Debugging

Help with finding bugs in your code

Produced by: Dr. Siobhán Drohan
Ms. Mairéad Meagher
Siobhan Roche

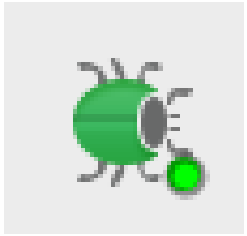
Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

Department of Computing and Mathematics
http://www.setu.ie/

Topic List

1. What are **bugs**?

2. What are **debuggers**?



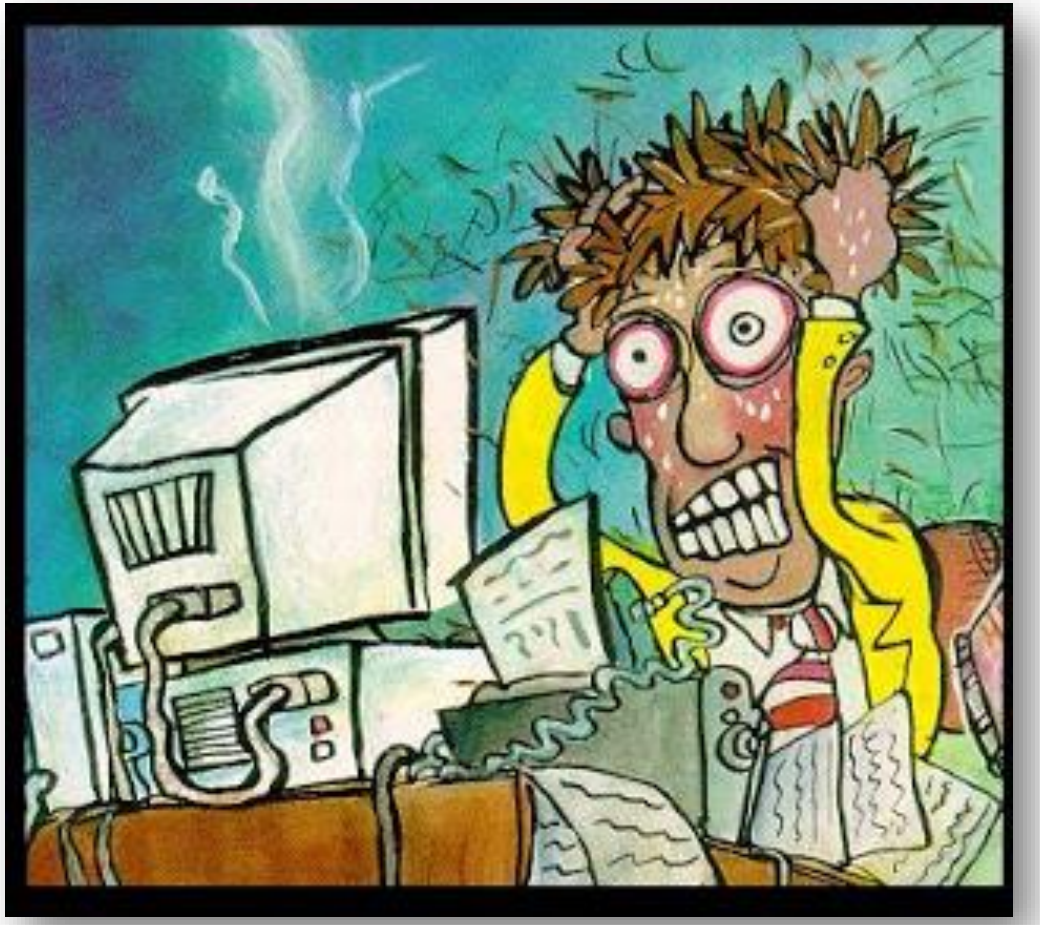
3. How do I use them?

What are **bugs**?

A software **bug** is an error, flaw, failure or fault in a **computer** program or system that causes it to produce an incorrect or unexpected result, or to behave in unintended ways.

Software bug - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/Software_**bug**

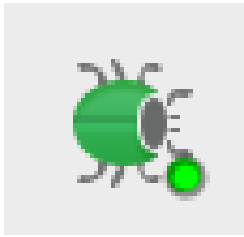
Bugs can be frustrating to find/fix



Topic List

1. What are **bugs**?

2. What are **debuggers**?



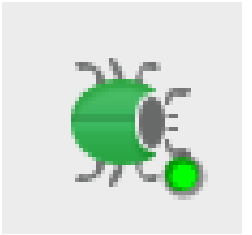
3. How do I use them?

Help is at hand…debuggers!

A debugger can be **used to fix bugs**

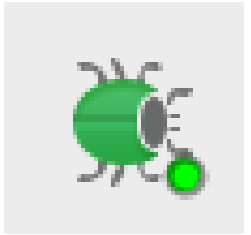
…hence the name debugger!

Debugger

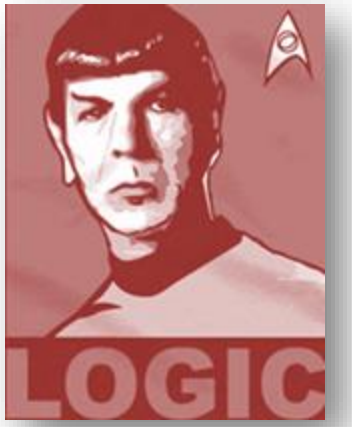


- A **debugger** is a software tool that
  - helps in examining how an application executes
  - allows programmers execute an application one statement at a time.
    - **Step, step into, step out**.
  - typically provides functions
    - to stop and start a program at selected points in the source code(**breakpoints**)
    - to examine the values of variables (**watch, trace**)

Debugger



- Debuggers are especially useful when your program contains logical errors.
  - i.e. errors that the compiler will not pickup but that lead to incorrect results
  - e.g. if your syntax is correct but the logic of your problem solution is faulty.

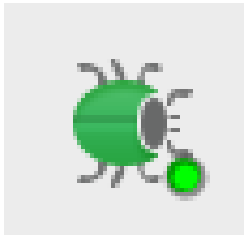


- Using the debugger, you can **trace** how each of the calculations and changes made to fields/variables happen and hopefully **figure out where the error is occurring**.

Topic List

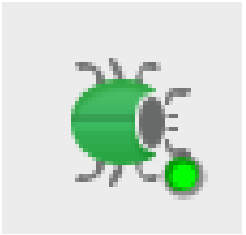
1. What are **bugs**?

2. What are **debuggers**?



3. How do I use them?

Debugger



- Most IDEs come with a debugger; IntelliJ has one.

- We are going to use the IntelliJ Debugger to **step** through the debugging of a small program

  – The program iterates over a primitive array of ***int*** and prints out the largest number in the array.

```java
public class Driver {

    public static void main(String args[])
    {
        int list[] = {2,5,3,4};
        int largestNumber = Largest.findLargest(list);
        System.out.println("Largest number is: " + largestNumber);
    }
}
```

Given this code…

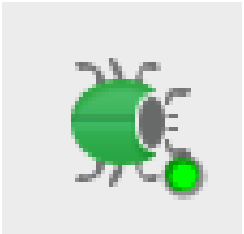
We are expecting this output:

