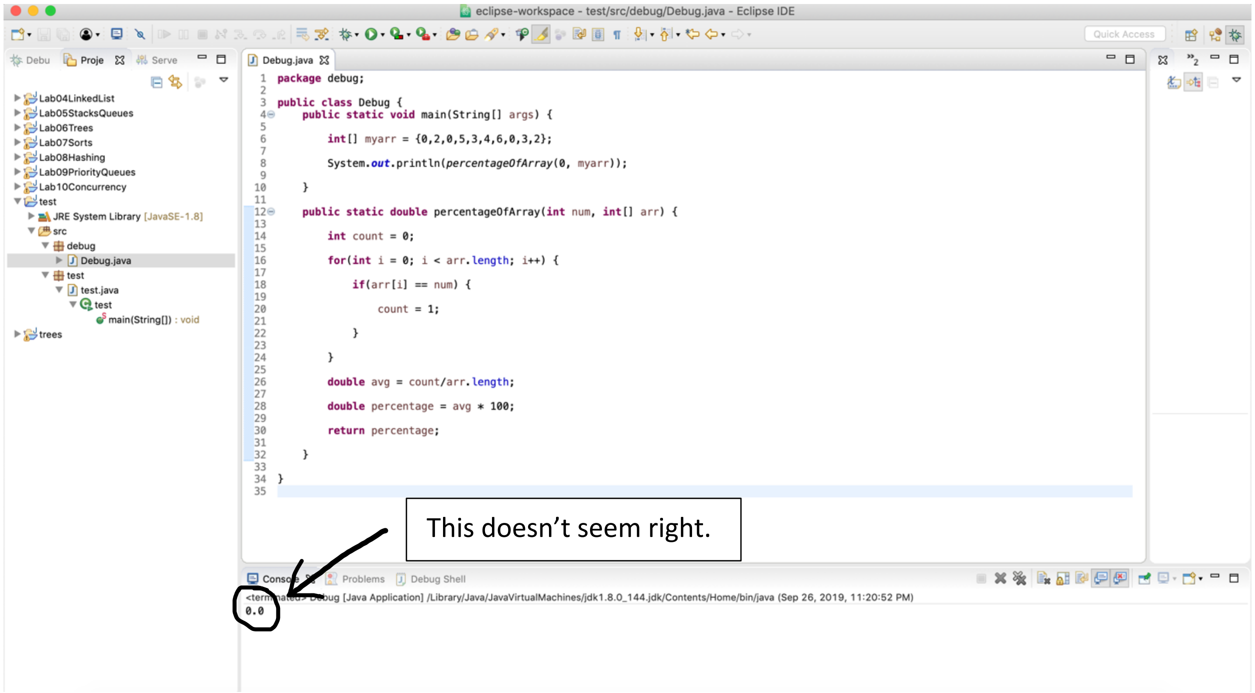
Eclipse Debugger Tutorial

Created by Cassandra Quach for DSA1 with Floryan



```java
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count = 1;

            }

        }

        double avg = count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

}
```

This doesn’t seem right.

```
<terminated> Debug [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Sep 26, 2019, 11:20:52 PM)
0.0
```

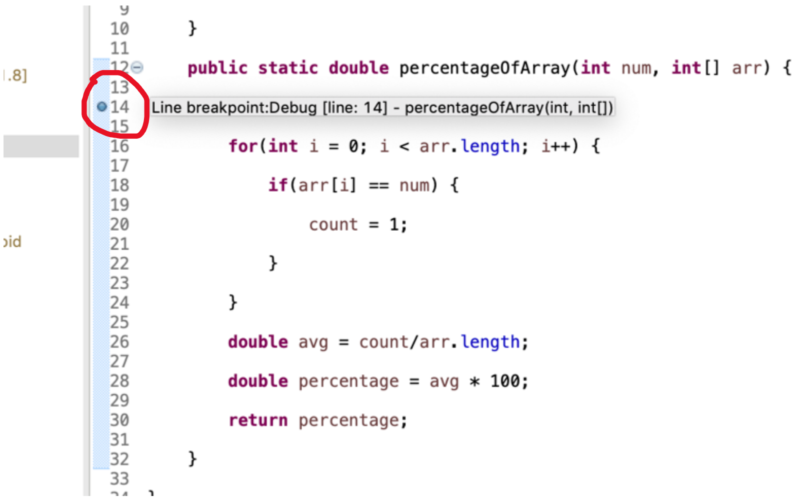
The method percentageOfArray() is given two parameters: an integer and an array of integers.

It should return the percentage of occurrences of that integer in the given array.

The number 0.0 is printed to the console when 30.0 should be printed instead since 30% of the array is filled with “0”.

