

Note: This lesson uses the instruction and activities from Introduction to Computer Science using Java by Bradley Kjell
https://chortle.ccsu.edu/Java5/Notes/chap15/ch15_2.html and
<https://curriculum.code.org/csp-18/unit5/11/> and it is modified for grade 7 students

1. Standards:

- 7-8.CT.9-Read and interpret code to predict the outcome of various programs that involve conditionals and repetition for the purposes of debugging.

2. **Lesson objective:** Analyze a while loop to determine if the initial condition will be met, how many times the loop will run, and if the loop will ever terminate.

3. **Aim:** I can write a while loop and diagram how it works.

4. **Warm up-time: 5 minutes** (think - pair - share)

- Questions
 - i. What are some activities that repeat over and over again?
 - ii. What are some modern machines that work by repeating the same motions?
- Reading - Match the cycle with machine

Word Bank: *Laundry Dryer* *Water Pump* *Sun and the Earth*
Clock TV Set *CD Player* *Bicycle*

_____ : your legs drive the pedals connected to a gear which spins.

_____ : the disk spins (cycles) as the laser moves across it.

_____ : pictures are put one the screen one after another as long as the set is on.

_____ : often a piston repeatedly moves in and out of a cylinder.

_____ : rotating drum.

_____ : shows the same times every day. If the clock is mechanical, its insides are gears and springs with many mechanical cycles.

_____ : endlessly cycling, seasons flowing one into the next.

5. Lesson Content - 10 minutes

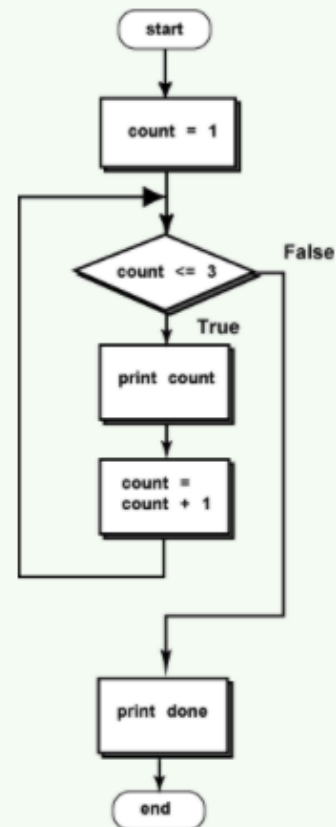
Introduction: Most modern machines work by endlessly repeating the same motions. The engine in your car cycles through the same motions over and over as it burns gasoline to provide power. Electric motors are similar. They convert electric power into circular motion. Because of their circular motions, these machines keep going as long as you want. Computer programs use cycles, also. In programming, a cycle is called a loop. When a program has a loop in it, some statements are done over and over as long as is needed to get the work done. Most computer programs execute many millions of program statements each time they are used. Usually the same statements are executed many times.

Model: Review the code and flowchart with students and [syntax](#) and [semantics](#) of while loop

```
// Example of a while loop
public class LoopExample
{
    public static void main (String[] args )
    {
        // start count out at one
        int count = 1;

        // loop while count is <= 3
        while ( count <= 3 )
        {
            System.out.println( "count is:" +
count );

            // add one to count
            count = count + 1;
        }
        System.out.println( "Done with the loop"
);
    }
}
```



The flowchart shows how the program works. First, `count` is set to one. Then it is tested by the `while` statement to see if it is less than or equal to three.

The test returns *true* so the statements in the block following the `while` are executed. The current value of `count` is printed, and `count` is incremented. Then execution goes back to the `while` statement and the test is performed again.

`count` is now two, the test returns *true* and the block is executed again. The last statement of the block increments `count` to three, then execution goes back to the `while` statement.

`count` is now three, the test returns *true* and the block is executed again. The last statement of the block increments `count` to four, then execution goes back to the `while` statement.

After the block has executed three times, `count` is four. Execution goes back to the `while` statement, but now the test returns *false*, and execution goes to the "Done with loop" statement. Then the program ends.

6. Lesson Activity 20 minutes

1. Copy the above program to a file and run it. Then play with it. See if you can change the program so it prints one through ten. Then change it so that it prints zero through ten.

