Computer Science Virtual Learning

HS Computer Science A

May 4th, 2020

Lesson: What is a Loop?

Objective/Learning Target:

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

What is a Loop?

When you play a song, you can set it to **loop**, which means that when it reaches the end it starts over at the beginning. A **loop** in programming is a way to repeat one or more statements. If you didn't have loops to allow you to repeat code, your programs would get very long very quickly!

The keywords `while` or `for` both indicate the start of a loop (the header or declaration). The **body of the loop** will be repeated while the loop condition is true.

The **body of the loop** is either a single statement following the `while` or `for` or a block of statements after an opening curly brace `{` and before a closing curly brace `}`.

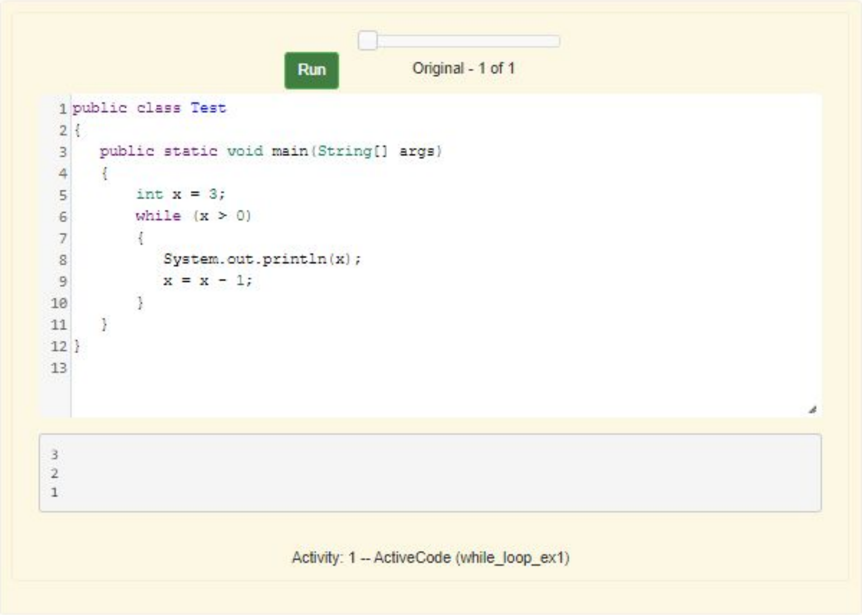
What is a Loop?

There are many different types of loops in Java, but the AP CS A exam only covers three:

- `while`: repeats the body of the loop while a Boolean expression is true
- `for`: contains a header with 3 possible parts: declaration/initialization, condition, and change. Before the loop starts it does the declaration/initialization. Then it repeats the body of the loop while the condition is true. The code in the change part is executed each time at the end of the body of the loop.
- for-each: loop through a collection (list or array) and each time through the loop set a variable to the next item from the collection. We will discuss this in the section about arrays.

While Loops

Here is an example `while` loop that just prints the numbers until 0 is reached. How would you modify it to print 0 too?



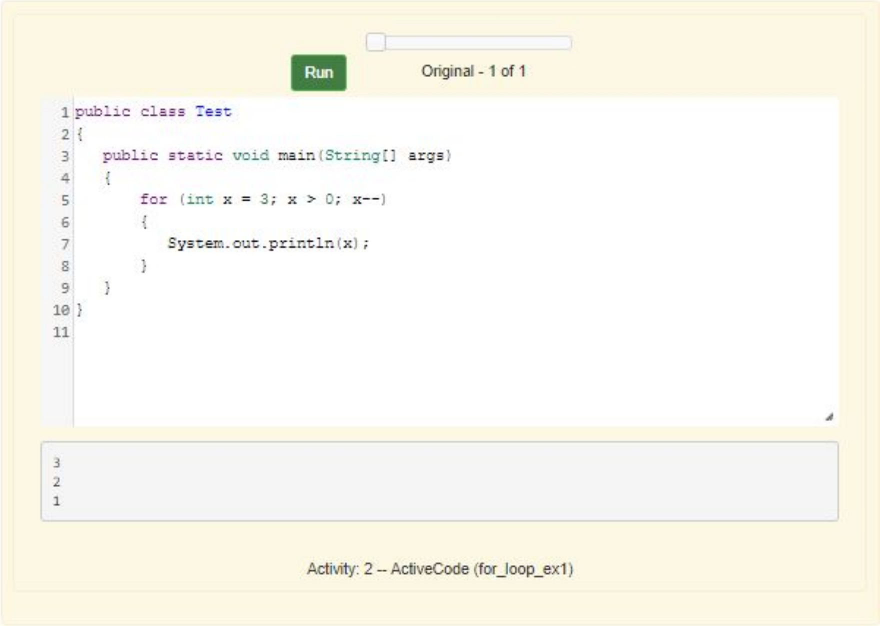
```
public class Test
{
   public static void main(String[] args)
   {
       int x = 3;
       while (x > 0)
       {
          System.out.println(x);
          x = x - 1;
       }
   }
}
```

```
3
2
1
```