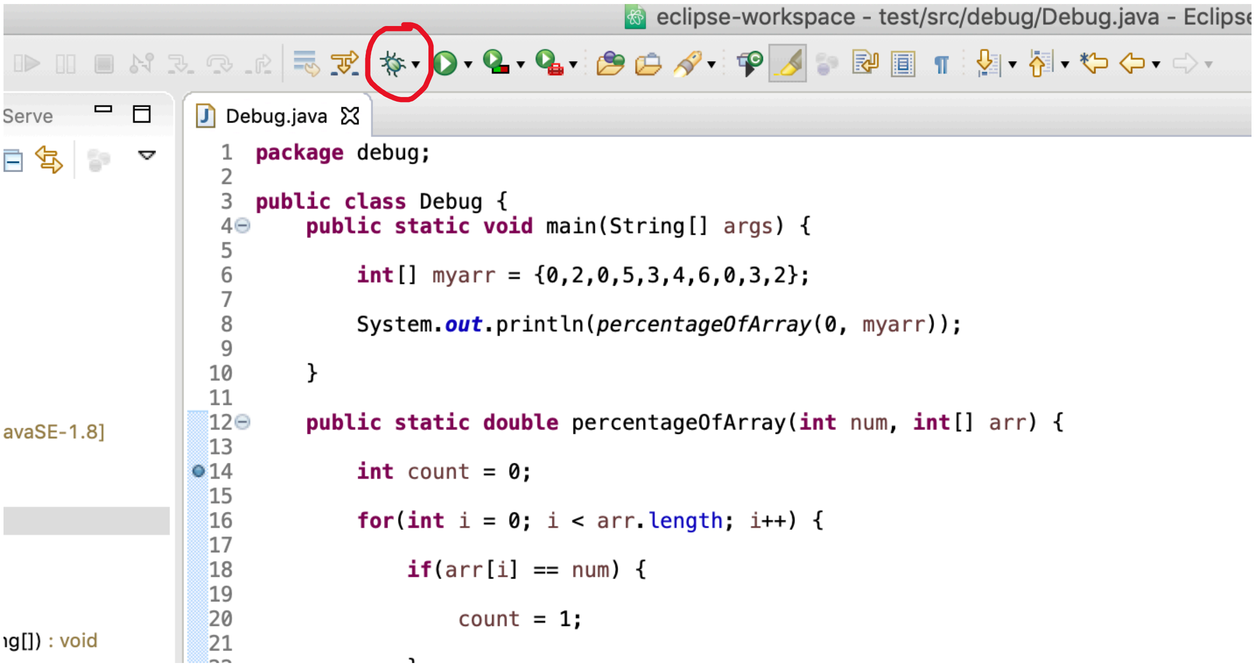
How do we use Eclipse to help find the problem? Use the built-in debugger tool!

Step 1: Set a breakpoint by double clicking to the left of the line number you want the program to stop (remove breakpoints by double clicking on them; this toggles it on and off)



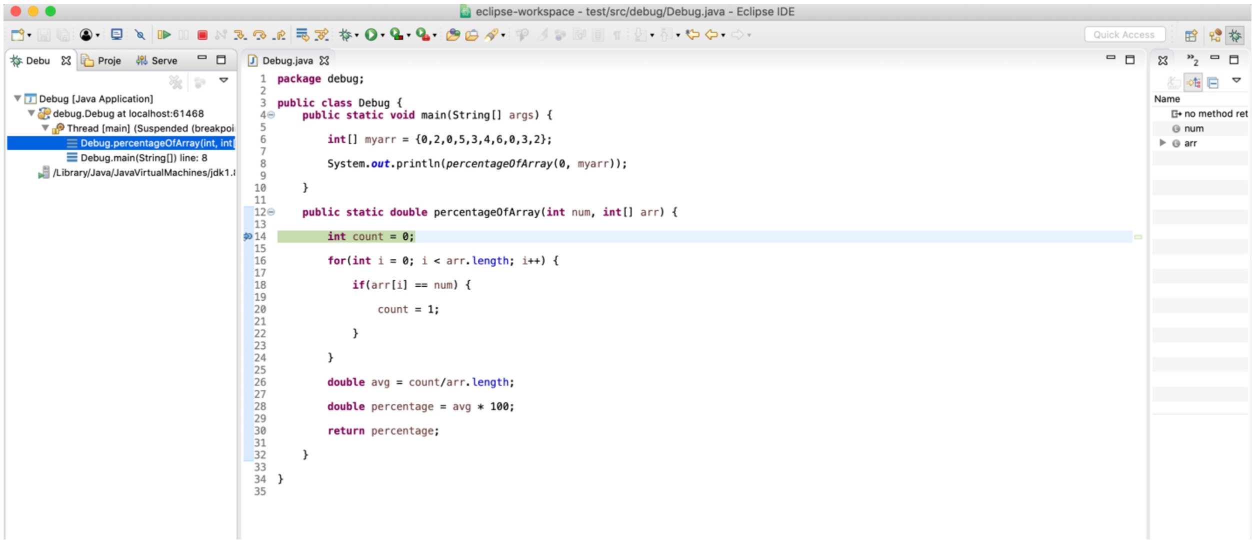
```java
    }

    public static double percentageOfArray(int num, int[] arr) {

Line breakpoint:Debug [line: 14] - percentageOfArray(int, int[])

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count = 1;

            }

        }

        double avg = count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }
```