2. What does this statement do: `count = count + 1;`

3. What will the program print if the **initialization** is changed as follows?

```
int count = 0;                // count is initialized
while ( count <= 3 )          // count is tested
{
    System.out.println( "count is:" + count );
    count = count + 1;         // count is changed
}

System.out.println( "Done with the loop" );
```

4. Look at [these flowcharts](#) and write how many times you believe the loop will run; explain your reasoning.

7. Closing - 5 minutes

Here is a [complete program](#) that lets the user pick the initial value and the limit value. Copy this program to a file and compile and run it. If the user sets the

initial value to -2 and the limit value to 1 what values will be printed out? Draw a flowchart to explain how the program runs.

8. Homework/Practice - write a simple program that uses a while loop

<https://docs.google.com/document/d/1XHW8WGth5YtVhkaFtTw7eWY1y653D4unbaG1-IHDeQ/edit>

Activity Guide - Flowcharts with `while` Loops



Introduction

We are going to be learning a new programming structure today called `while` loops. **`while` loops allow us to control program flow by repeating a set of commands until a condition is met.** When we control program flow it is often helpful to think about the ideas in a visual way first. You will use flowcharts to begin thinking about `while` loops.

Flowchart Components

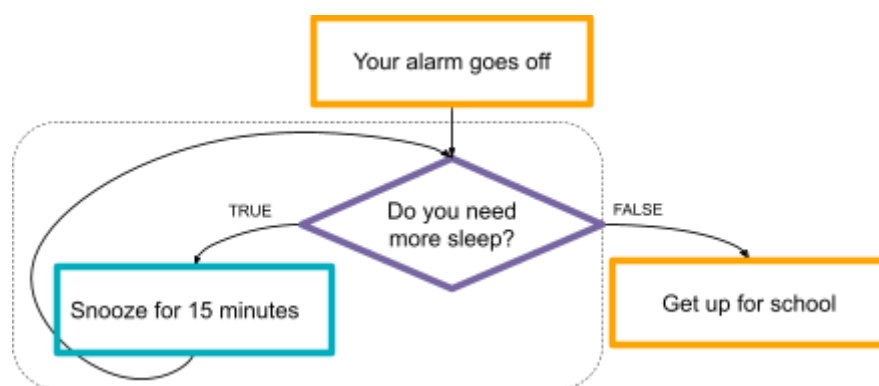
As a reminder we have included the flowchart components here as a reference for you as you work on this sheet.



Real-Life `while` Loops

Each flowchart below contains a “loop” that runs “while” a condition is true. Investigate each flowchart, mark how many times you believe the loop will run on the line provided, and then justify your reasoning.

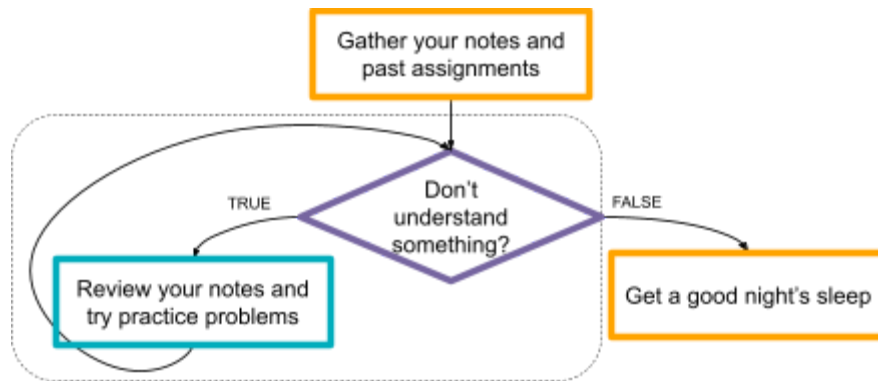
Note: There may be many possible answers, so pick the one you believe makes the most sense.



How many times will the loop run?



Justification: This is just one possible answer representing snoozing a few times. You could of course never snooze, but you would likely not need more sleep forever.



