Image Credit: Julius Schorzman

Java Lesson 6

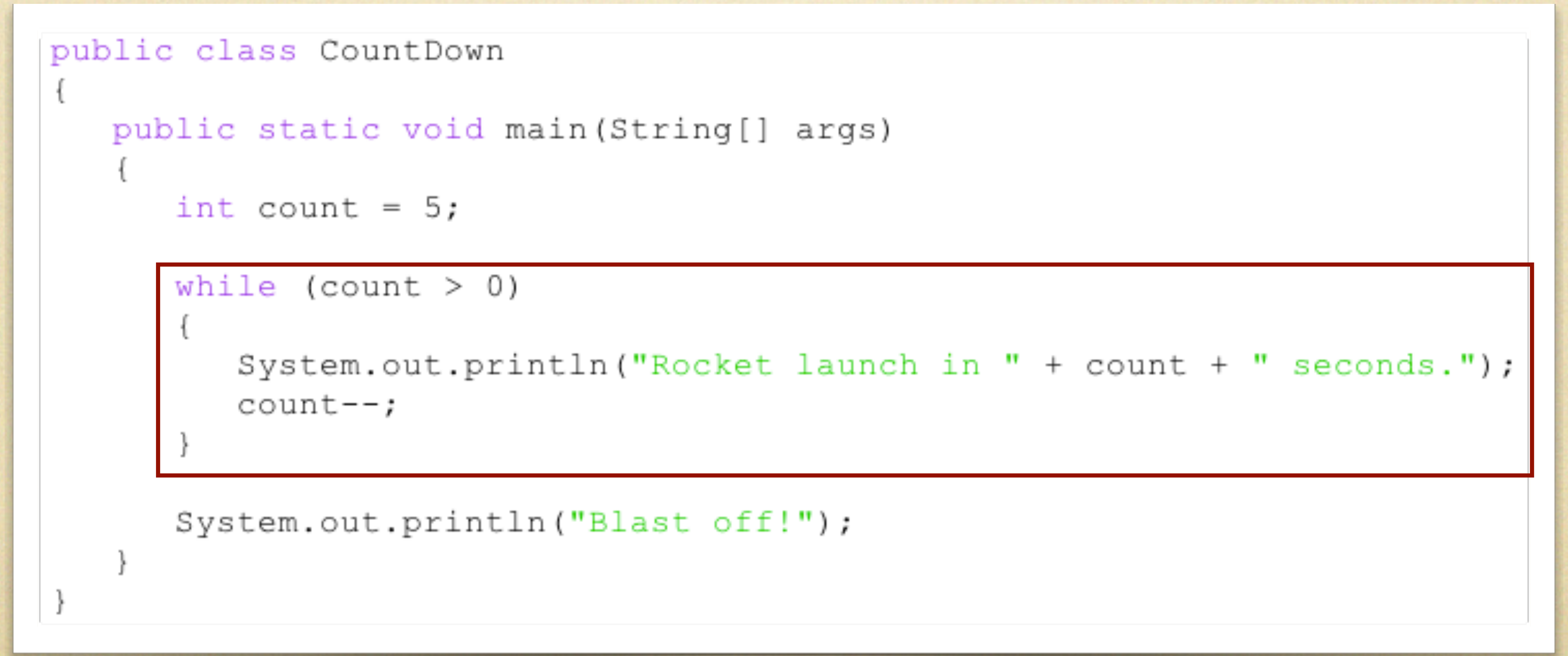
Concept Review

Loops

- A loop is a structure that allows for a block of code to be repeated multiple times, as desired.
- Some of the most common looping structures are:
    - while loop
    - for loop
    - do...while loop