Step 2: Click the little bug button next to the run button



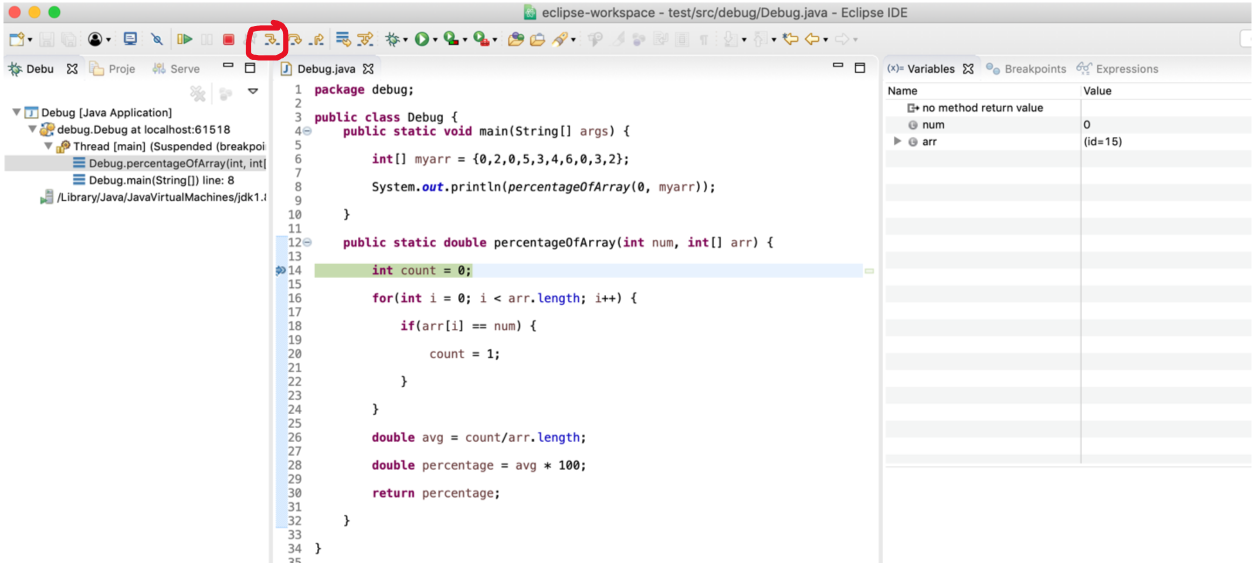
```
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count = 1;

```

Once you click the button your window should look like this, where the line you set the breakpoint at is highlighted in green and the right most window/view lists the variables available to you within the method; here we have num and arr. You can drag the bar separating the views to the left to expand it:


eclipse-workspace - test/src/debug/Debug.java - Eclipse IDE
Quick Access
Debu
Proje
Serve
Debug.java
Debug [Java Application]
debug.Debug at localhost:61468
Thread [main] (Suspended (breakpoi
Debug.percentageOfArray(int, int[
Debug.main(String[]) line: 8
/Library/Java/JavaVirtualMachines/jdk1.
Name
no method ret
num
arr

```
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count = 1;

            }

        }

        double avg = count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

}

```

Step 3: Click the “Step Into” button circled in the image below to move the program forward by one line:



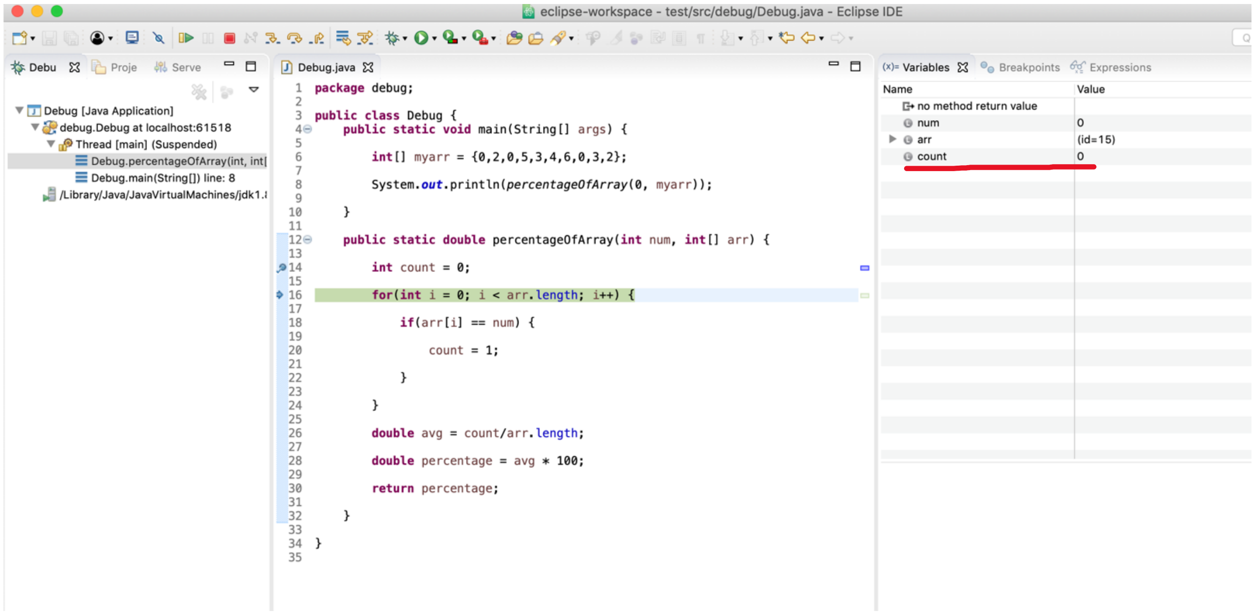
eclipse-workspace - test/src/debug/Debug.java - Eclipse IDE

```
Debug [Java Application]
  debug.Debug at localhost:61518
    Thread [main] (Suspended (breakpoi
      Debug.percentageOfArray(int, int[
      Debug.main(String[]) line: 8
  /Library/Java/JavaVirtualMachines/jdk1.
```

```java
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count = 1;

            }

        }

        double avg = count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

}
```

Name	Value
no method return value	
num	0
▶ arr	(id=15)

The next line in the code is highlighted and the count variable has now been instantiated:



