
- *Include a test condition that causes the loop to end when the condition is false:*

  _____
- *Within the loop body, update the variable used in the test condition:*

  _____

2. Enter and execute the following code. Beside each line of code explain what the code does.

```
1   # This program prints numbers from 1 to the
2   # value entered by the user
3   number = int(input("Enter a number: "))
4   x = 1
5   while(x <= number):
6       if(x % 10 == 0):
7           print(x)
8       else:
9           print(x, end=" ")
10      x = x + 1
```

3. The following code should print the numbers from 1 to 10, but it does not print anything. Correct the problem.

```
number = 12
while number <= 10:
  print(number)
  number = number + 1
```

4. Examine the following code:

```
number = 0
while number <= 10:
  print(number)
  number = number - 1
```

a. What will the output be? _____

b. Does the program end? Why or why not? _____

_____

5. Examine the following code:

```
number = 1
while number <= 10:
  if number % 2 == 0:
     print(number, end= "   ")
  number = number + 1
```

a. State the output.

_____

b. What caused the output to display on one line?

_____

c. What control structures are used in this code?

_____ and _____

6. The following directions will create a program that prompts the user to enter a number between 1 and 10. As long as the number is out of range the program re-prompts the user for a valid number. Complete the following steps to write this code.

   a. Write a line of code that prompts the user for a number between 1 and 10.

   _____

   b. Write a **Boolean expression** that tests the number the user entered by the code in step "a." to determine if it is **not** in range.

   _____

   c. Use the Boolean expression created in step "b." to write a **while loop** that executes when the user input is out of range. The body of the loop should tell the user that they entered an invalid number and prompt them for a valid number again.

   _____

   _____

   _____

   _____

   d. Write the code that prints a message telling the user that they entered a valid number.

   _____

   e. Put the segments of code from steps "a-d" together. Enter and execute the code. Does it work properly? If not, correct it and test it again. _____

   f. How many times does the loop execute? _____

> **FYI:** A **looping structure** for which you know the number of times it will execute is known as a *count-controlled* loop.

7. Sometimes a programmer does not know how many times data is to be entered. For example, suppose you want to create a program that adds an unknown amount of positive numbers that will be entered by the user. The program stops adding numbers when the user enters a zero or a negative number. Then the program prints the total. Before creating this program, review the three actions required for all loops:

   a. *Initialize a variable that will be used in the test condition:* What will be tested to determine if the loop is executed? Write a line of code that initializes a variable to be used in the test condition of the loop for this program. The variable should contain a value entered by the user.

   _____

   b. *Include a test condition that causes the loop to end when the condition is false:* What is the test condition for the while loop used in this program?

   _____

   c. *Within the loop body, update the variable used in the test condition:* Write the code for the loop body. Include the code to update the variable in the test condition.

   _____

   _____

   _____

   _____

d. Is this a *count-controlled* loop? Why or why not?

_____

e. Complete the program. Enter and execute the code. Does it work properly? _____

> **FYI:** **Short-cut operators** provide a concise way of creating assignment statements when the variable on the left-hand side of the assignment statement is also on the right-hand side. The addition short-cut operator (+=) is usually used for incrementing a variable.

8. Enter and execute the following code:

```
number = 1
number += 3
print(number)
```

a. What does the "+=" shortcut operator do? _____

b. The code: `x += 5` is equivalent to which of the following lines of code?

- x = 5
- x = y + 5
- x = x + 5
- y = x + 5

c. Replace the operator '+=' with the following **shortcut operators** and execute the code. Explain what each operator does.

- -= _____
- *= _____

9. Enter and execute the following code:

```
bonus = 25
salary += bonus
print("Total salary:", salary)
```

a. What is the output of the preceding code? Is it what you expected?

_____

b. Rewrite the code so that it produces valid output.

_____

_____

_____

_____

c. Is the following line of code valid: `23 += total?` Why or why not? _____

_____

10. The following code should print the numbers beginning with 100 and ending with 0. However it is missing a line of code. Add the missing code, using the shortcut operator. Draw an arrow to indicate where the code belongs.

```
countdown = 100
while countdown > 0:
    print(countdown)
print("Done!")
```

_____

11. Enter and execute the following code:

```
doAgain = "y"
while doAgain == "y":
   word = input("Enter a word:")
   print("First letter of " + word  + " is " + word[0])
   doAgain = input("Type 'y' to enter another word and anything
                    else to quit.")
print("Done!")
```

a. What does the program do? _____

_____

b. What is the variable name used to store the user's input? _____

c. In the print statement, what does **word[0]** represent? _____

d. Change 0 to 1 in **word[0]** in the print statement above. What is printed? _____

e. When does the program end? _____

FYI: A **sentinel-controlled while loop** is a loop that repeats the loop body until the user enters a pre-specified value.

f. Why is the loop in this program an example of a **sentinel control** loop?

_____

g. Examine the print statement in this program:

```
print("First letter of " + word  + " is " + word[0])
```

What happens if you replace the "+" with a ","?

_____

12. Examine the code below.

```
name = "Simone"
cost = 3.56
numApples = 89
```

What type of data is stored in each variable: (integer, floating point, or string)

- name – _____
- cost – _____
- numApples – _____

FYI: A variable that can store only the values **True** and **False** is called a **Boolean variable**.

13. Given the assignment statement: `foundCost = False`
    - What value is stored in the variable **foundCost?** _____
    - What type of data is stored in **foundCost?** _____

Application Questions: Use the Python Interpreter to check your work

1. Write a code segment that prompts the user for an even number. As long as the number is not even, the user should be given a message and prompted again for an even number.

_____

_____

_____

2. Write code segment that prompts the user for a letter from 'a-z'. As long as the character is not between 'a-z', the user should be given a message and prompted again for a letter between 'a-z'.

_____

_____

_____

_____