The while loop

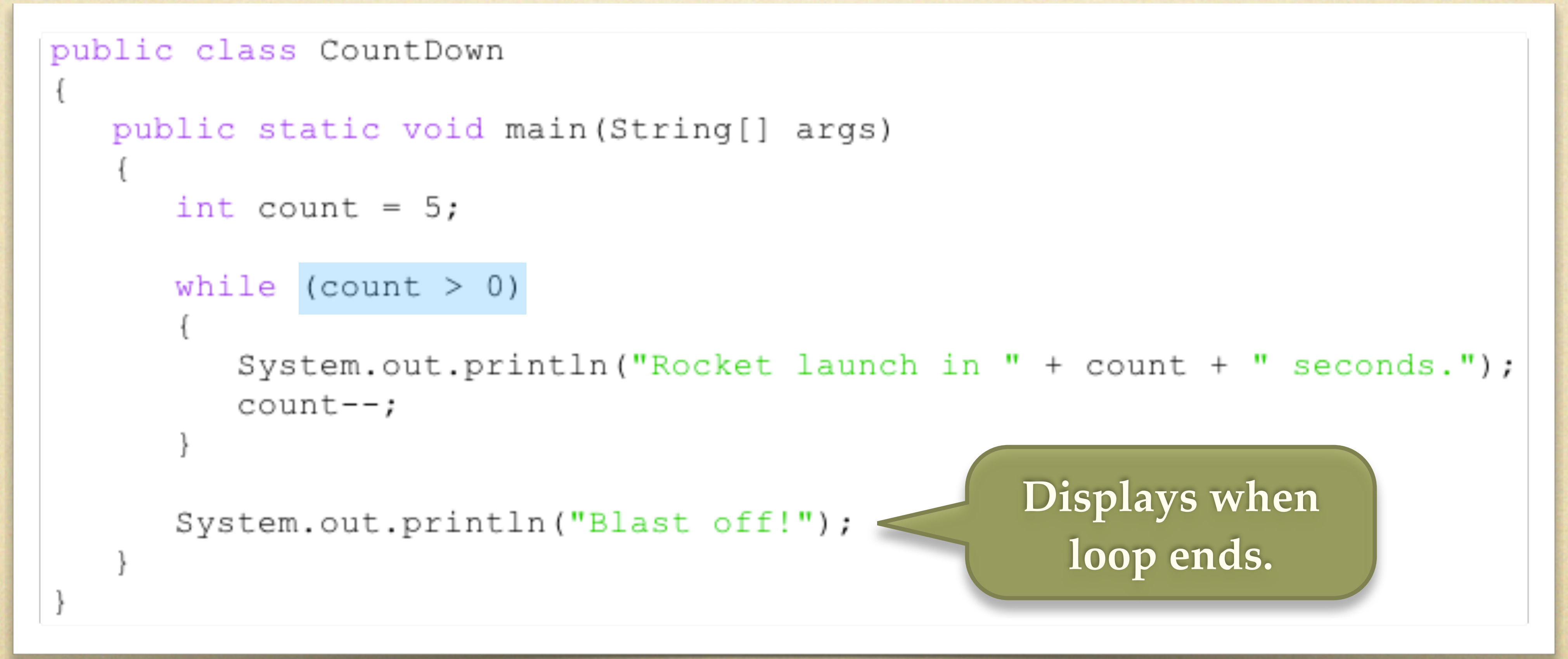
- The while loop is commonly used in Java, as it allows you to cause a block of code to keep repeating *while* a specified condition is true.
- Here's an example:



```
public class CountDown
{
    public static void main(String[] args)
    {
        int count = 5;

        while (count > 0)
        {
            System.out.println("Rocket launch in " + count + " seconds.");
            count--;
        }

        System.out.println("Blast off!");
    }
}
```

The while loop

- Notice that the loop keeps repeating *while* count is greater than 0. The moment that this conditions is no longer true, the loop exits and the program proceeds to print “Blast off!” onto the screen.

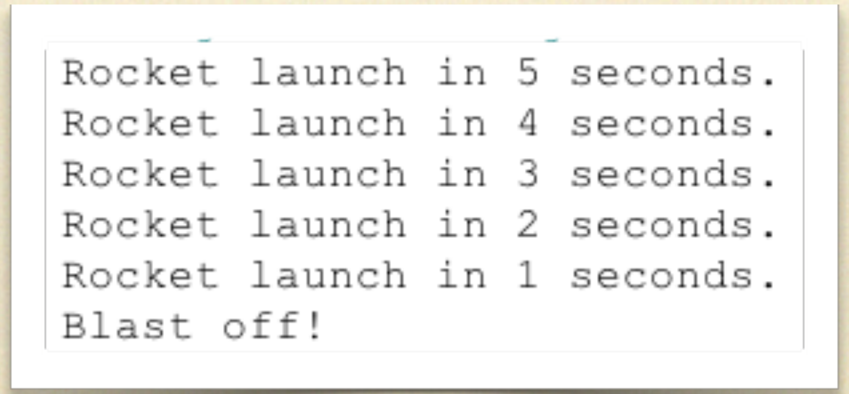


```java
public class CountDown
{
   public static void main(String[] args)
   {
      int count = 5;

      while (count > 0)
      {
         System.out.println("Rocket launch in " + count + " seconds.");
         count--;
      }

      System.out.println("Blast off!");
   }
}
```

Displays when loop ends.