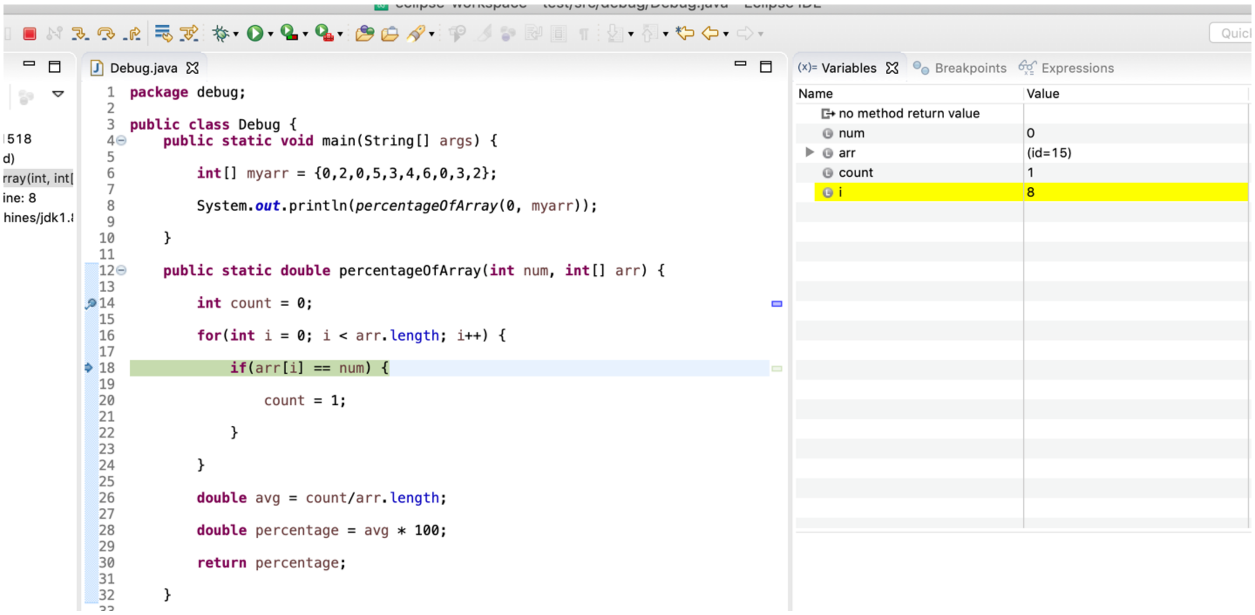
eclipse-workspace - test/src/debug/Debug.java - Eclipse IDE

```
Debug [Java Application]
  debug.Debug at localhost:61518
    Thread [main] (Suspended)
      Debug.percentageOfArray(int, int[
      Debug.main(String[]) line: 8
  /Library/Java/JavaVirtualMachines/jdk1.
```

```java
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count = 1;

            }

        }

        double avg = count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

}
```

Name	Value
no method return value	
num	0
▶ arr	(id=15)
count	0

Step 4: Keep clicking the "Step Into" button to see how the variables change and cross check it with your logic. Notice anything odd during the for-loop?

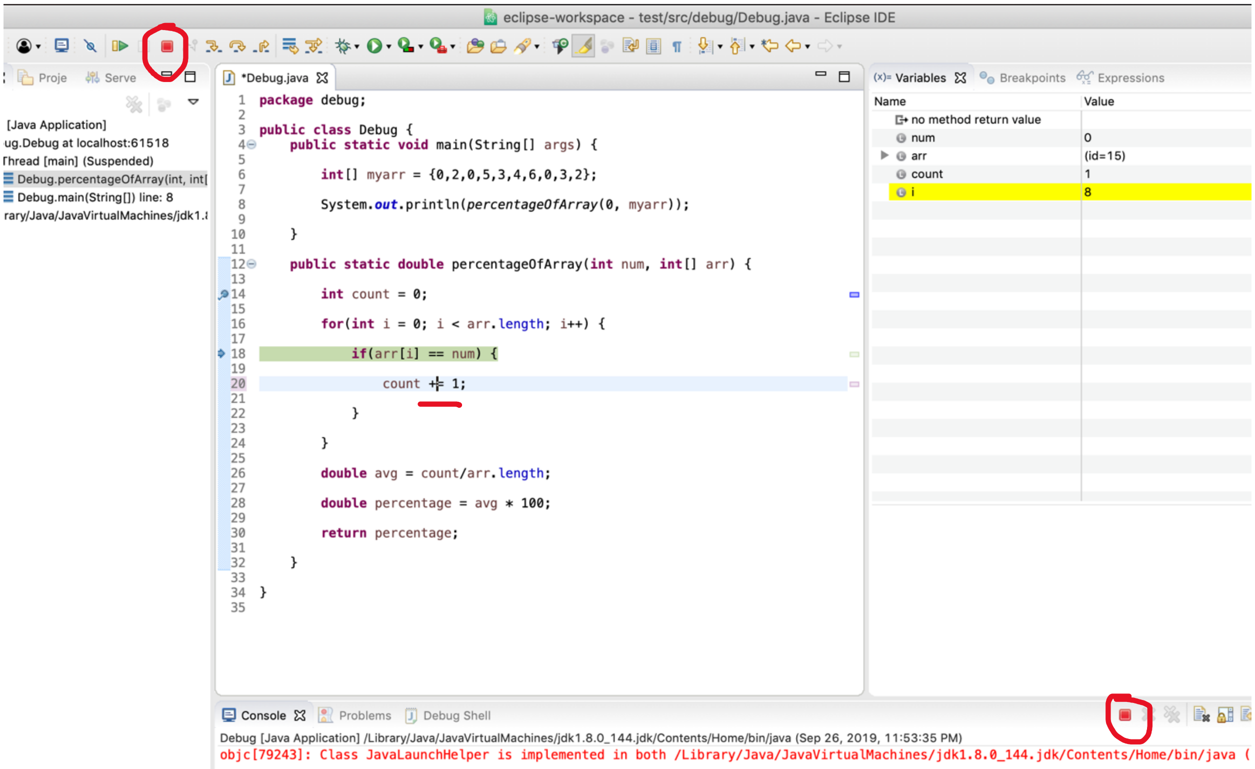


```
518
d)
rray(int, int[
ine: 8
hines/jdk1.8
```

```java
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count = 1;

            }

        }

        double avg = count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }
```

