**Python Lab 5a: while loops**

|  |  |
| --- | --- |
| 1. The while statement allows the commands with the body of the loop to be repeated, as long as a certain condition is true.  Ask the user to enter an integer. Store the value in x. Then add this code which counts down.  **while (x > 0):**  **print x**  **x = x - 1**  **print ('blastoff!!')** | **enter an integer: 3**  **3**  **2**  **1**  **blastoff!!** |
| 2. Modify the loop to print out whether a number is even or odd. An even number is divisible by 2, and an odd number is not. | **enter an integer: 3**  **3 is odd**  **2 is even**  **1 is odd**  **blastoff!!** |
| 3. Ask the user to enter in the amount of decrease. Use this value in your loop. | **enter an integer: 11**  **enter decrease: 4**  **11 is odd**  **7 is odd**  **3 is odd**  **blastoff!!** |
| 4. Beware infinite loops! Take out the line which decreases x. What happens? Then put the line back in. |  |
| 5. Write a different loop to print out words until the word has a length less than 5. The len() function returns the length of a string.  **print ( word, 'has', len(word), 'letters')** | **enter a word: hello**  **hello has 5 letters**  **enter a word: yellow**  **yellow has 6 letters**  **enter a word: dog**  **goodbye** |
| 6. There are several different ways to write the code for the problem above. Discuss with your peers, or explore other possibilities on your own. |  |
| 7. Write a while loop that counts from 10 to 100 in decimal, binary, and hex. | **10 0b1010 0xa 11 0b1011 0xb 12 0b1100 0xc 13 0b1101 0xd 14 0b1110 0xe 15 0b1111 0xf** |

**Python Lab 5b: Applications of while loop**

1. Write a while loop that prints out all the even numbers between two numbers that the user inputs. A sample run is below.

**Enter the lower number: 36**

**Enter the higher number: 43**

**The even numbers between 36 and 43 are: 36 38 40 42**

2. Write a while loop that prints out all the factors of a given positive integer. Use the mod operation to check if a certain number evenly divides into the given number.

**Enter a positive integer: 27**

**The integers that are factors of 27 are: 1 3 9 27**

3. Question 3, at the end of Section 5.2, asks to express the Euclidean algorithm in pseudocode.

"The Euclidean algorithm finds the greatest common divisor of two positive integers X and Y by the following process: As long as the value of neither X nor Y is zero, assign the larger the remainder of dividing the larger by the smaller. The greatest common divisor, if it exists, will be the remaining non-zero value."

Take a moment and go over the algorithm you created. Check your answer in the back of the book. Then, convert the pseudocode into a Python program and test your code.