Inspiring Greatness
ISD
INDEPENDENCE SCHOOL DISTRICT

Computer Science Virtual Learning

HS Computer Science A

May 5th, 2020

Lesson: While Loops

Objective/Learning Target:

Understanding what a While Loop is and the how to apply it using Java

What is a While Loop?

A while loop executes the body of the loop as long as (or while) a Boolean condition is true. When the condition is false execution continues after the body of the while loop.

If the condition is false the first time you check it, the body of the loop will not execute.

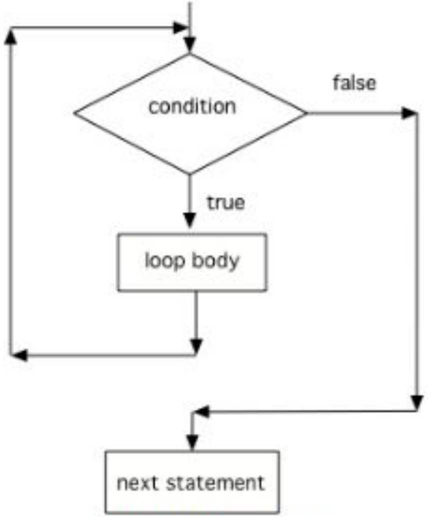
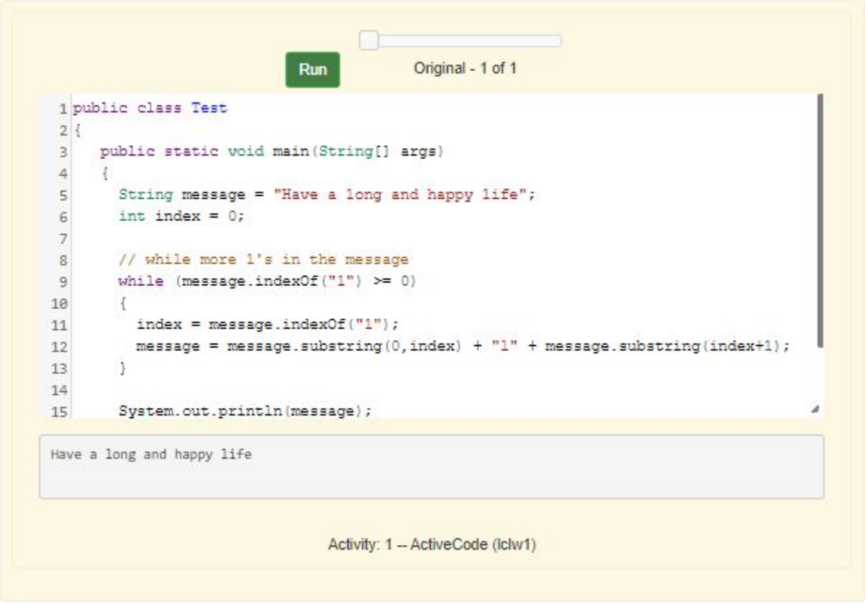


Figure 1: Flow in a while loop

What is a While Loop?

You can use a while loop to repeat the body of the loop a certain number of times as shown above. However, a while loop is typically used when you don’t know how many times the loop will execute. You might use it while processing a string. Google has been scanning old books and then using software to read the scanned text. But, the software can get things mixed up like using 1 for l. The following code loops through a string replacing all 1’s with l’s.



```java
public class Test
{
   public static void main(String[] args)
   {
     String message = "Have a long and happy life";
     int index = 0;

     // while more 1's in the message
     while (message.indexOf("1") >= 0)
     {
       index = message.indexOf("1");
       message = message.substring(0,index) + "l" + message.substring(index+1);
     }

     System.out.println(message);
```