The while loop

- When you run this program, you'll see the following output:



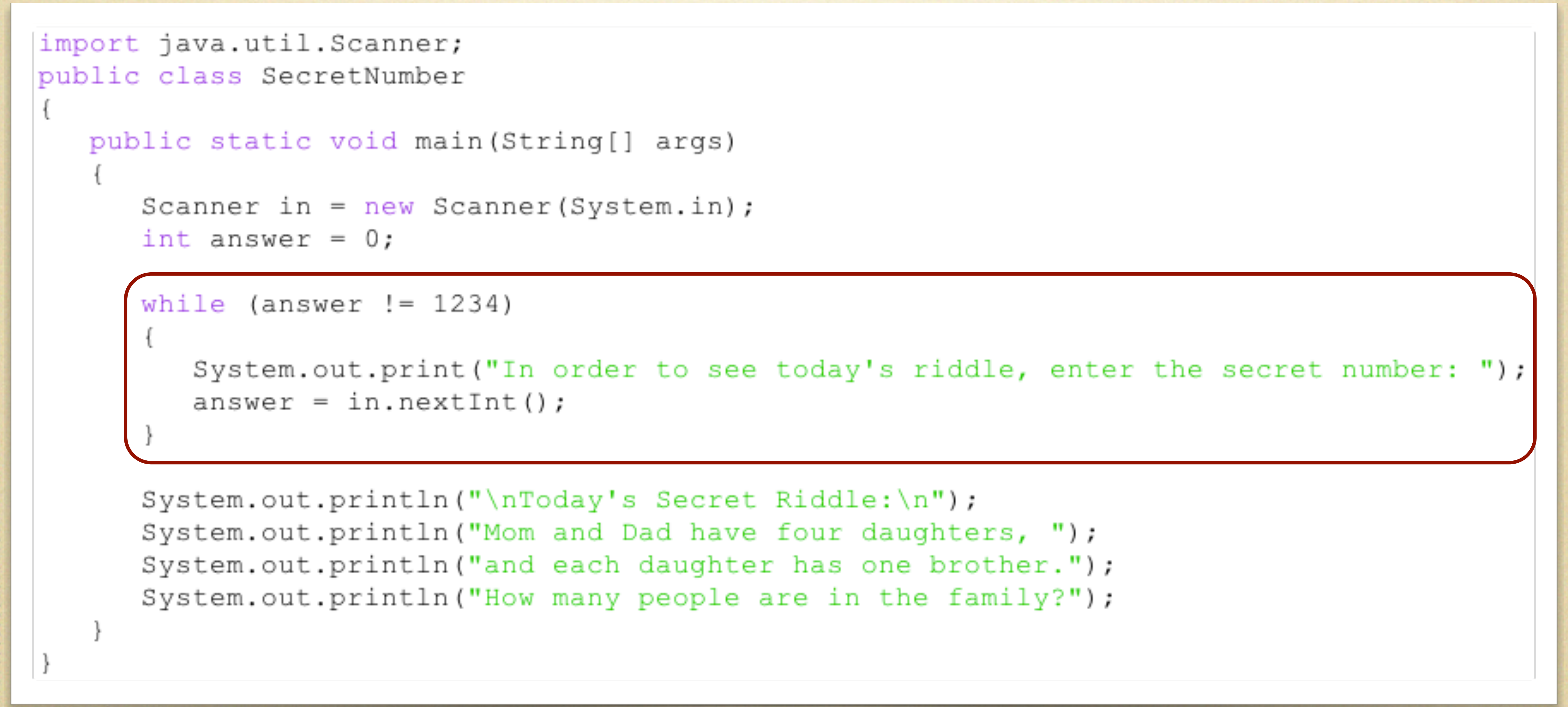
```
Rocket launch in 5 seconds.
Rocket launch in 4 seconds.
Rocket launch in 3 seconds.
Rocket launch in 2 seconds.
Rocket launch in 1 seconds.
Blast off!
```

- Remember, while loops will keep repeating their blocks of code until the condition eventually is false.

Indefinite Loop

- Say you want to create a program that asks the user for the secret password before continuing. Well, this is very easy to do with an indefinite loop!
- An indefinite loop is simply a term that refers to the idea that a loop can keep repeating for an unknown amount of times (until the user eventually enters the correct password!)
- On the next slide, I've provided an example of this.

Indefinite Loop



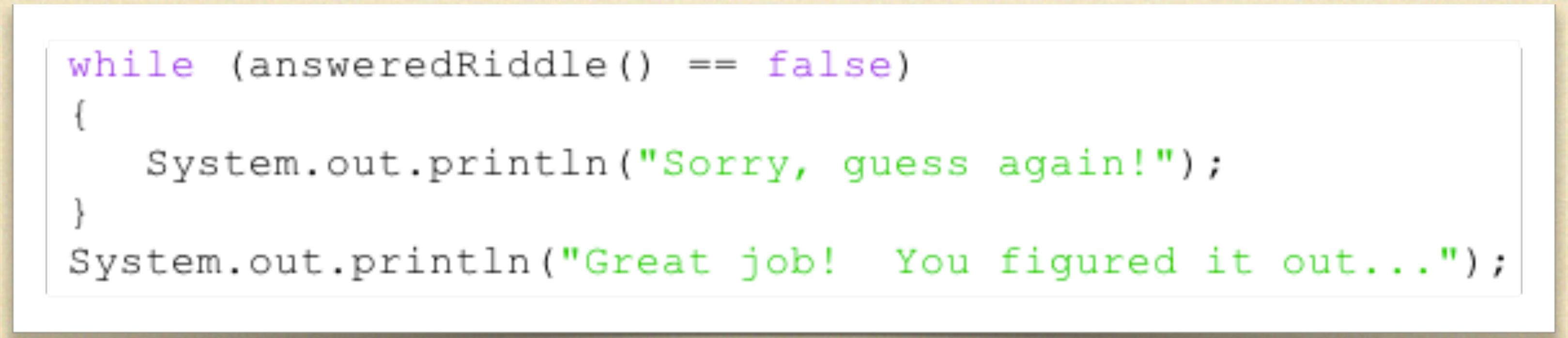
```
import java.util.Scanner;
public class SecretNumber
{
   public static void main(String[] args)
   {
      Scanner in = new Scanner(System.in);
      int answer = 0;

      while (answer != 1234)
      {
         System.out.print("In order to see today's riddle, enter the secret number: ");
         answer = in.nextInt();
      }

      System.out.println("\nToday's Secret Riddle:\n");
      System.out.println("Mom and Dad have four daughters, ");
      System.out.println("and each daughter has one brother.");
      System.out.println("How many people are in the family?");
   }
}
```