Name	Value
no method return value	
num	0
arr	(id=15)
count	1
i	8

The count variable always seems to be equal to 1 even as I increases and we find more zeroes in the array! Let's change this line to += rather than just = so we are incrementing rather than just reassigning. Let's also stop the debugger by clicking either of the two red squares:



```
[Java Application]
ug.Debug at localhost:61518
Thread [main] (Suspended)
Debug.percentageOfArray(int, int[
Debug.main(String[]) line: 8
rary/Java/JavaVirtualMachines/jdk1.8
```

```java
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count += 1;

            }

        }

        double avg = count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

}
```

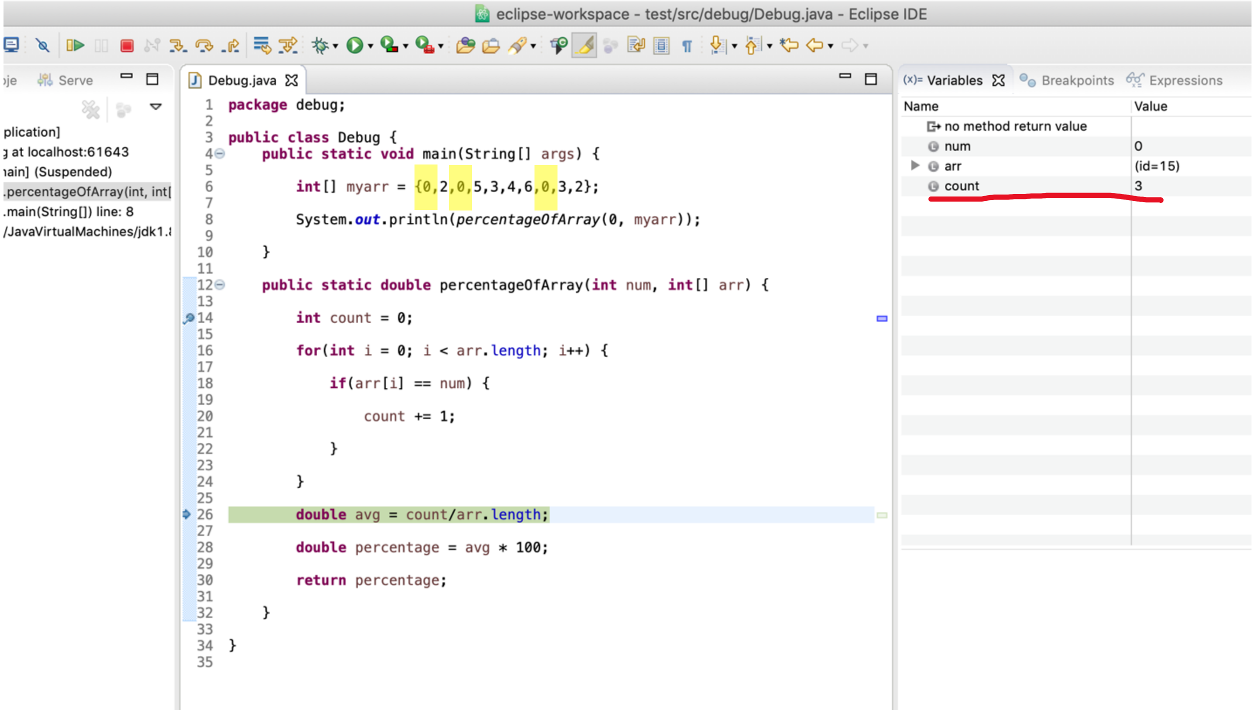
Name	Value
no method return value	
num	0
arr	(id=15)
count	1
i	8

```
Debug [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (Sep 26, 2019, 11:53:35 PM)
objc[79243]: Class JavaLaunchHelper is implemented in both /Library/Java/JavaVirtualMachines/jdk1.8.0_144.jdk/Contents/Home/bin/java (
```

Once you run it again, you'll find that 0.0 is still being printed!! What's the issue now? Run through the debugging process again and see if anything has changed from our count+=1 fix.

(side note: count+=1 is the same as count = count + 1)

After the for-loop ends, our count looks right! There are 3 zeroes in the array:



