05.01 Virtual Lecture Notes (Part 1): while Loops

Counting with a `while` Loop

It is often necessary to execute a loop a specific number of times. This is easily accomplished as indicated in the following example. When a loop counts from a lower number to a higher number, it is referred to as ascending.

```
< 1>  public class CountingWithWhileLoops
< 2>  {
< 3>      public static void main(String [] args)
< 4>      {
< 5>          int counter = 0;
< 6>          int theTerminator = 10;
< 7>
< 8>          while(counter <= theTerminator)
< 9>          {
<10>              System.out.println("counter = " + counter);
<11>              counter++;
<12>          }//end of while loop
<13>
<14>      }//end of main method
<15>  }//end of class
```

Type the code for the CountingWithWhileLoops class into the 5.01 Iterations BlueJ project in the Mod05 Lessons folder. Compile the program and fix any errors, then run the program and observe the output. Carefully study the following line-by-line explanation of the program.

Lines	
< 1>	declares a class name CountingWithWhileLoops
< 2>	opening curly brace to start the class (matches with closing curly brace on line <15>)
< 3>	the **main()** method where program execution begins
< 4>	opening curly brace to start the **main()** method (matches up with line <14>)
< 5>	declares and initializes the **counter** variable for the loop
< 6>	declares and initializes **theTerminator** variable for the loop
< 7>	white space to improve program readability
< 8>	beginning of the while loop with the **boolean** condition in parentheses
< 9>	opening curly brace starting the **while** loop code block (matches up with line < 12>)
<10>	prints the value of the **counter** variable
<11>	increments the **counter** variable
<12>	closing curly brace ending the **while** loop code block (matches up with line < 9>)
<13>	white space to improve program readability
<14>	closing curly brace to end the **main()** method (matches up with line < 4>)

<15> closing curly brace to end the class (matches with opening curly brace on line < 2>)

Modifications

1. Run the program again after changing the initial value of the **theTerminator** variable to various larger numbers. Record the amount of time (in seconds or minutes) a while loop takes to count to 100, 1,000, 10,000, etc.
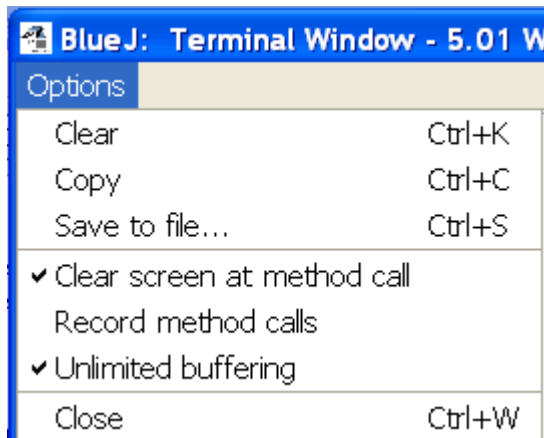
The default option for the BlueJ Terminal Window does not clear the screen before each new execution. Change these settings to start with a fresh uncluttered screen for each run of your program

Initially, the Terminal Window is set to display about 25 rows. If your output scrolls off the top of the screen, select Unlimited buffering from the Options menu.

2. Programmers often start loop counting variables at 0, but there are many times when you will need to start at a different number.

   - What happens if you change the initial value of the counter variable to a higher number, or a lower (negative) number?
   - What happens if you accidentally start the **counter** variable at a number larger than **theTerminator** variable?

3. Modify the program so that it counts backwards. What is the shortcut operator for decrement?

4. Modify the program so that it increments/decrements by a different integer (e.g., 2, 3, 15, etc.). You will no longer be able to use a shortcut increment operator, but what about the += or -= operators?

5. Change the **counter** and **theTerminator** variables to **doubles** and the values to decimals. Also, change the incrementation to a small decimal value (e.g., .5, .1, .01). Be aware that sometimes counting with decimals can lead to problems due to the way **double** values may be approximated.

6. What happens if you delete the curly braces around the block of code within the **while** loop?

7. What happens if you delete the parentheses around the condition in the **while** statement?

8. WARNING! You are about to create an infinite loop that will tie up your processor unless you are careful!

Select View on the menu bar and choose the Show Debugger option.

A large window for the BlueJ Debugger will appear on the screen.

In the bottom right corner there will be a big **red X** labeled Terminate.

If the Debugger is open and your program enters an infinite loop, simply click on the **red X** and the program will terminate.
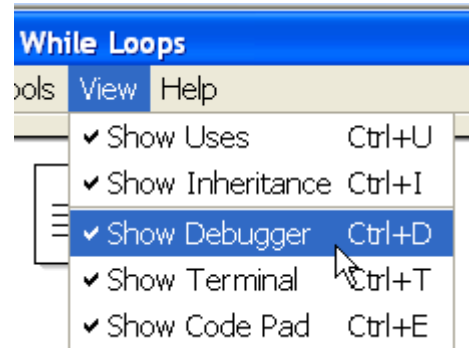
If you enter an infinite loop without the Debugger open, you may have to shut down BlueJ or your entire computer in order to regain control. Better to be safe than sorry when working with loops, so it is a good idea to always activate the Debugger.

9. With the Debugger open so you can see the big **red X**, delete line <11>. Run the program again. Why does deleting this line send the program into an infinite loop?