Activity: 1 -- ActiveCode (while_loop_ex1)

For Loops

Here is an example for loop that just prints the numbers until 0 is reached. How would you modify it to print 0 too?

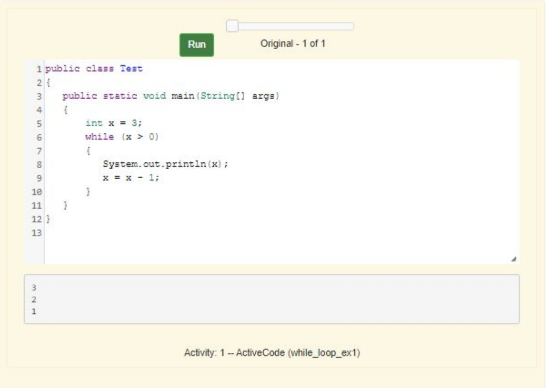


```java
public class Test
{
   public static void main(String[] args)
   {
      for (int x = 3; x > 0; x--)
      {
         System.out.println(x);
      }
   }
}
```

```
3
2
1
```

Activity: 2 -- ActiveCode (for_loop_ex1)

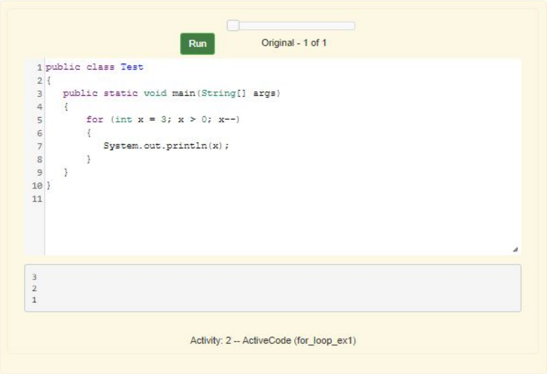
Which of the two loops takes less code? While you can write any loop with either a `while` or `for`, programmers tend to use the `while` when they don’t know how many times the loop will execute and the `for` when they know the number of times to execute the loop. The problem with using a `while` loop to execute a loop a certain number of times is that you have to remember to update the variable in the loop. The `for` loop allows you to specify all of the important things about a loop in one place (what value do variables start at, what is the condition to test, and how the loop variables change).



```java
public class Test
{
  public static void main(String[] args)
  {
      int x = 3;
      while (x > 0)
      {
         System.out.println(x);
         x = x - 1;
      }
  }
}
```

```
3
2
1
```

Activity: 1 -- ActiveCode (while_loop_ex1)



```java
public class Test
{
  public static void main(String[] args)
  {
      for (int x = 3; x > 0; x--)
      {
         System.out.println(x);
      }
  }
}
```

```
3
2
1
```

Activity: 2 -- ActiveCode (for_loop_ex1)

Check Your Understanding

1. Highlight all the statements that are part of the body of the while loop.

```
int x = 5;
while (x > 0)
{
System.out.println(x);
 x = x - 1;
}
```

2. Highlight all the statements that are part of the body of the for loop. If you make a mistake you can click on the statement again to unhighlight it.

```
for (int x = 5; x > 0; x--)
System.out.println(x);
```

3. Highlight all the statements that are part of the body of the for loop. If you make a mistake you can click on the statement again to unhighlight it.

```
String message1 = "I ";
String message2a = "love ";
String message3 = "you";
String message2b = "miss ";
for (int x = 1; x < 4; x++)
{
       System.out.println(message1 + message2a +
   message3);
       System.out.println(message1 + message2b +
   message3);
}
```

For More Resources and to Check Answers

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