Largest number is: 5

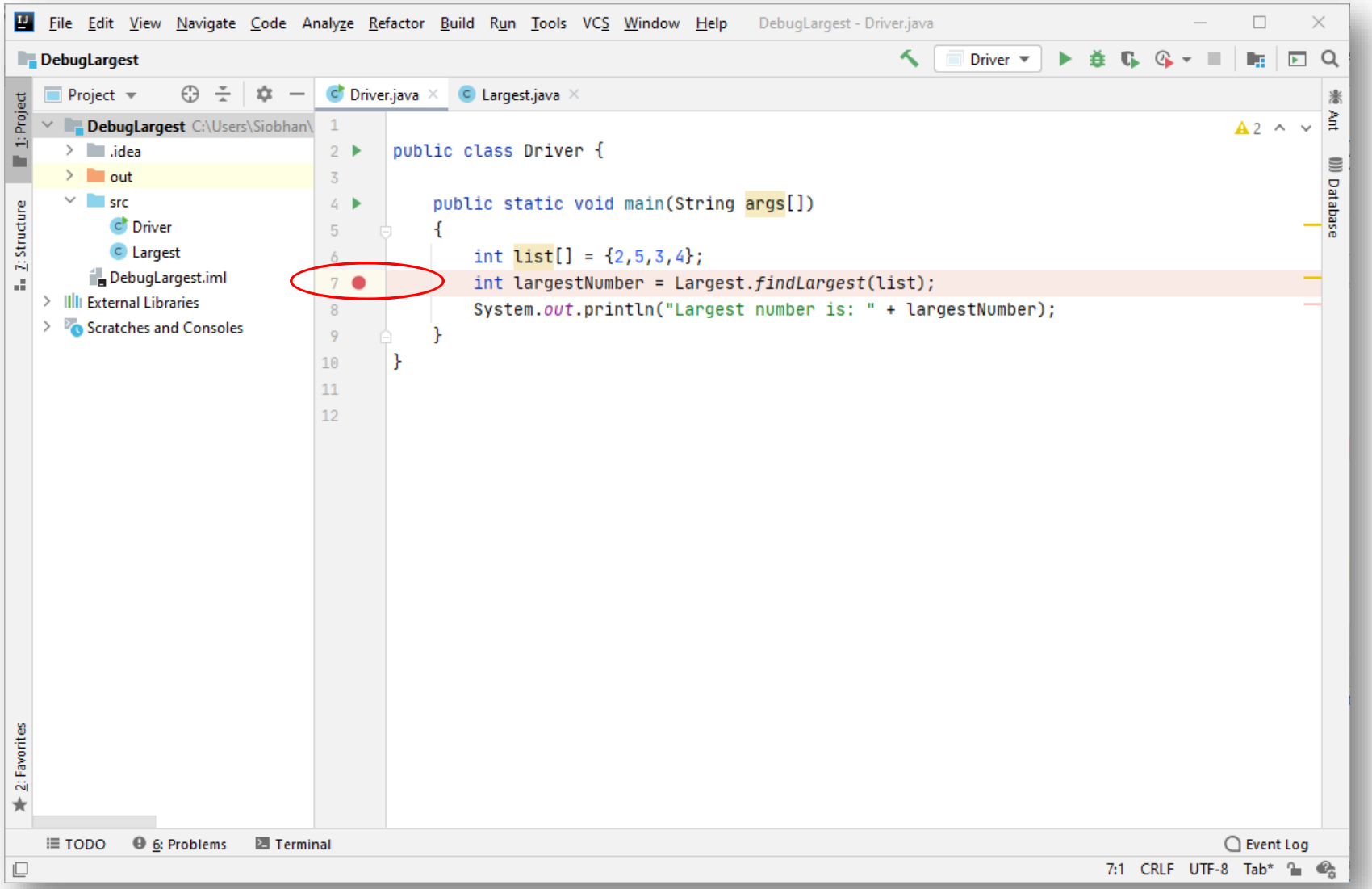
But we get:

Largest number is: 2147483647

```java
public class Largest {

    public static int findLargest (int[] list) {
        int index = 0;
        int max = Integer.MAX_VALUE;

        for (index = 0; index < list.length; index++) {
            if (list[index] > max) {
                max = list[index];
            }
        }

        return max;
    }
}
```