- Notice that the loop keeps repeating until the user types in the correct answer! Any guesses as to the answer of the riddle?

Optional Learning

- One common technique that's used with *while* loops is to place a method inside the loop's condition that will either return a boolean value of either *true* or *false*.

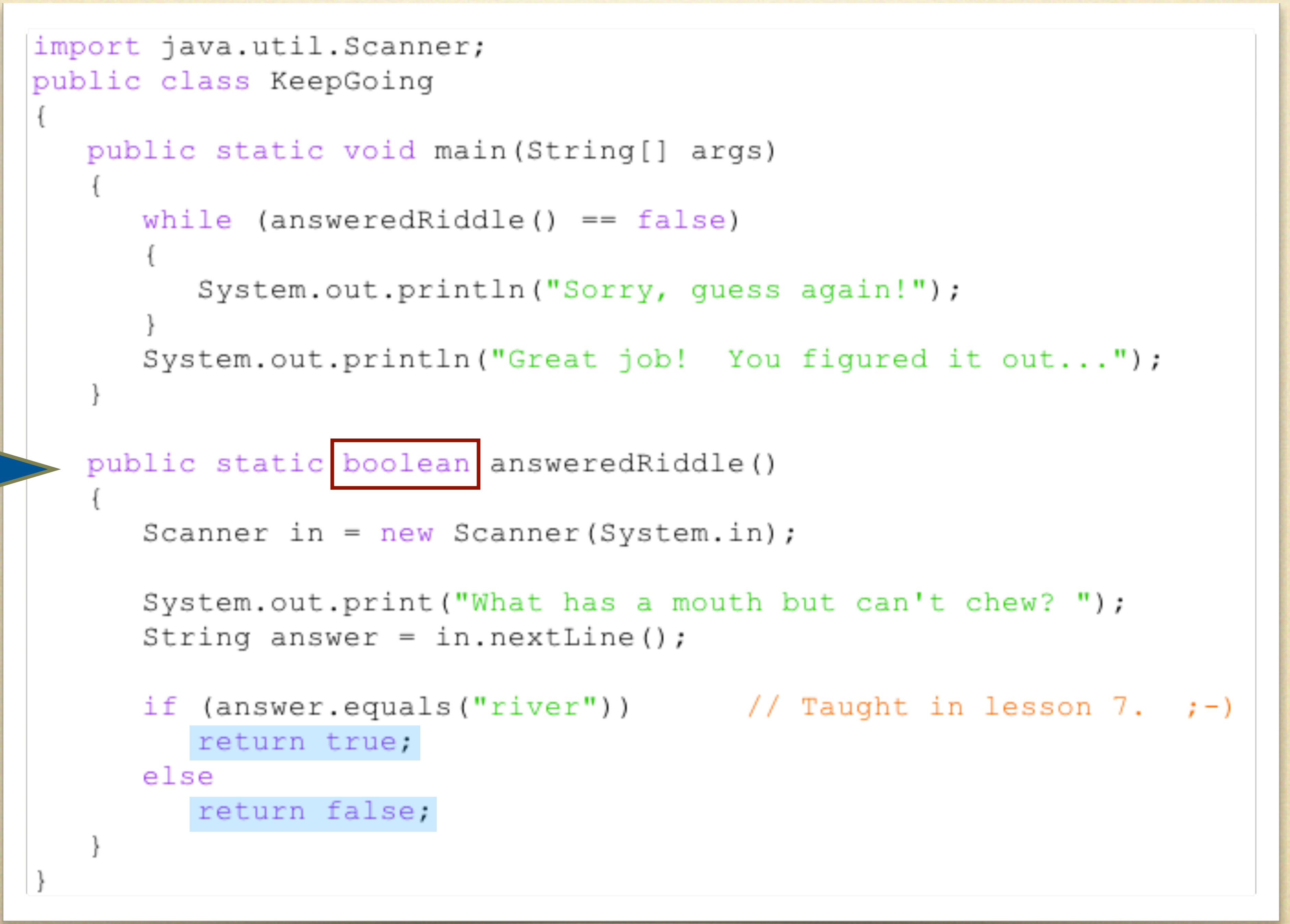
- Consider the following example.



```
while (answeredRiddle() == false)
{
   System.out.println("Sorry, guess again!");
}
System.out.println("Great job!  You figured it out...");
```

- This code piece calls the answeredRiddle() method over and over again until the user finally guesses the correct answer to the riddle.