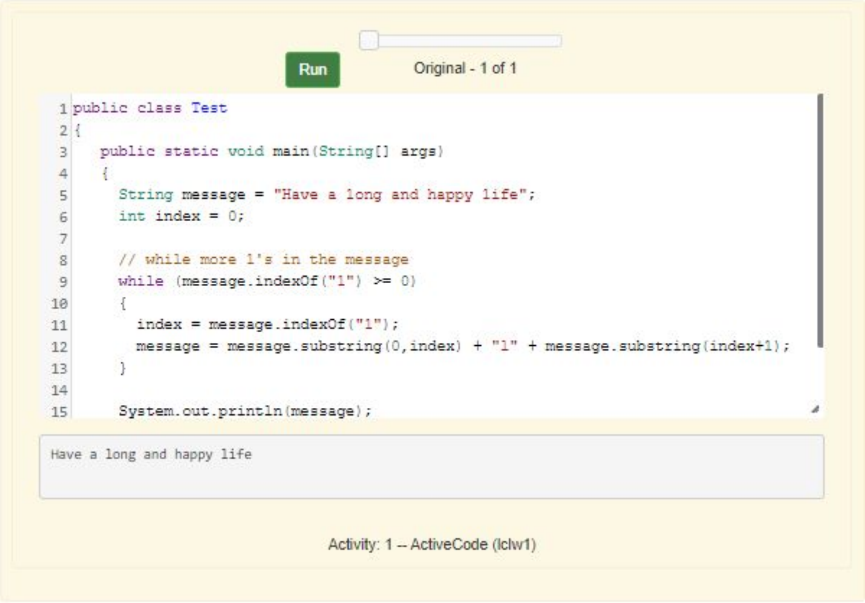
```
Have a long and happy life
```

Activity: 1 -- ActiveCode (lclw1)

While Loops

The while loop starts on line 9 in the code above. Statements 10 through 13 are the body of the loop (from the opening parenthesis on line 10 to the closing one on line 13).

Java doesn’t require your code to be correctly indented (code moved to the right a few spaces) to make it clear what statements are part of the body of the loop, but it is good practice. On the free response part of the exam, the reader will use the indention when determining the meaning of your code, even if you forget the open or close curly brace.



```
 1 public class Test
 2 {
 3    public static void main(String[] args)
 4    {
 5      String message = "Have a long and happy life";
 6      int index = 0;
 7
 8      // while more 1's in the message
 9      while (message.indexOf("1") >= 0)
10      {
11        index = message.indexOf("1");
12        message = message.substring(0,index) + "1" + message.substring(index+1);
13      }
14
15      System.out.println(message);
```

```
Have a long and happy life
```

Activity: 1 -- ActiveCode (lclw1)

While Loops

One thing to be careful about with while loops is making sure that you don’t end up with an infinite loop. An infinite loop is one that never stops (the condition is always true).

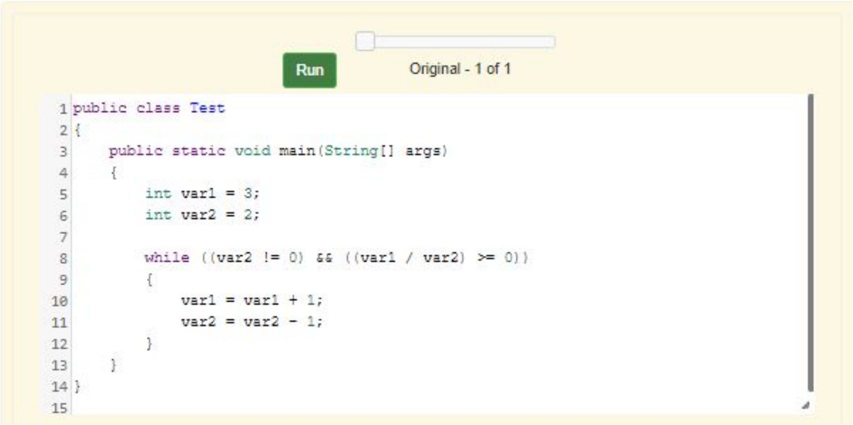
```
// an infinite loop
while (true)
{
        System.out.println("This is a loop that never ends");
}
```

The infinite loop above is pretty obvious. But, most infinite loops are accidental. They usually occur because you forget to change the thing you are checking in the condition.

Tracing Variables in Loops

A really important skill to develop is the ability to trace the values of variables and how they change during each time through a loop.

Here is a complex loop. See if you can trace the code on paper to predict what it will do when you run it.

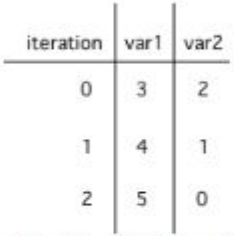


```
public class Test
{
    public static void main(String[] args)
    {
        int var1 = 3;
        int var2 = 2;

        while ((var2 != 0) && ((var1 / var2) >= 0))
        {
            var1 = var1 + 1;
            var2 = var2 - 1;
        }
    }
}
```

Tracing Variables in Loops

You can create a table that keeps track of the variable values each time through the loop as shown below. This is very helpful on the exam. Studies have shown that students who create tables like this do much better on code tracing problems on multiple choice exams.