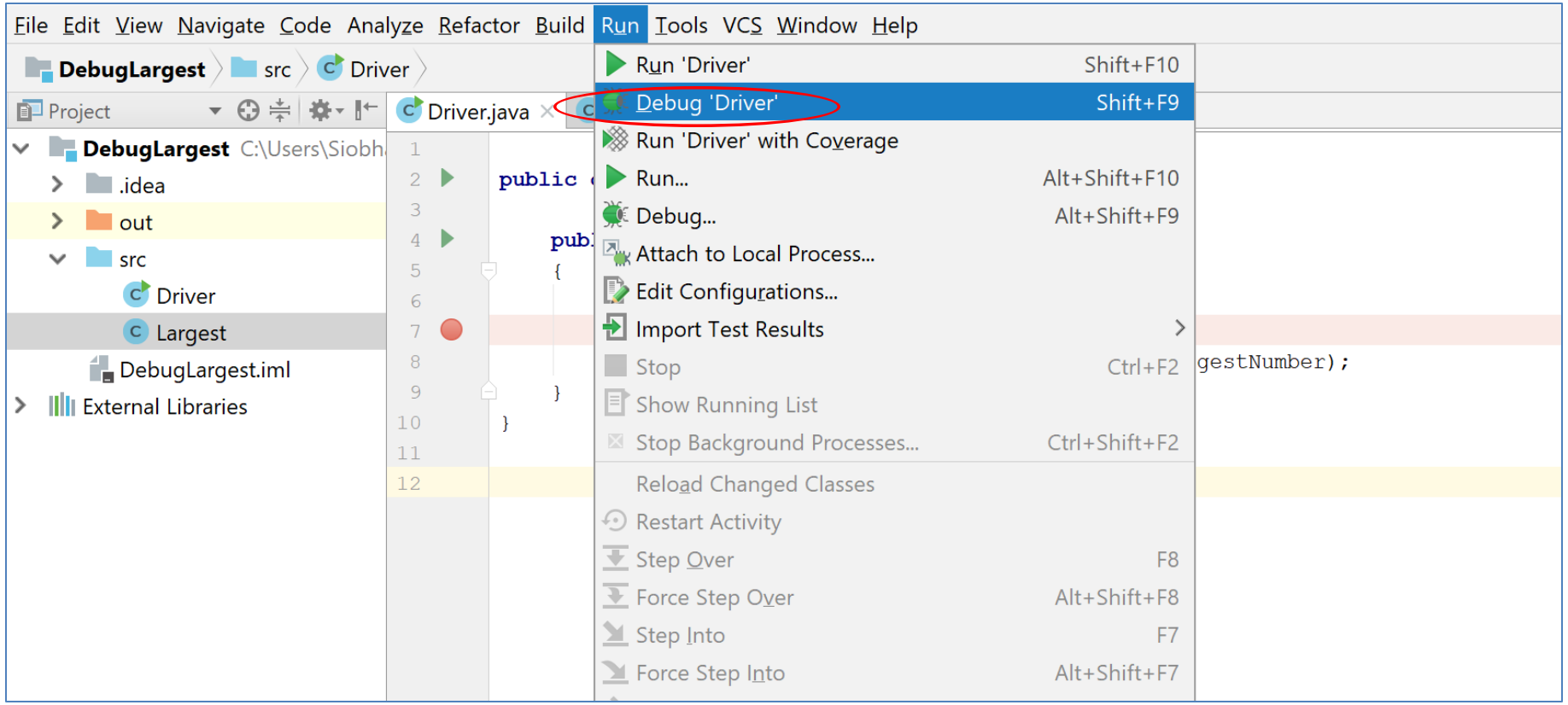


Let’s debug the code
in IntelliJ
to help us find the error…



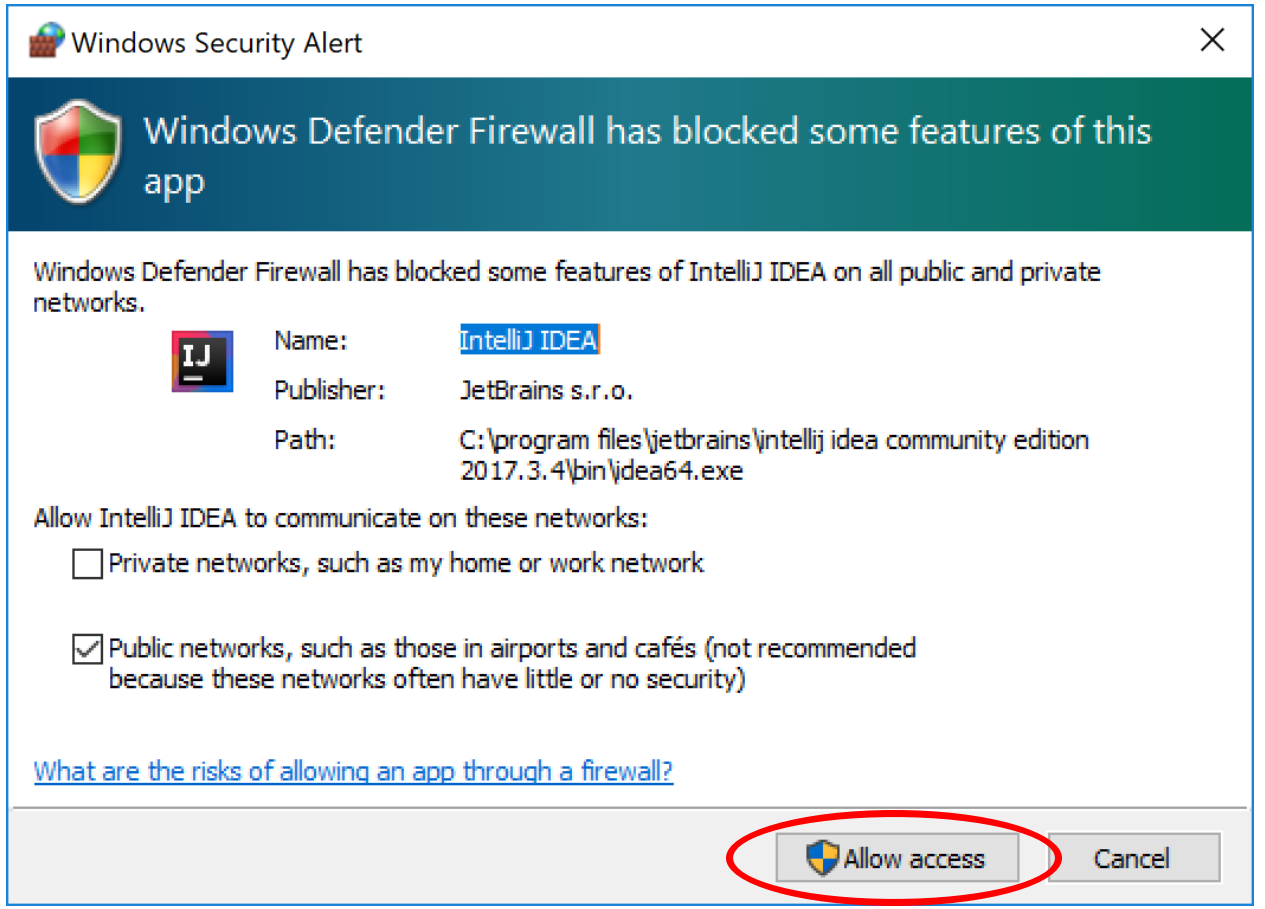
1

Click in the grey margin beside line 7. This will set up a **breakpoint** on this line.



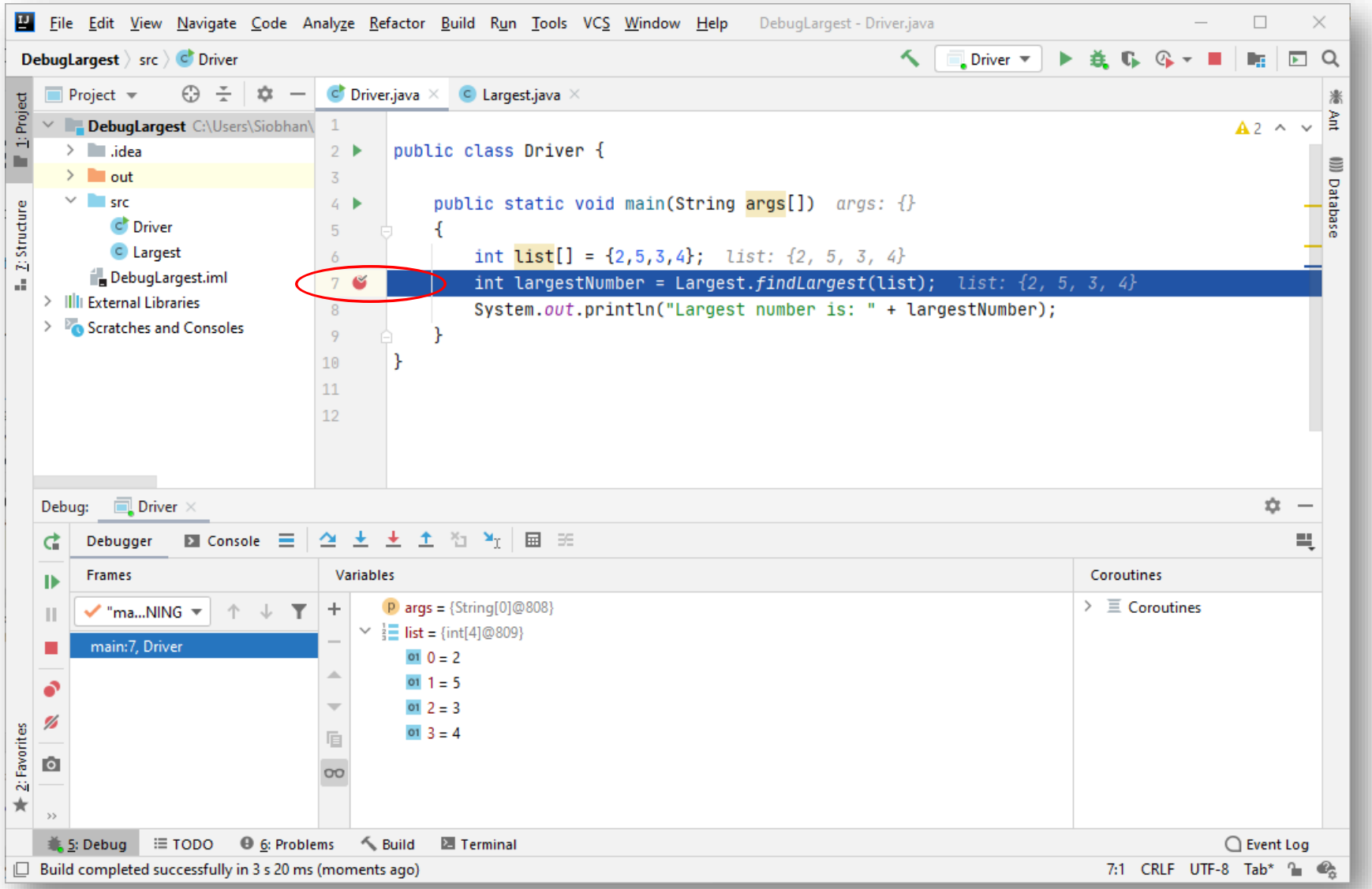
2

From the Run menu, select **Debug** 'Driver'.