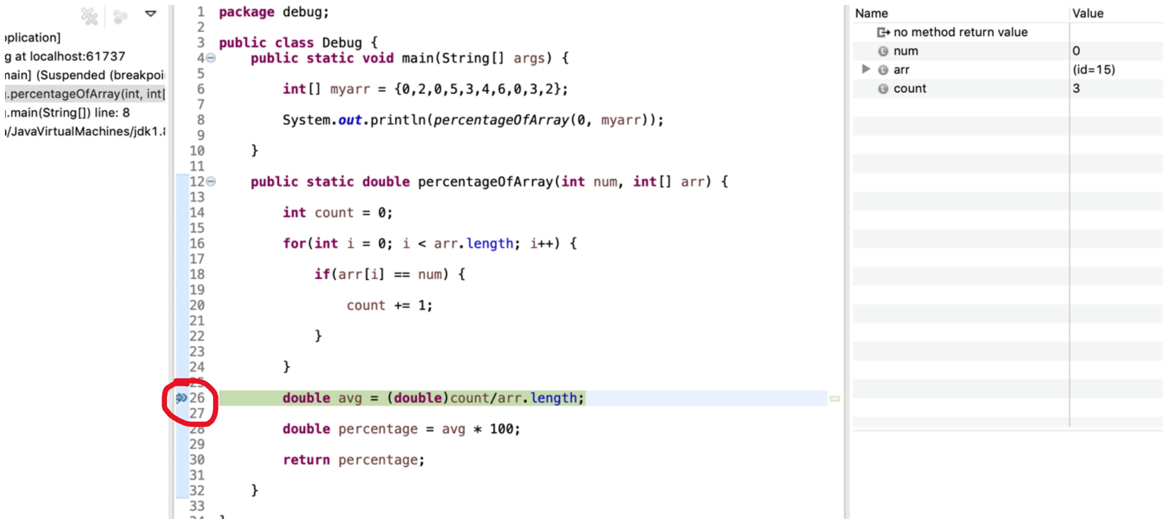
eclipse-workspace - test/src/debug/Debug.java - Eclipse IDE

plication]
g at localhost:61643
ain] (Suspended)
.percentageOfArray(int, int[
.main(String[]) line: 8
/JavaVirtualMachines/jdk1.

```
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count += 1;

            }

        }

        double avg = count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

}
```

Name	Value
no method return value	
num	0
arr	(id=15)
count	3

If we keep going, we find that avg is equal to 0.0, but that isn't right. If count is 3 and the length of arr is 10, what is 3/10 in Java? 0! And making avg a double just turns that into 0.0. How do we fix this? Cast count to a double so Java knows not to throw away the remainder. Also since we know that the for-loop for count is correct, we can remove the breakpoint and add a new one further down in the code at avg:

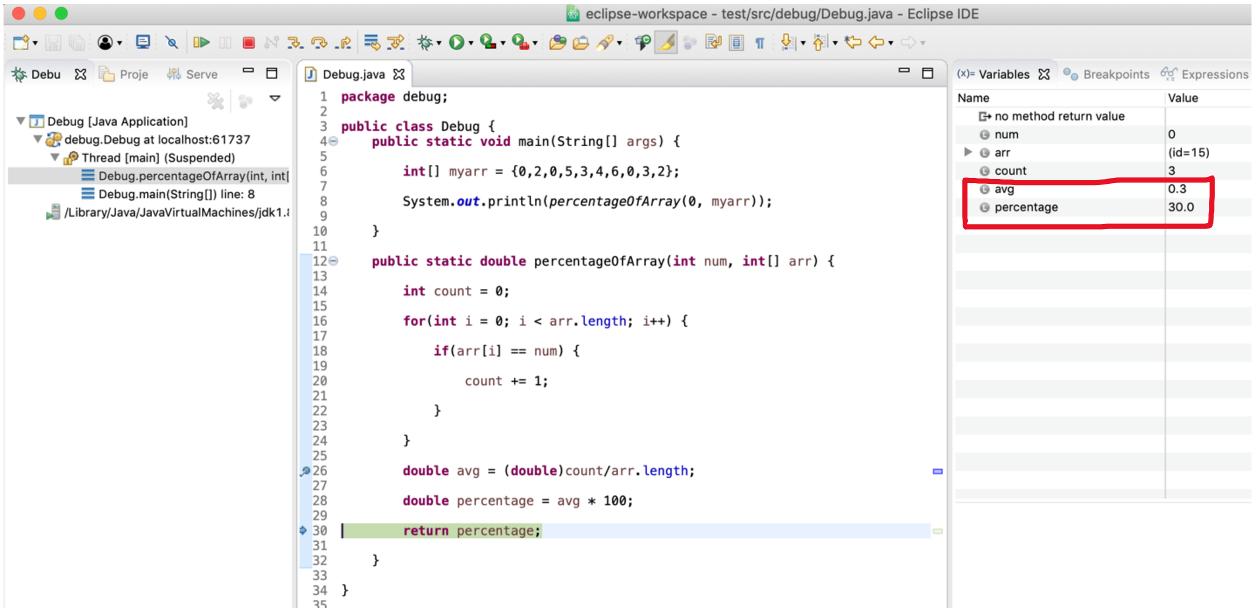


plication]
g at localhost:61737
ain] (Suspended (breakpoi
.percentageOfArray(int, int[
.main(String[]) line: 8
/JavaVirtualMachines/jdk1.

```
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count += 1;

            }

        }

        double avg = (double)count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

```

Name	Value
no method return value	
num	0
arr	(id=15)
count	3

After you step through the rest of the code you'll find that avg and percentage now equal the right numbers!:



```
Debug [Java Application]
  debug.Debug at localhost:61737
    Thread [main] (Suspended)
      Debug.percentageOfArray(int, int[
      Debug.main(String[]) line: 8
  /Library/Java/JavaVirtualMachines/jdk1.
```

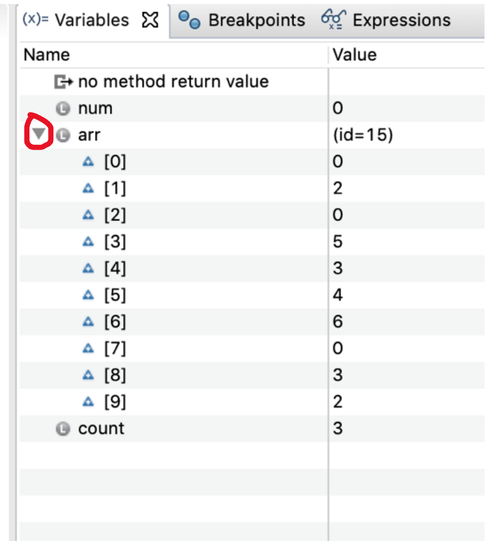
```java
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count += 1;

            }

        }

        double avg = (double)count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

}
```

Name	Value
no method return value	
num	0
arr	(id=15)
count	3
avg	0.3
percentage	30.0

Yay the method is now working properly!