Optional Learning



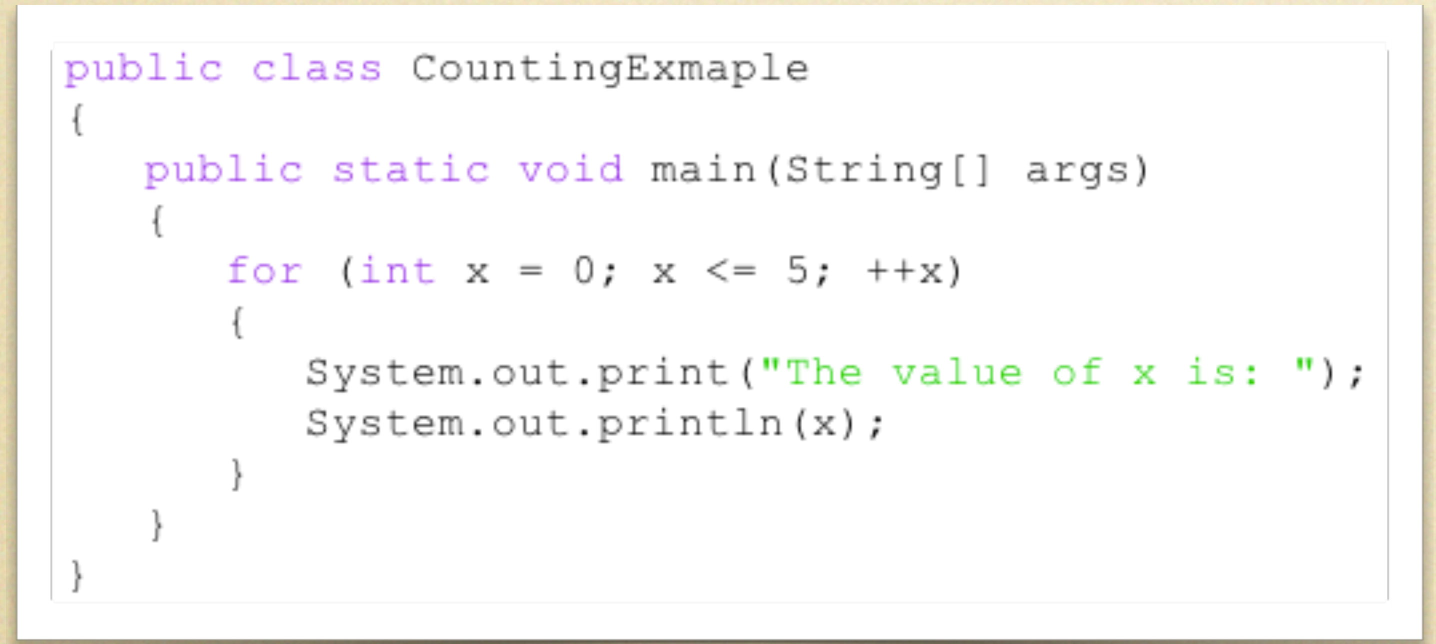
```
import java.util.Scanner;
public class KeepGoing
{
   public static void main(String[] args)
   {
      while (answeredRiddle() == false)
      {
         System.out.println("Sorry, guess again!");
      }
      System.out.println("Great job!  You figured it out...");
   }

   public static boolean answeredRiddle()
   {
      Scanner in = new Scanner(System.in);

      System.out.print("What has a mouth but can't chew? ");
      String answer = in.nextLine();

      if (answer.equals("river"))       // Taught in lesson 7.  ;-)
         return true;
      else
         return false;
   }
}
```

Asks the user for the answer to the riddle.