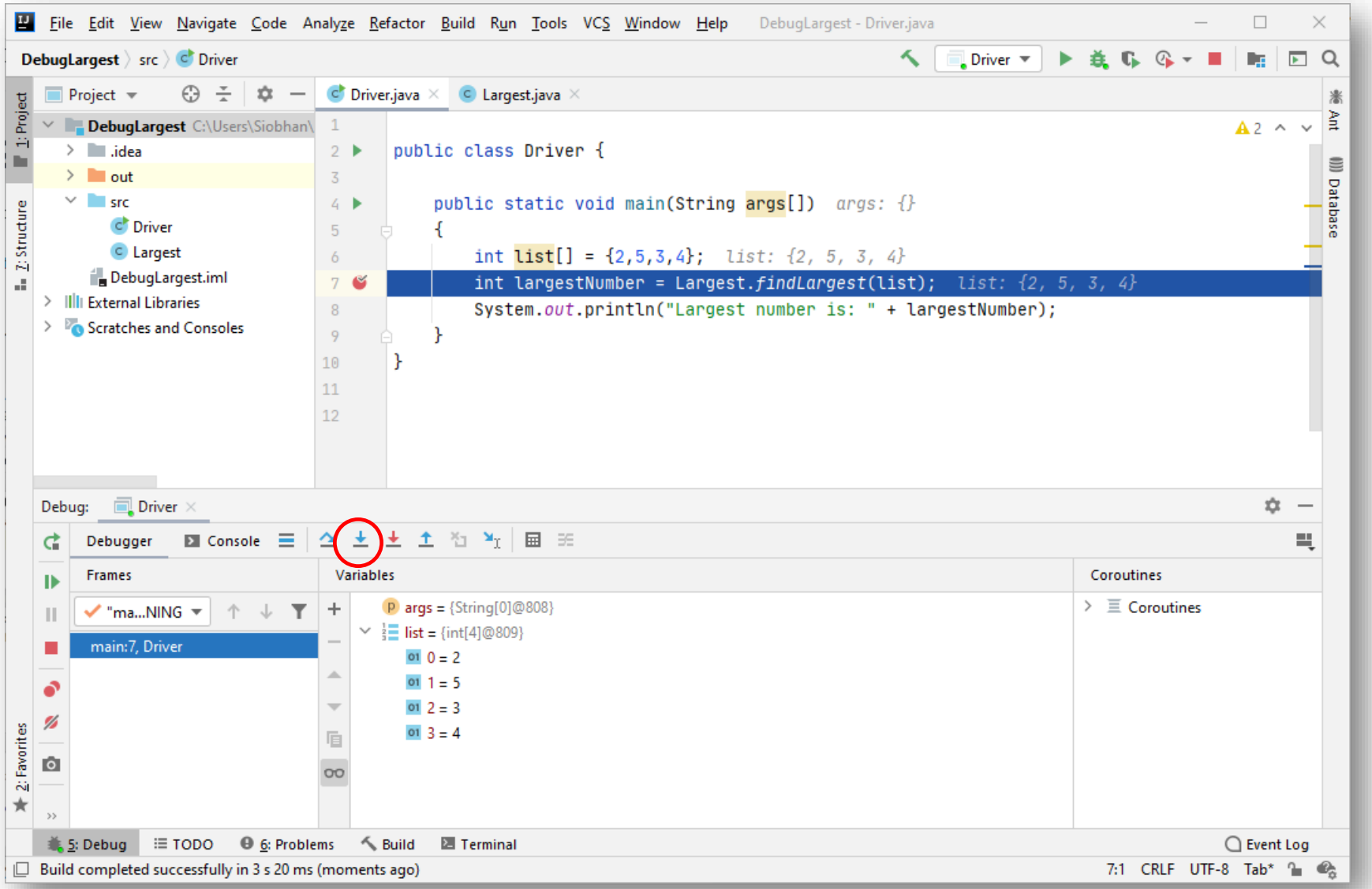
3

If this window appears, click on “Allow access”.



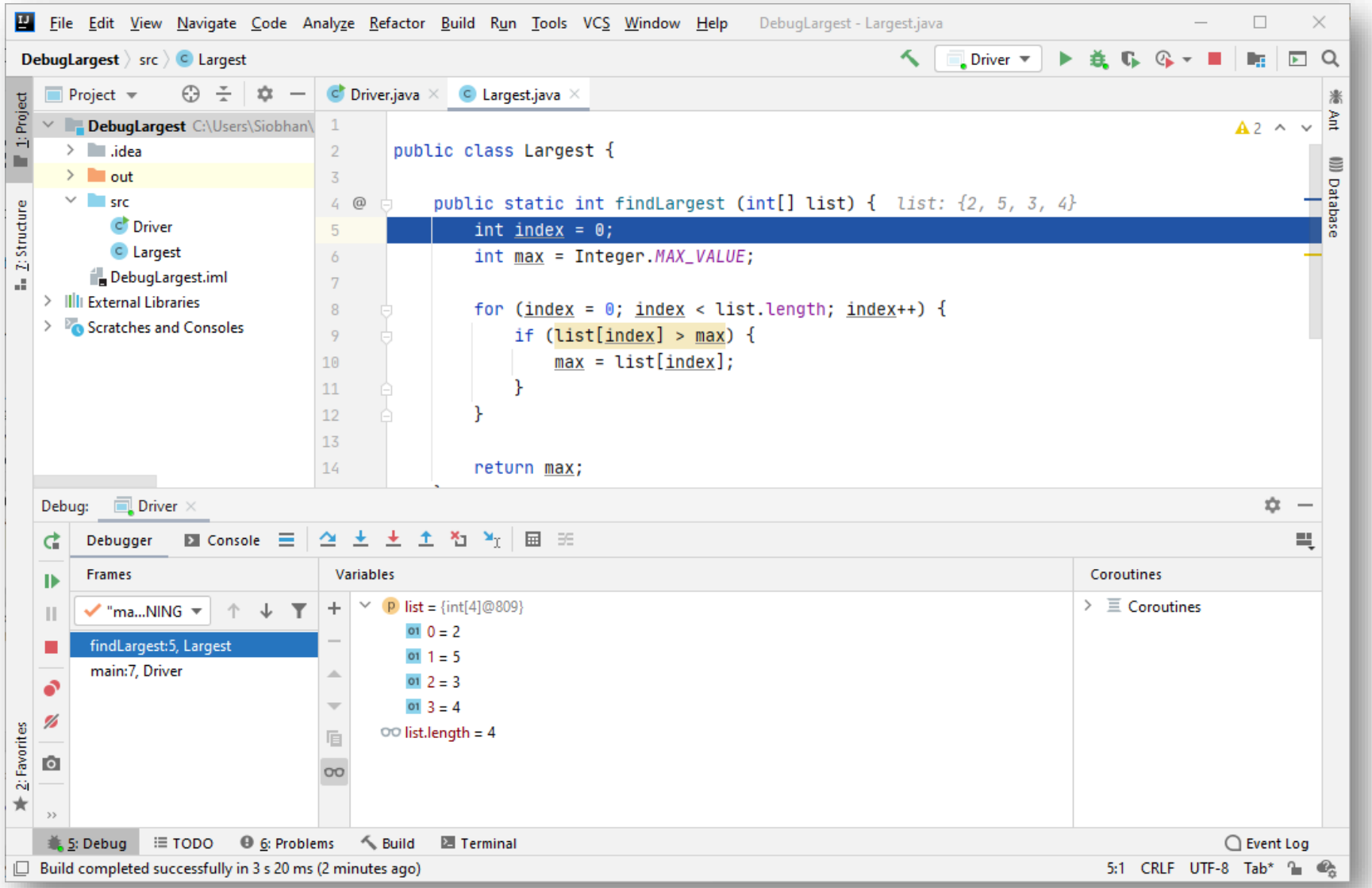
4

You are now in Debug mode...the program has stopped just before executing line 7.