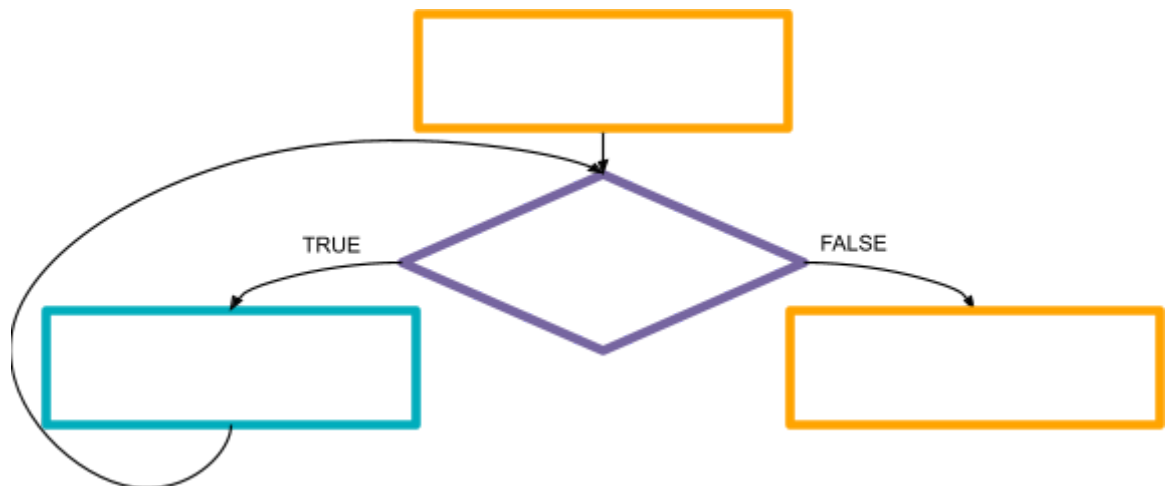
How many times will the loop run?

Never A few times Forever

Justification: As with the previous solution, so long as the justification matches the placement of the X, the answer is appropriate. "Never" would mean you understand everything. "Forever" would mean there are some concepts you never grasp.

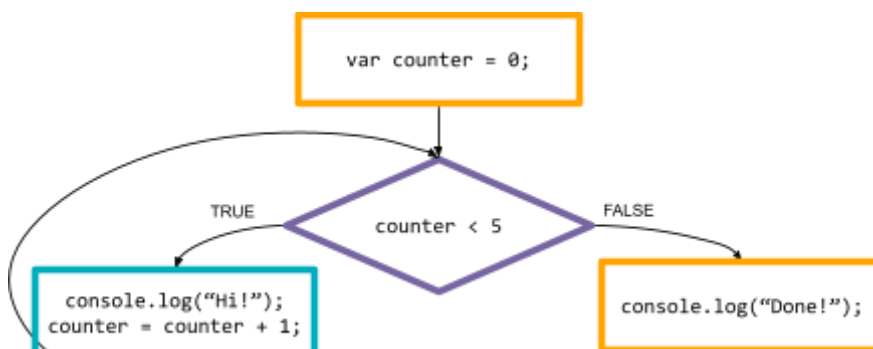
Complete this flowchart with a real-life `while` loop of your own choosing. Exchange with a partner and see how many times they think your loop would run.

Student responses will vary but should be similar in format to the ones shown above.



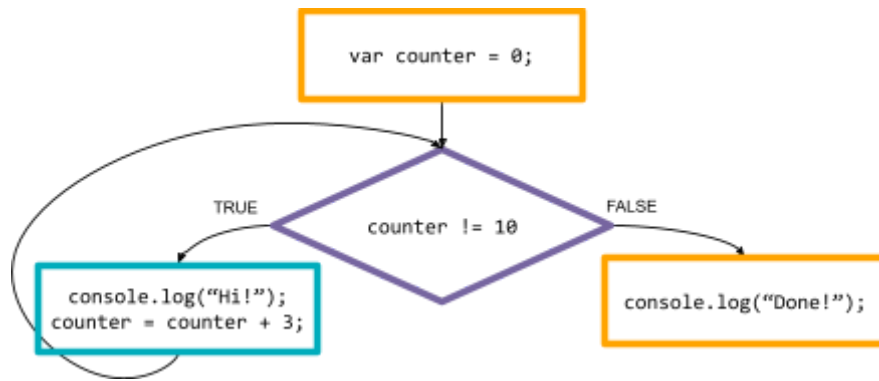
Programming with `while` Loops

The next two examples begin making the transition from real-life `while` loops to ones you might see while programming. Use the space provided to write the output that would be generated from the program.



Output:
Hi!
Hi!
Hi!

Hi!
Hi!
Done!



Output:

Hi!
Hi!
Hi!
Hi!

....

(This program produces
an infinite loop and so
would print "Hi!"
forever.)