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

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

“While(x < 20):

print(Name)

x= x+ 1

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

It adds one to the previous value of x

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

Everything after while is what the loop is and then how many times the it should loop by expressing the condition the variable is must be less then the total amount of loops wanted

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:*

X = 0

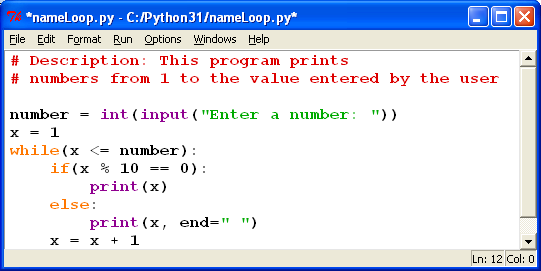
* + - *Include a test condition that causes the loop to end when the condition is false:*

X <20

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

X= X+1

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



Asks for a number and makes it an int \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Sets the variable x equal to one

Prints every number between 1 and the input

If there is a remainder of x divided by 10 it stops printing \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_if not it just prints nothing \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**number = 0**

**while number <= 10:**

**print(number)**

**number = number + 1**

4. Enter and execute the following code:

**number = 0**

**while number <= 10:**

**print(number)**

**number = number - 1**

a. Describe the output. It continues to print numbers 1 less then the previous forever,

b. Does the program end? Why or why not? No because number will never be >=10

5. Enter and execute the following code:

**number = 1**

**while number <= 10:**

**if number % 2 == 0:**

**print(number, end= " ")**

**number = number + 1**

a. State the output.

2, 4, 6, 8, 10,

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

The few digits weren’t enough characters to reach the next line,

c. What control structures are used in this code?

If the remainder of number and 2 isn’t 0 it won’t print and The end=” “ is the spacing between numbers

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.

number = int(input("Enter 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.

0 < number <=10:

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.

while (0> number or number >10):

number = int(input("Enter a number and make sure it is between 1 and 10: "))

if 0 < number <=10:

print("Valid")

else:

print("Invalid number choice")

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

See above

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. Yes see above

f. How many times does the loop execute? As many as it takes to get a valid number

|  |
| --- |
| **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.

num = int(input("Enter your data point: "))

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?

while(num > 0):

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.

while(num > 0):

num = int(input("Enter yout next data point: "))

if (num >0):

num2=(num2 + num)

print("Your total so far is :", num2, sep="")

else:

print("Your final total is :", num2, sep="")

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

Yes, because there is a condition that can end the loop. Also, it only moves forward with input not endlessly

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

|  |
| --- |
| **FYI:** S**hort-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? It adds the number to the variable and stores the new value.

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.

* -= subtracts and stores it under the variable
* \*= multiplies and stores it under the variable

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?

Yes. Error because there is no value for salary.

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

bonus = 25

salary= 100

salary += bonus

print("Total Salary:", salary)

* 1. Is the following line of code valid:  **23 += total?** Why or why not?

No because there is no value for the variable total so it cant run.

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!”)**

countdown -=1

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? It figures out the first letter if a word.

b. What is the variable name used to store the user’s input? word

c. In the print statement, what does **word[0]**represent? That it is supposed to grab the first letter from the entered word.

d. Change 0 to 1 in **word[0]**in the print statement above. What is printed? The second letter of the entered word.

* 1. When does the program end? If u type something other than Y

|  |
| --- |
| **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?

Because if a value other than y is entered the loop ends

g. Examine the print statement in this program:

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

What happens if you replace the “+” with a “,”?

It adds an extra space per comma.

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 – string
  + cost – float
  + numApples – Int

|  |
| --- |
| **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? Str False**

**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.

num=int(input("Enter a number: "))

while(num%2>0):

num=int(input("Enter another 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’.

I don’t understand what this is asking

letter=input("Enter a letter between 'a' and 'z': ")

while(letter<'z' and letter<'a'):

letter=input("Enter a letter between 'a' and 'z': ")

print("Done!")