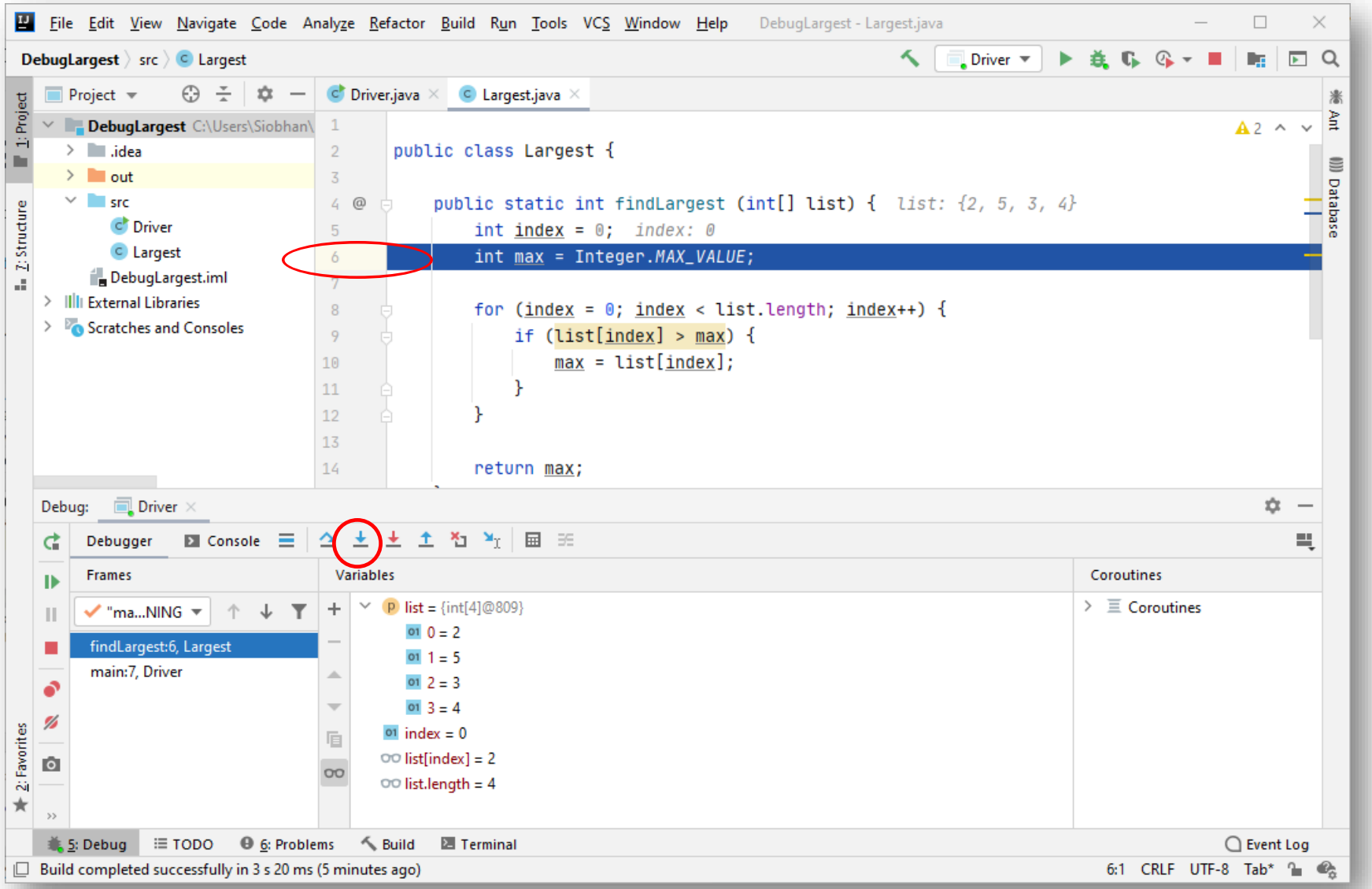
5

‘**Step Into**’ the findLargest method…



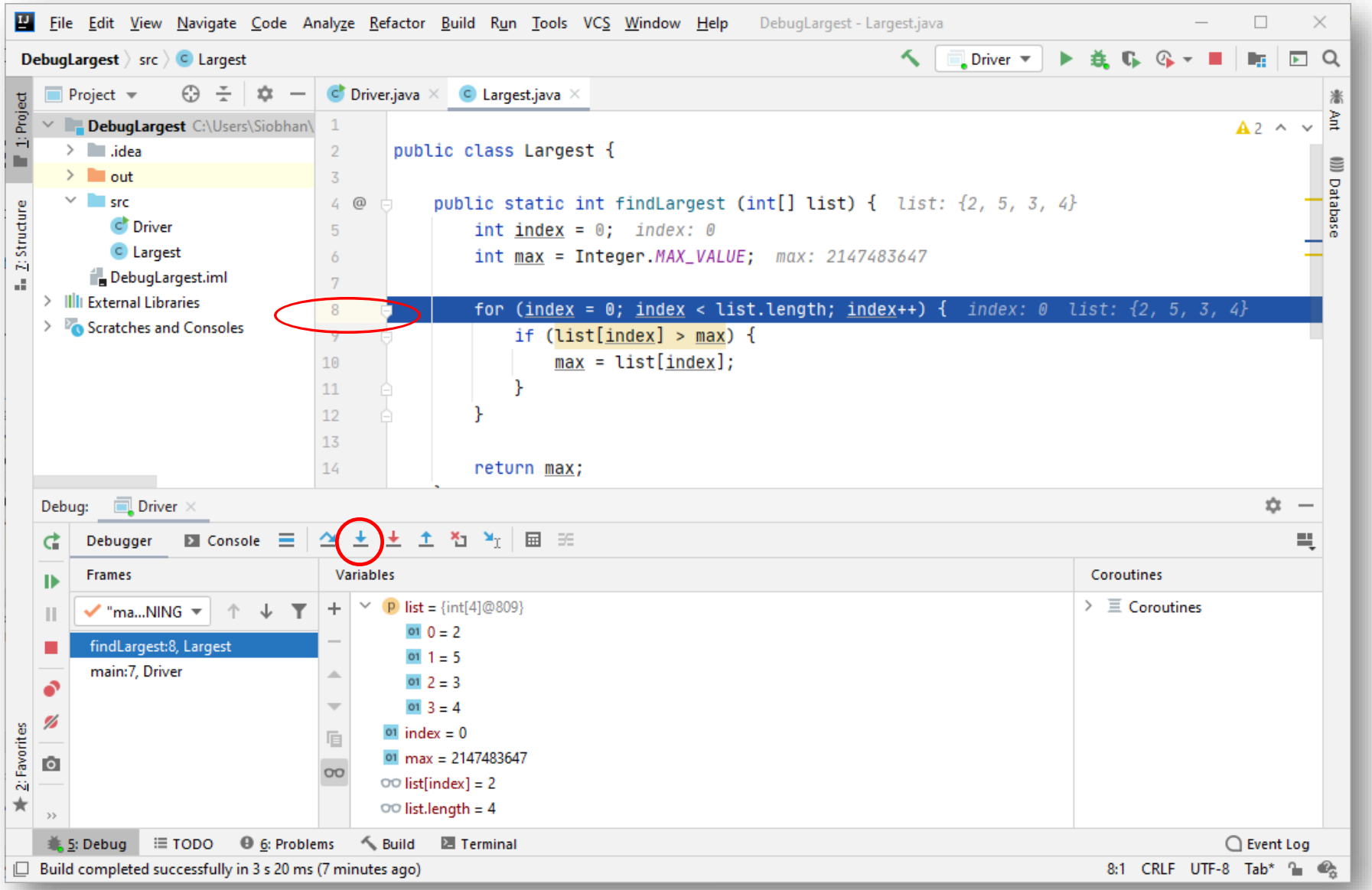
6

Now that we are in the findLargest method, we want to execute each line while monitoring the value of Max. This should help us locate the error…