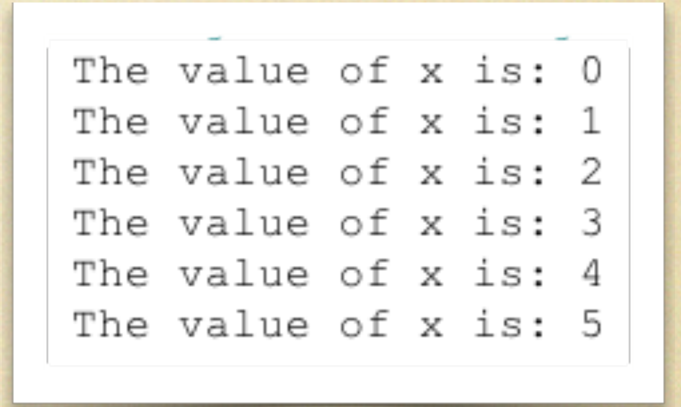
If correct, return as true. If not, returns as false.

for loops

- *for* loops are used when a definite number of looping iterations need to happen.
- Say that you wanted to create a program that could count from 0 through 5. Here's an example:



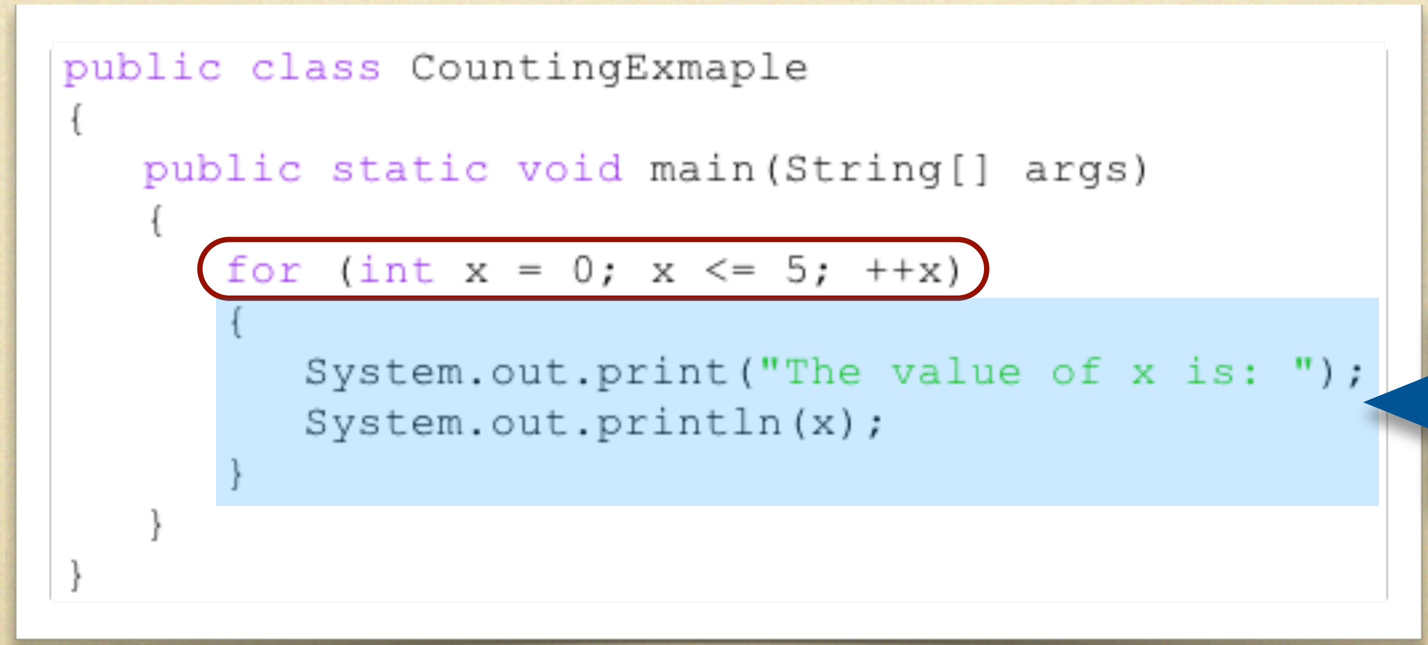
```
public class CountingExmaple
{
    public static void main(String[] args)
    {
        for (int x = 0; x <= 5; ++x)
        {
            System.out.print("The value of x is: ");
            System.out.println(x);
        }
    }
}
```



```
The value of x is: 0
The value of x is: 1
The value of x is: 2
The value of x is: 3
The value of x is: 4
The value of x is: 5
```

for loops

- Notice the for loop in the code below. There are three key parts to this loop structure, which are:
  - To create the variable *x* and assign it a value.
  - The condition, which is to keep looping while *x* <= 5.
  - To increment the value of *x* by 1.