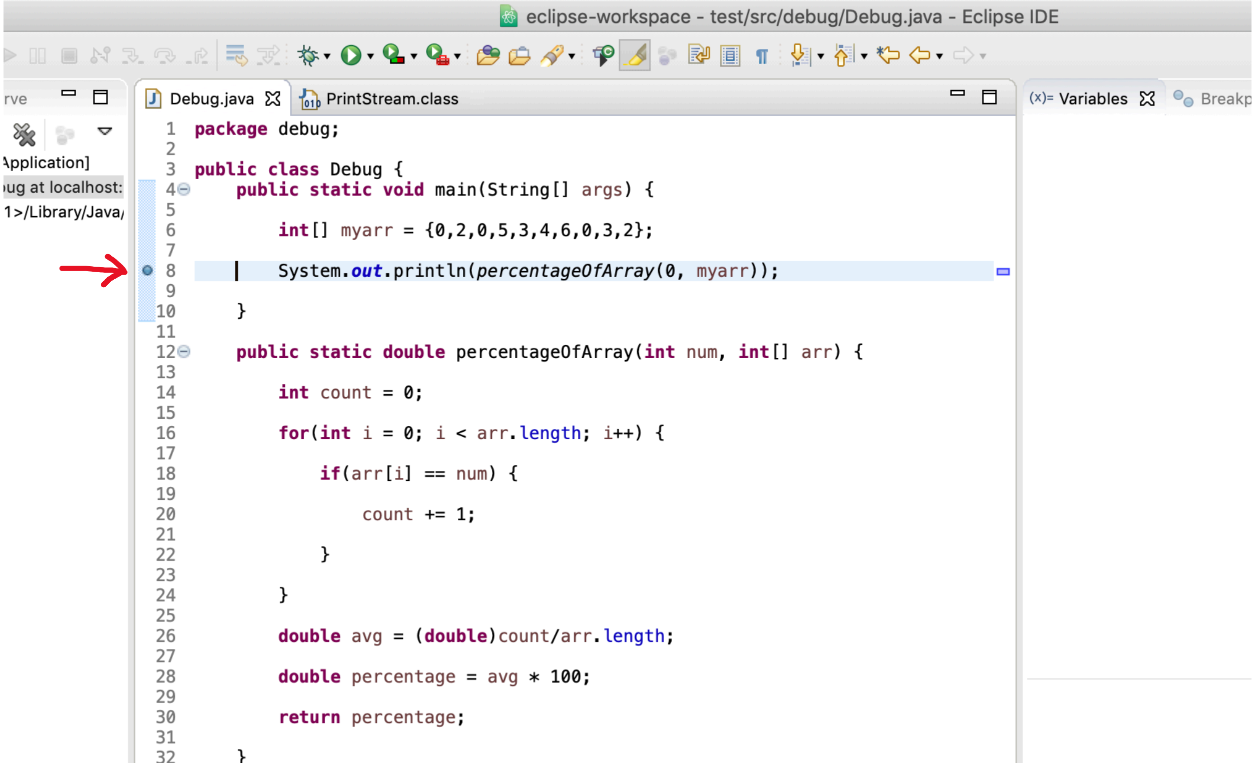
Some more features:

You can expand the array variable arr to see what's inside:



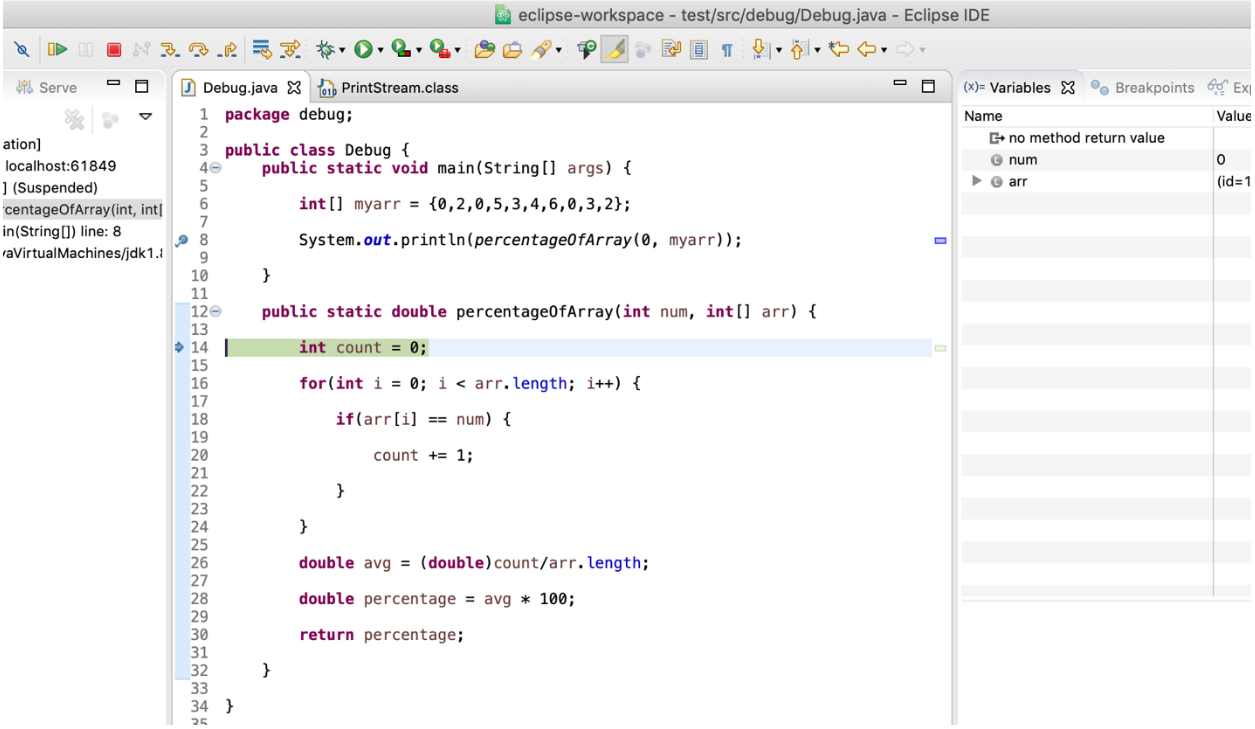
Name	Value
no method return value	
num	0
arr	(id=15)
[0]	0
[1]	2
[2]	0
[3]	5
[4]	3
[5]	4
[6]	6
[7]	0
[8]	3
[9]	2
count	3