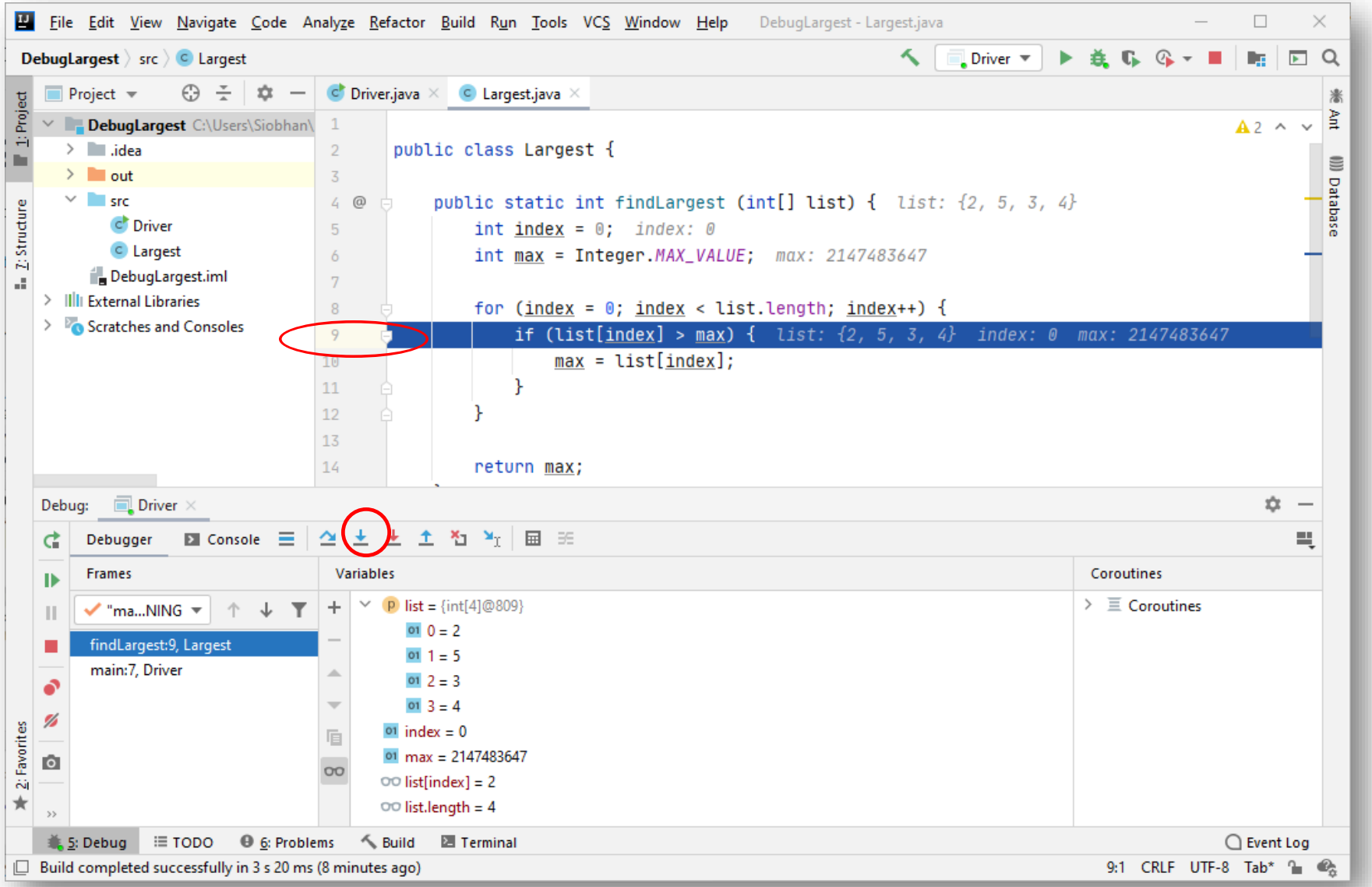
7

'Step Into' again → execution now stopped on line 6



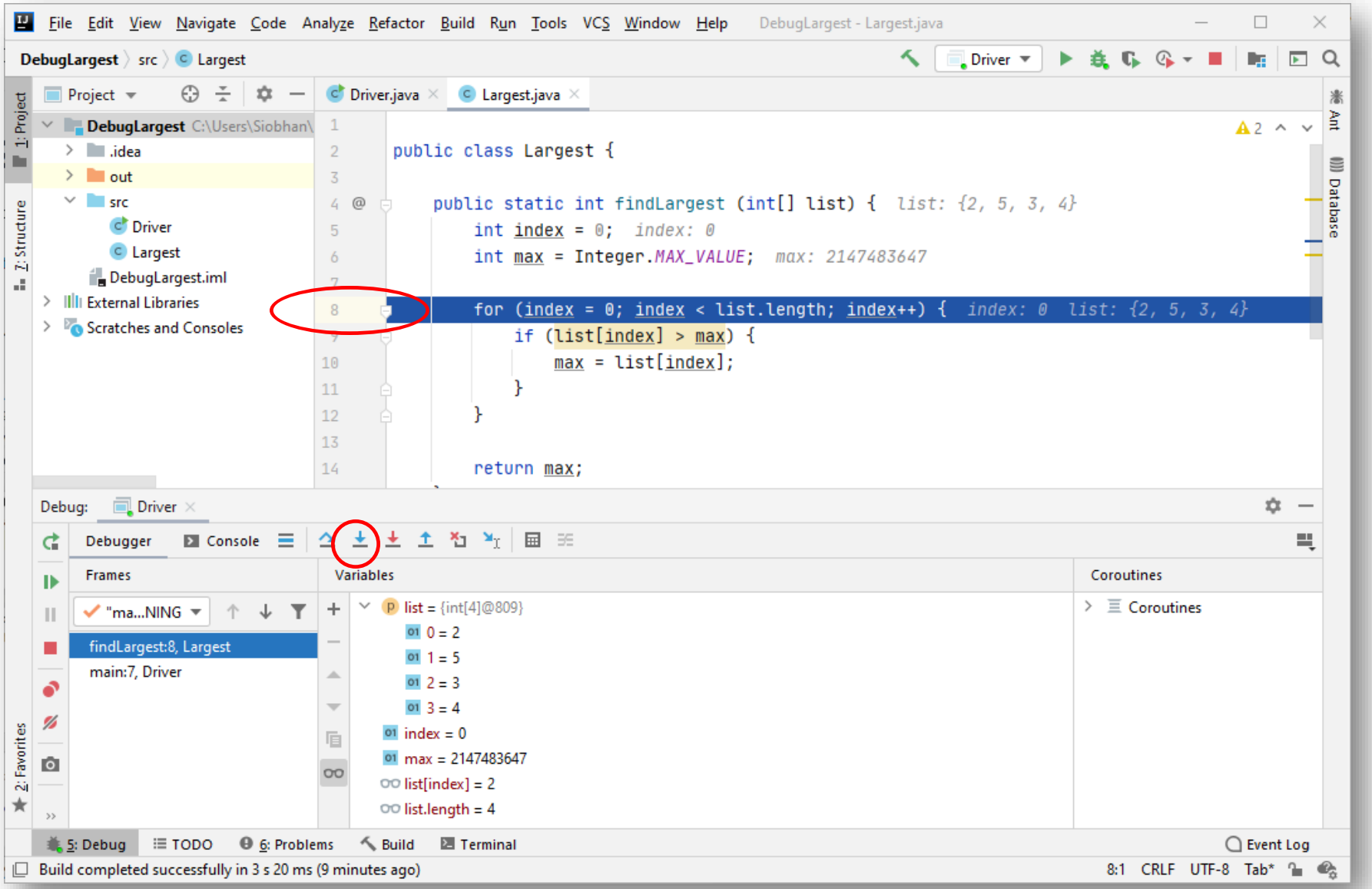
8

Step Into → execution now stopped on line 8…note the value of **max**.