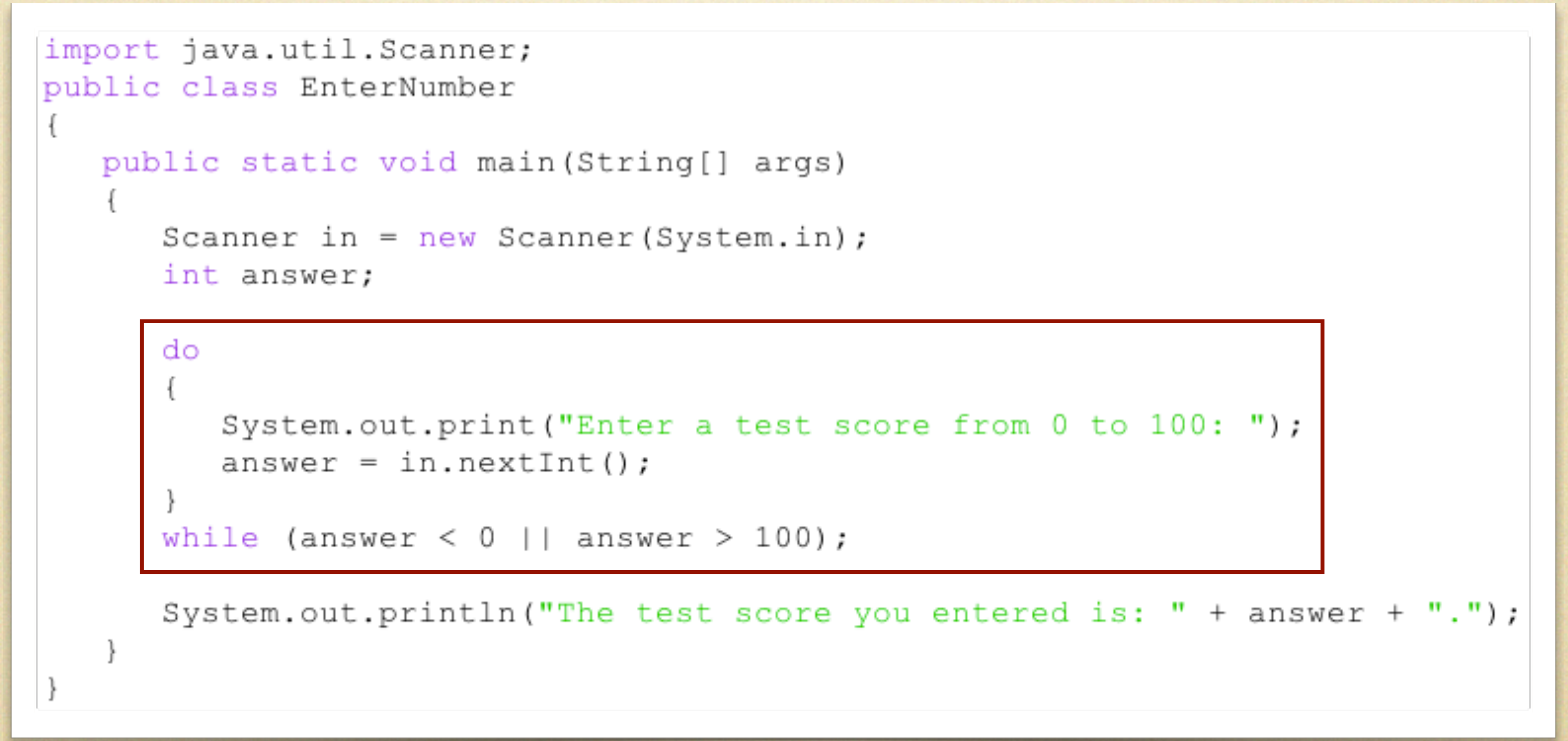


```
public class CountingExmaple
{
   public static void main(String[] args)
   {
      for (int x = 0; x <= 5; ++x)
      {
         System.out.print("The value of x is: ");
         System.out.println(x);
      }
   }
}
```

Code that repeats in the *for* loop.