The “Step Over” button directly to the right of the “Step Into” button will pass over methods called rather than stepping in. For example, if you put the breakpoint at the print statement in the main method:



```java
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count += 1;

            }

        }

        double avg = (double)count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }
```

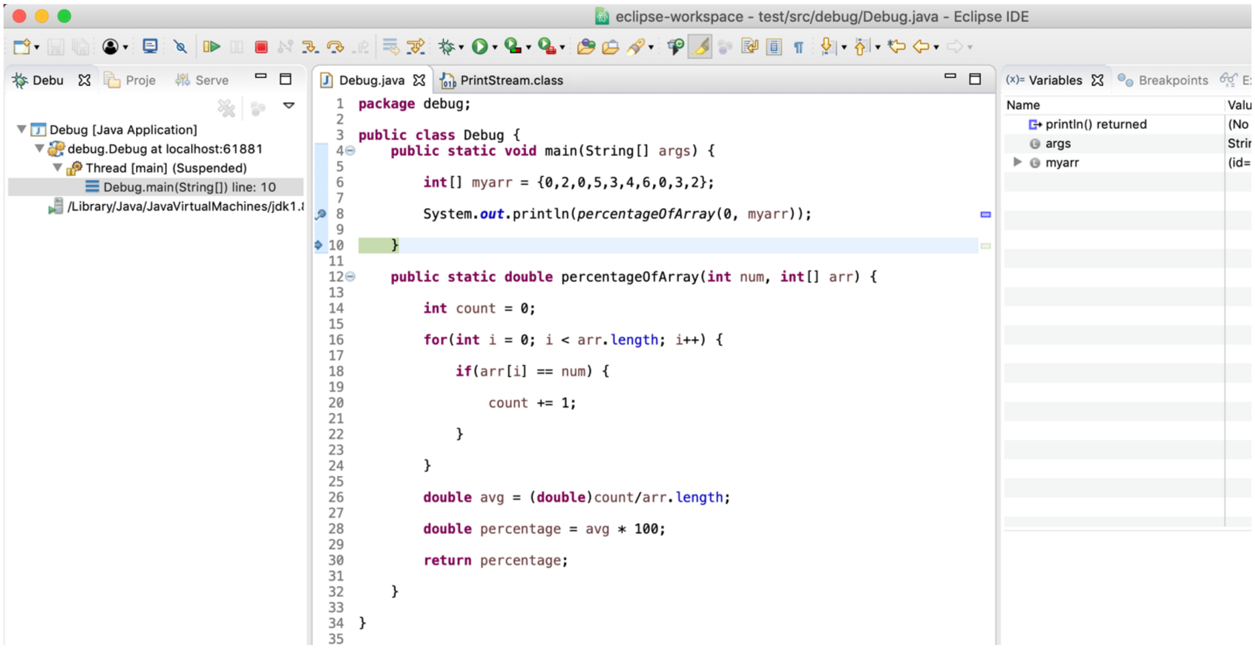
Pressing “Step Into” when debugging will move the program into the first line of percentageOfArray():



```java
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count += 1;

            }

        }

        double avg = (double)count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

}
```

Name	Value
no method return value	
num	0
arr	(id=1

Pressing “Step Over” when debugging will move the program over the whole percentageOfArray() method:



```
Debug [Java Application]
  debug.Debug at localhost:61881
    Thread [main] (Suspended)
      Debug.main(String[]) line: 10
  /Library/Java/JavaVirtualMachines/jdk1.8
```

```java
package debug;

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

        int[] myarr = {0,2,0,5,3,4,6,0,3,2};

        System.out.println(percentageOfArray(0, myarr));

    }

    public static double percentageOfArray(int num, int[] arr) {

        int count = 0;

        for(int i = 0; i < arr.length; i++) {

            if(arr[i] == num) {

                count += 1;

            }

        }

        double avg = (double)count/arr.length;

        double percentage = avg * 100;

        return percentage;

    }

}

```

Name	Valu
println() returned	(No
args	Strir
myarr	(id=