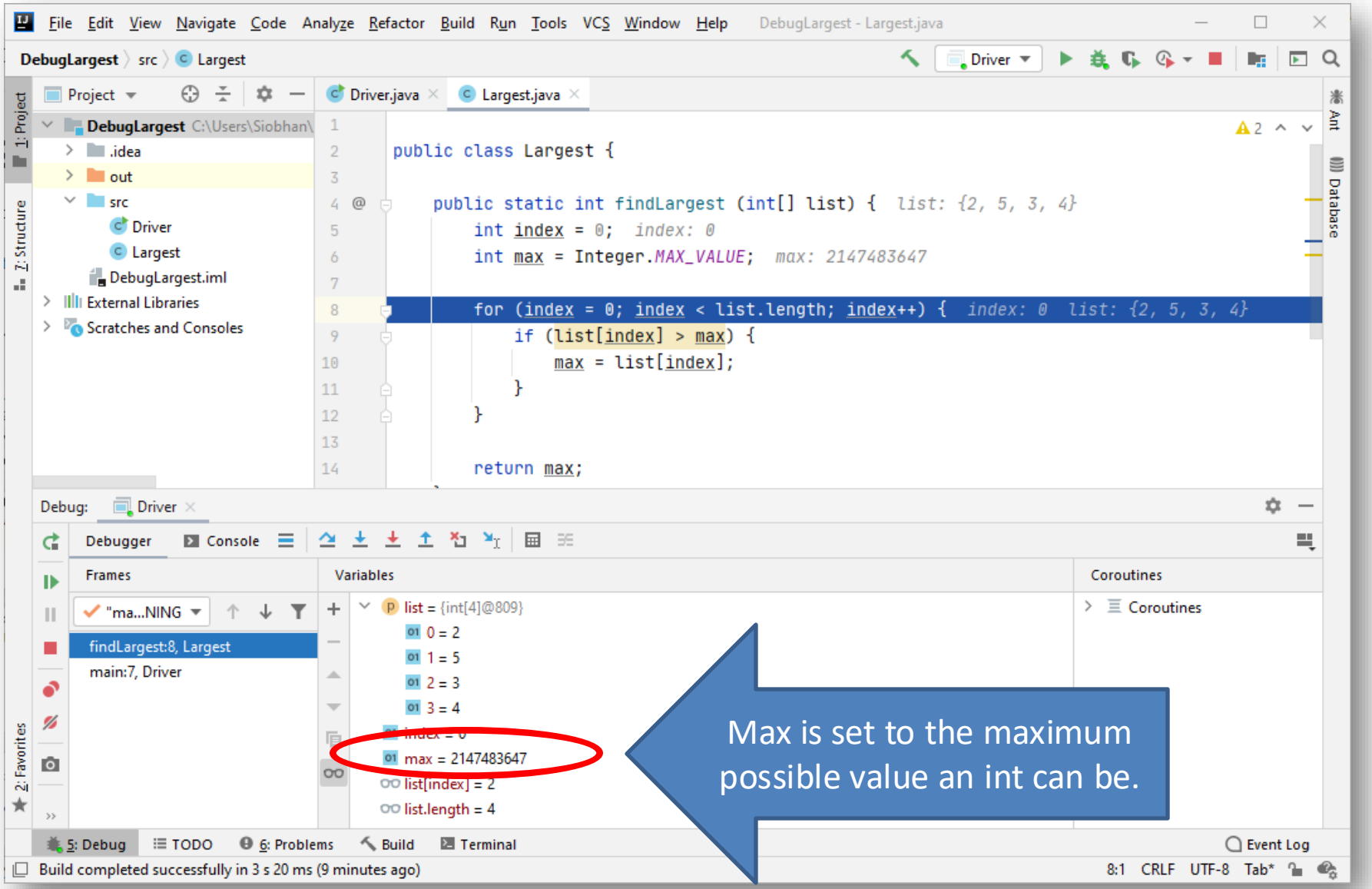
9

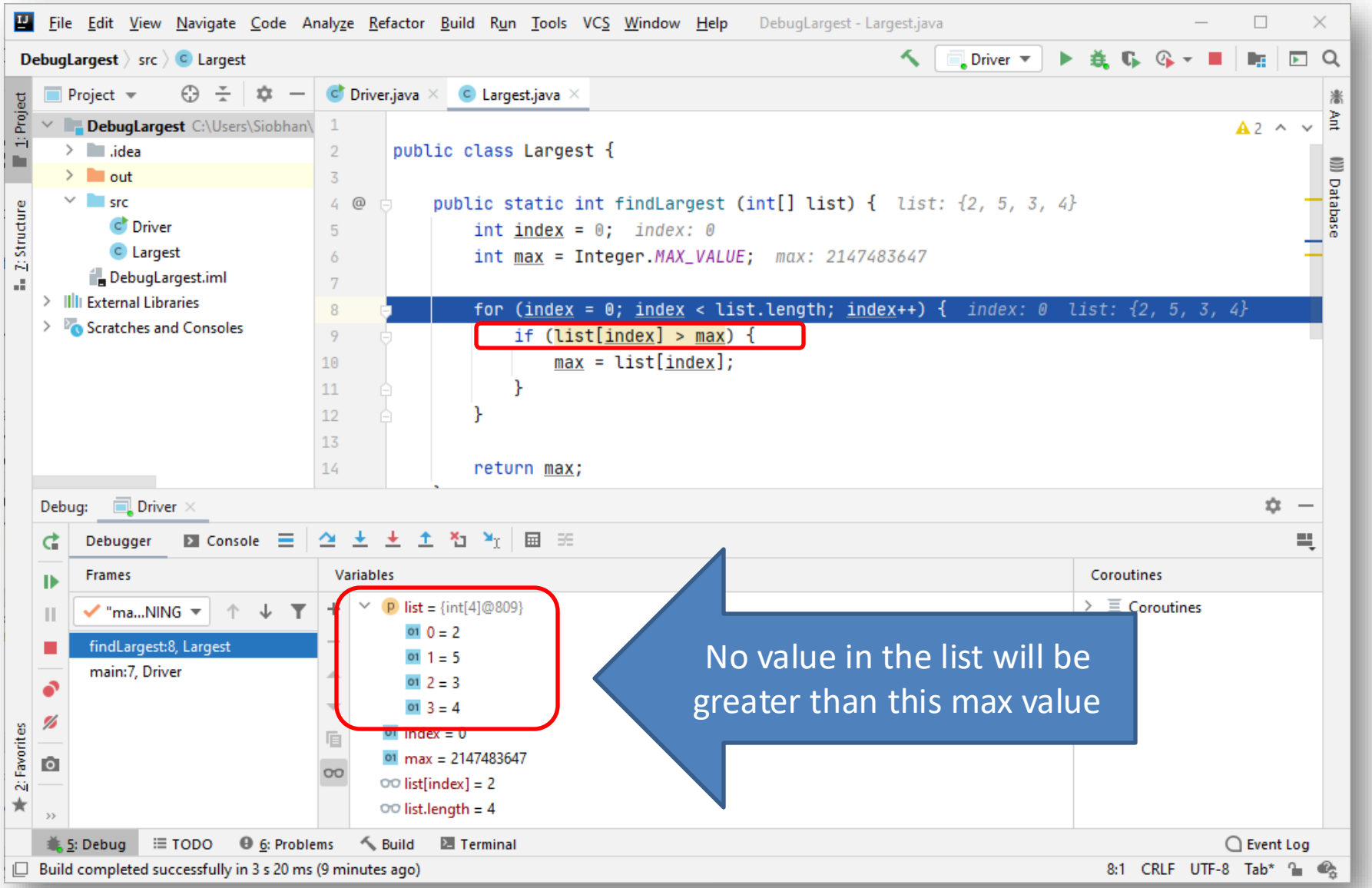
Step Into → execution now stopped on line 9...

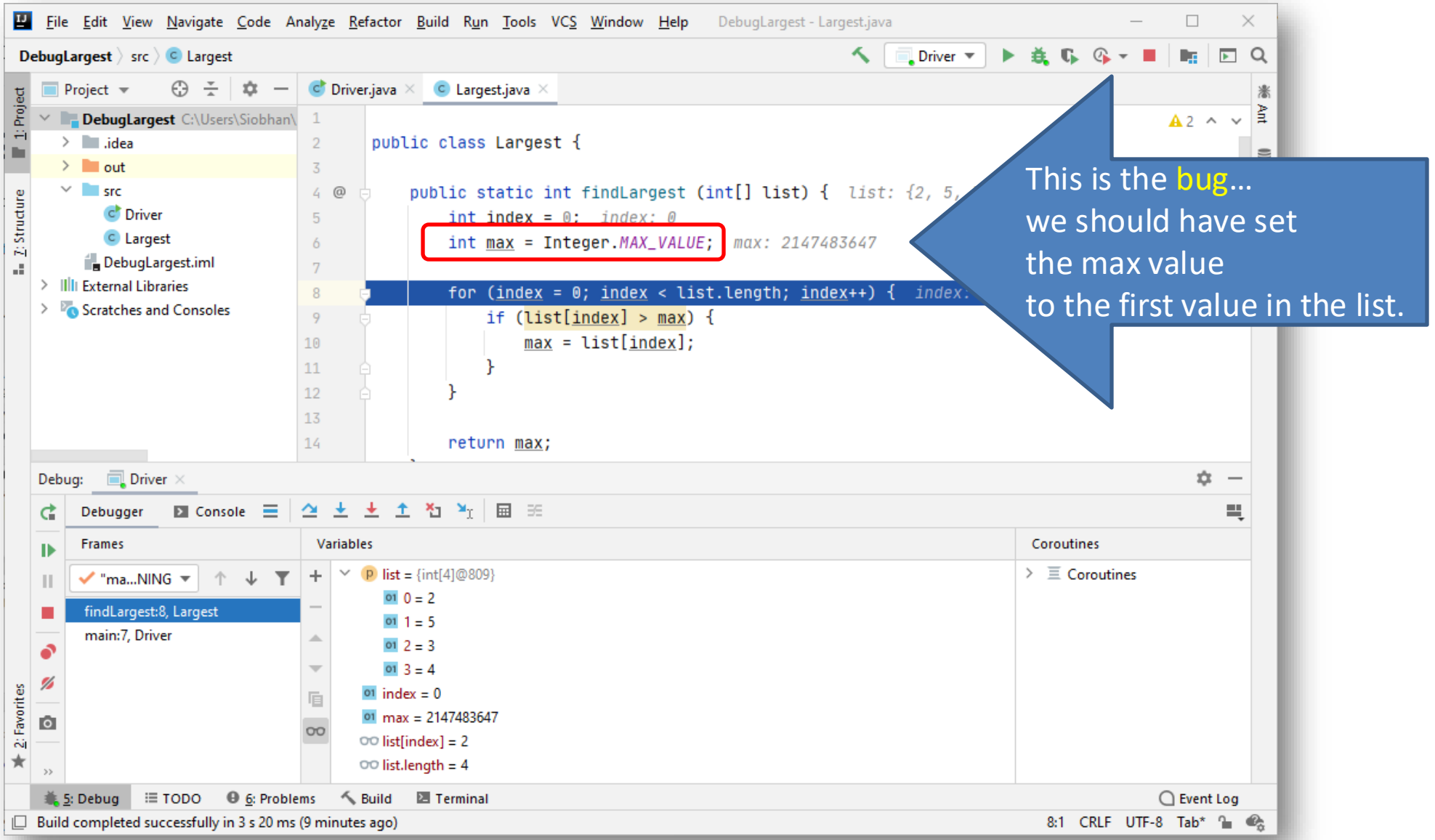


10

Step Into → execution now stopped back on line 8…
can you see the problem?