do…while loops

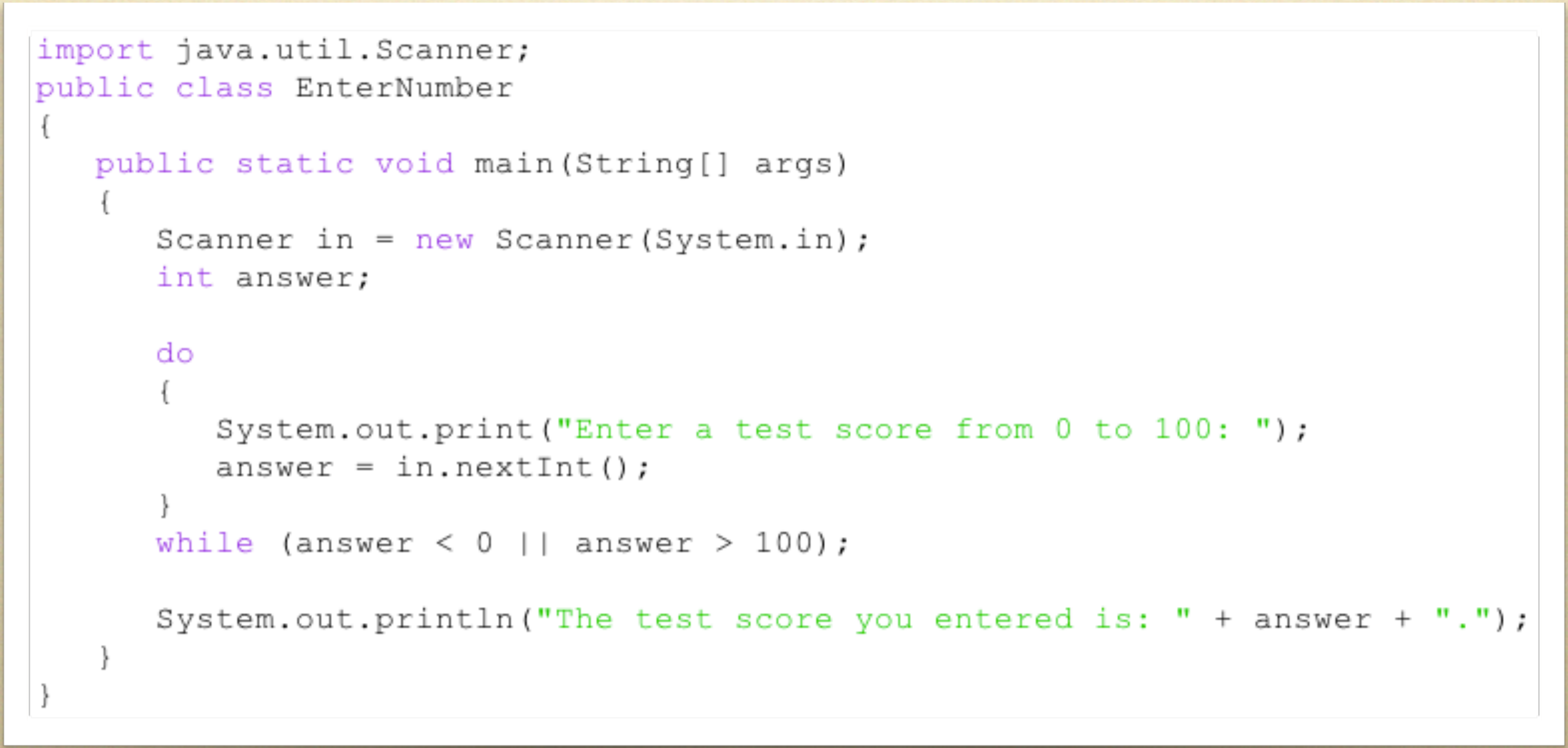
- *do…while* loops work great when you need to ask a user for information! Consider the following example:



```
import java.util.Scanner;
public class EnterNumber
{
   public static void main(String[] args)
   {
      Scanner in = new Scanner(System.in);
      int answer;

      do
      {
         System.out.print("Enter a test score from 0 to 100: ");
         answer = in.nextInt();
      }
      while (answer < 0 || answer > 100);

      System.out.println("The test score you entered is: " + answer + ".");
   }
}
```

do…while loops

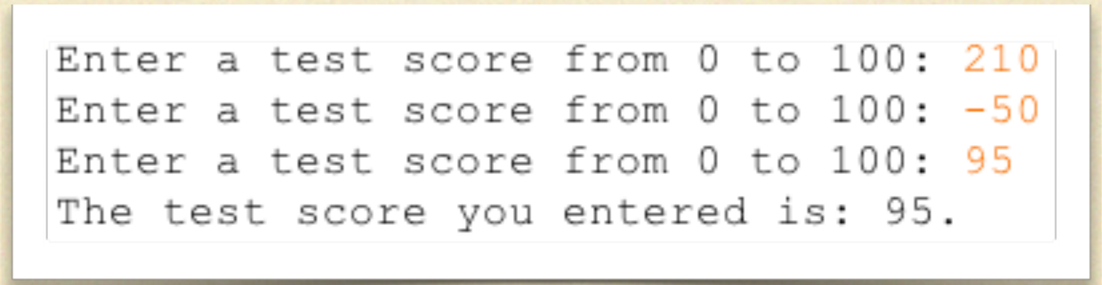
- In this example, the user is asked to enter a test score from 0 to 100.
- If the user enters a score outside of this range, the do...while loop will cause it to keep asking for the score until a valid answer is provided.



```java
import java.util.Scanner;
public class EnterNumber
{
   public static void main(String[] args)
   {
      Scanner in = new Scanner(System.in);
      int answer;

      do
      {
         System.out.print("Enter a test score from 0 to 100: ");
         answer = in.nextInt();
      }
      while (answer < 0 || answer > 100);

      System.out.println("The test score you entered is: " + answer + ".");
   }
}
```

do…while loops

- When the program is run, you can see that the do…while loop keeps displaying the question repeatedly to the user until a valid response is eventually provided.



```
Enter a test score from 0 to 100: 210
Enter a test score from 0 to 100: -50
Enter a test score from 0 to 100: 95
The test score you entered is: 95.
```