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

**Python Activity 9: Looping Structures: FOR Loops**

|  |
| --- |
| **Learning Objectives**  Students will be able to:  *Content:*   * Explain the difference between **while loop**  and a **FOR loop** * Explain the syntax of a **FOR loop** * Explain how to use the **range()** function in a **FOR loop** * Explain an **accumulator**  in a **FOR loop**   *Process:*   * Write code that includes **FOR loop** * Write code that uses use **FOR loops** within functions   **Prior Knowledge**   * Python concepts from Activities 1-8 |

**Critical Thinking Questions:**

1. Enter and execute the following two Python programs.

|  |
| --- |
| **WHILE LOOP -- Python Program** |
| **FOR LOOP – Python Program** |

a. What is the output for each program?

Print name 20 times

b. Both programs produce the same output. Which code fragment is more concise?

The for loop

|  |
| --- |
| **FYI:** The Python predefined function - **range()** - is used to define a series of numbers and can be used in a FOR loop to determine the number of times the loop is executed.. |

2. Enter and execute the following code fragments and state the output:

a. for x in range(5):

print(x, end=" ") 0 1 2 3 4

b. for x in range(1,5):

print(x, end=" ") 1234

c. for x in range(3,20,2):

print(x, end=" ") 3 5 9 11 13 15 17

d. numIterations = 6

for x in range(numIterations):

print(x, end=" ") 0 1 2 3 4 5

e. numIterations = 6

for x in range(1, numIterations+1):

print(x, end=" ") 1 2 3 4 5 6

3. After examining the five code fragments in #2, explain how the **range()** function works. Include an explanation of the arguments.

The range fuction takes 3 argyments: lowest value, highest value, and increment on each loop

|  |
| --- |
| **FYI:** In a FOR loop you can include a list of values in place of the **range()** function. |

4. Enter and execute the following code.

for x in [3,6,9,12,15,18]:

print(x, end=” “)

a. Rewrite this code using the **range()** function.

For x in range (3,19,3):

Print(x, end=’’)

b. Why would you use the **range()** function when you could just list the numbers?

When printing a large list the range() is more consise

|  |
| --- |
| **FYI:** The **str()** function converts what is the parentheses ( ) to a String. |

5. Read through the code and determine what it does.

**favorite = input("Enter your favorite ice cream flavor: ")**

**for x in range(1,5):**

**print(str(x) + “.”, favorite, end="\t")**

a. What do you think the program does? Asks for favorite ice cream flavor and prints an ordered list of the favorite ice cream.

b. Enter and execute the code to determine if you were correct. What does the program actually do? Provide a detailed explanation.

Asks for favorite ice cream flavor and prints a list with the favorite ice cream after each number

c. Why is the **str()** function needed in the print statement?

To concatenate the number with the period while printing

6. Complete the arguments in the following range function so that the code prints the even numbers between 100 and 200 inclusive.

**for x in range(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_):**

**print(x)**

7. Complete the arguments in the following range function so that the code prints: 5 4 3 2 1 0.

**for x in range(5,-1,-1\_):**

**print(x)**

|  |
| --- |
| **FYI:** An **accumulator** is a variable that stores the sum of a group of values. |

8. Examine the following code segment.

total = 0

for x in range(5):

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

total += number

print("The total is:",total)

a. Why is the variable **total** initialized to 0 in the first line of code?

Tota; cannot be added to itself without holding a value

b. Explain what the following code does:

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

Asks for input and casts the value to an integer

c. Explain what the following code does: **total += number**

Adds number to totals current value and updates total

d. How many numbers does the program prompt for? 5

e. What is the **accumulator** in the code segment? total

9. Is it better to use a **FOR loop** when you know the number of times the loop should be executed or when you do not know?

When you know the number of times it should run use the range() function otherwise a while loop works well

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

1. Write a code segment using a FOR loop that prints multiples of 5 from 5 to 500, one on a line.

For x in range(5,501,5):

Print(x,end=’’)