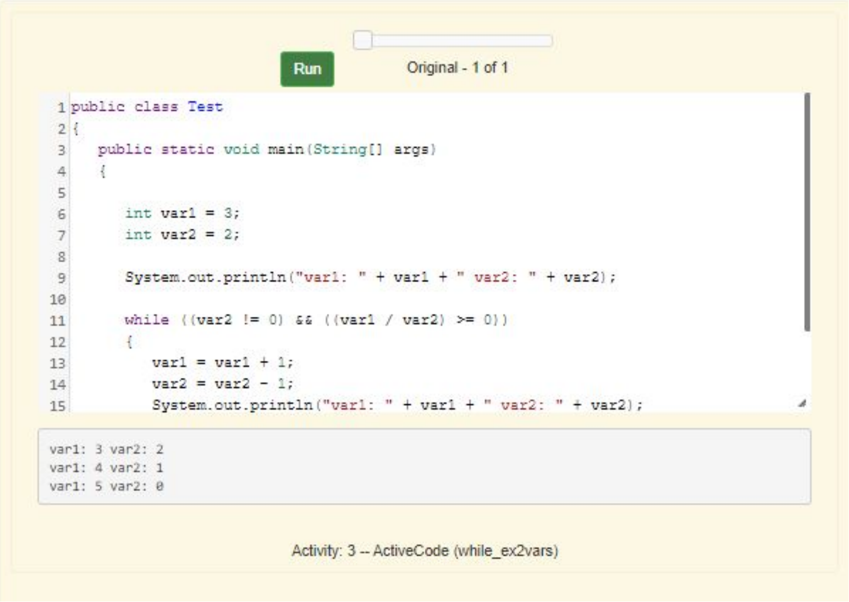


iteration	var1	var2
0	3	2
1	4	1
2	5	0

Figure 1: A table showing the values of all of the variables each time through the loop. The 0 means before the first loop.

Tracing Variables in Loops

You can also add `System.out.println(variable)` to print the value of a variable. In the code below I am printing the values of all of the variables before the loop and at the end of the loop body.



```
public class Test
{
    public static void main(String[] args)
    {

        int var1 = 3;
        int var2 = 2;

        System.out.println("var1: " + var1 + " var2: " + var2);

        while ((var2 != 0) && ((var1 / var2) >= 0))
        {
            var1 = var1 + 1;
            var2 = var2 - 1;
            System.out.println("var1: " + var1 + " var2: " + var2);
```

```
var1: 3 var2: 2
var1: 4 var2: 1
var1: 5 var2: 0
```

Activity: 3 -- ActiveCode (while_ex2vars)

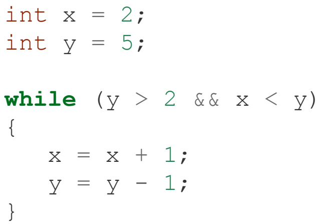
Check Your Understanding

1. What does the following code print?

```
int x = -5;
while (x < 0)
{
  x++;
  System.out.print(x + " ");
}
```

A. 5 4 3 2 1
B. -5 -4 -3 -2 -1
C. -4 -3 -2 -1 0

2. What are the values of x and y when the code finishes executing?



```
int x = 2;
int y = 5;

while (y > 2 && x < y)
{
    x = x + 1;
    y = y - 1;
}
```

A. x = 5, y = 2
B. x = 5, y = 2
C. x = 3, y = 4
D. x = 4, y = 3
E. x = 2, y = 5

3. What are the values of var1 and var2 when the code finishes executing?

```
int var1 = 0;
int var2 = 2;

while ((var2 != 0) && ((var1 / var2) >= 0))
{
  var1 = var1 + 1;
  var2 = var2 -1;
}
```

A. var1 = 1, var2 = 1
B. var1 = 2, var2 = 0
C. var1 = 3, var2 = -1
D. var1 = 0, var2 = 2
E. The loop will cause a run-time error with a division by zero

For More Resources and to Check Answers

Go to: https://runestone.academy/runestone/books/published/apcsareview/LoopBasics/lWhile.html