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
DebugLargest - Largest.java
DebugLargest src Largest
Driver
Project
Driver.java
Largest.java
DebugLargest C:\Users\Siobhan\
.idea
out
src
Driver
Largest
DebugLargest.iml
External Libraries
Scratches and Consoles
public class Largest {
public static int findLargest (int[] list) { list: {2, 5, 3, 4}
int index = 0; index: 0
int max = Integer.MAX_VALUE; max: 2147483647
for (index = 0; index < list.length; index++) { index: 0 list: {2, 5, 3, 4}
if (list[index] > max) {
max = list[index];
}
}
return max;
Debug: Driver
Debugger
Console
Frames
Variables
Coroutines
"ma...NING
findLargest:8, Largest
main:7, Driver
list = {int[4]@809}
0 = 2
1 = 5
2 = 3
3 = 4
max = 2147483647
list[index] = 2
list.length = 4
Coroutines
Max is set to the maximum possible value an int can be.
5: Debug
TODO
6: Problems
Build
Terminal
Event Log
Build completed successfully in 3 s 20 ms (9 minutes ago)
8:1 CRLF UTF-8 Tab*
1: Project
7: Structure
2: Favorites
Ant
Database


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
DebugLargest - Largest.java
DebugLargest > src > Largest
Driver
Project
Driver.java
Largest.java
DebugLargest C:\Users\Siobhan\
.idea
out
src
Driver
Largest
DebugLargest.iml
External Libraries
Scratches and Consoles
1 public class Largest {
2
3 public static int findLargest (int[] list) { list: {2, 5, 3, 4}
4 int index = 0; index: 0
5 int max = Integer.MAX_VALUE; max: 2147483647
6
7 for (index = 0; index < list.length; index++) { index: 0 list: {2, 5, 3, 4}
8 if (list[index] > max) {
9 max = list[index];
10 }
11 }
12
13 return max;
14
Debug: Driver
Debugger
Console
Frames
Variables
Coroutines
"ma...NING
findLargest:8, Largest
main:7, Driver
list = {int[4]@809}
0 = 2
1 = 5
2 = 3
3 = 4
index = 0
max = 2147483647
list[index] = 2
list.length = 4
Coroutines
No value in the list will be greater than this max value
5: Debug
TODO
6: Problems
Build
Terminal
Event Log
Build completed successfully in 3 s 20 ms (9 minutes ago)
8:1 CRLF UTF-8 Tab*
1: Project
7: Structure
2: Favorites
Ant
Database


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
DebugLargest - Largest.java
DebugLargest src Largest
Driver
Project
Driver.java
Largest.java
DebugLargest C:\Users\Siobhan\
.idea
out
src
Driver
Largest
DebugLargest.iml
External Libraries
Scratches and Consoles
1 public class Largest {
2
3 public static int findLargest (int[] list) { list: {2, 5,
4 int index = 0; index: 0
5 int max = Integer.MAX_VALUE; max: 2147483647
6
7 for (index = 0; index < list.length; index++) { index:
8 if (list[index] > max) {
9 max = list[index];
10 }
11 }
12
13 return max;
14
This is the bug...
we should have set
the max value
to the first value in the list.
Debug: Driver
Debugger Console
Frames
Variables
Coroutines
"ma...NING
findLargest:8, Largest
main:7, Driver
list = {int[4]@809}
0 = 2
1 = 5
2 = 3
3 = 4
index = 0
max = 2147483647
list[index] = 2
list.length = 4
Coroutines
5: Debug TODO 6: Problems Build Terminal Event Log
Build completed successfully in 3 s 20 ms (9 minutes ago)
8:1 CRLF UTF-8 Tab*

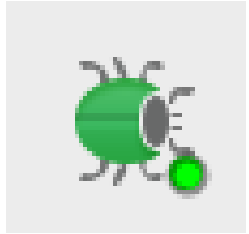
Fixing the bug

- Instead of the line of code:

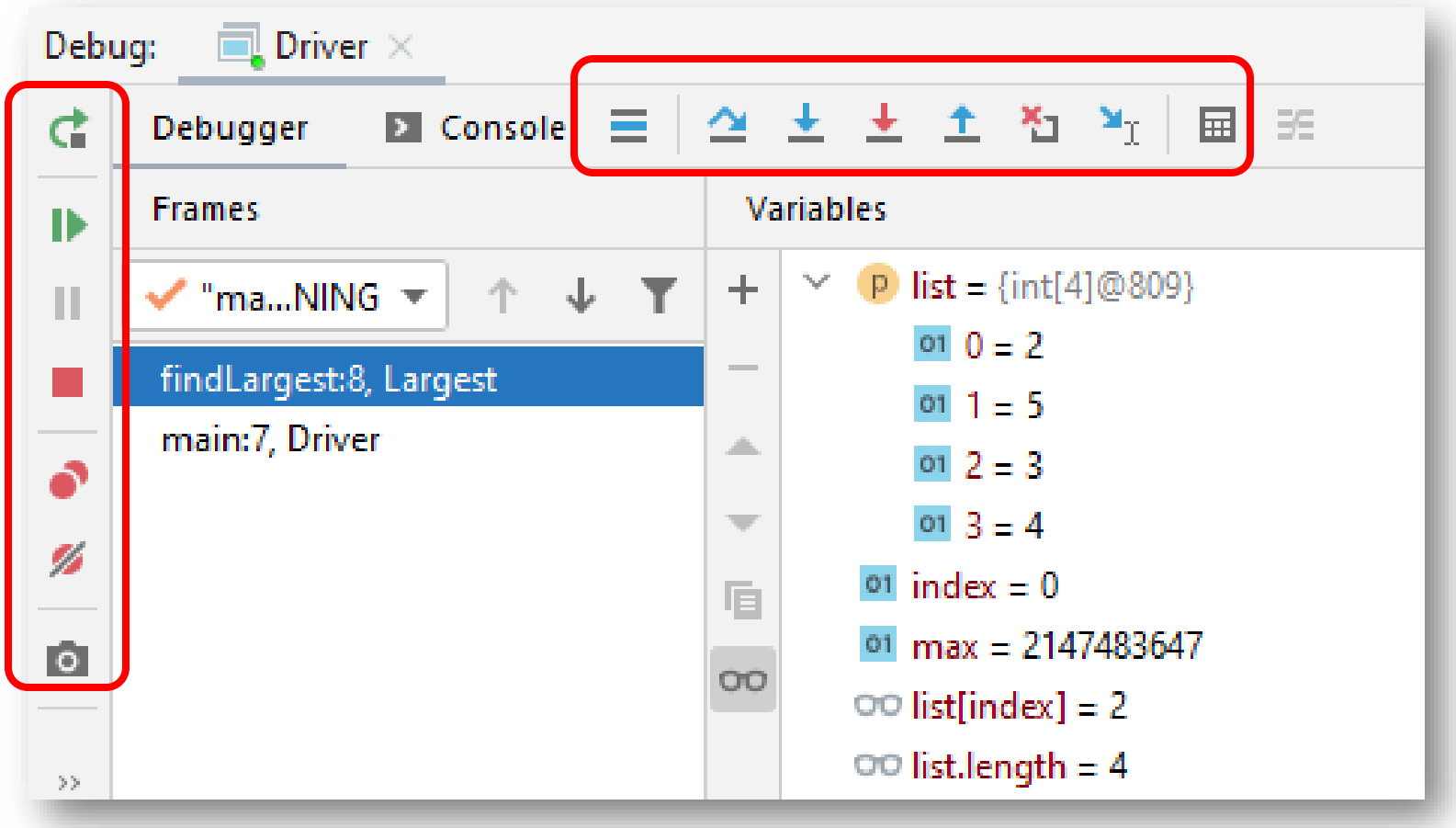
```
int max = Integer.MAX_VALUE;
```

- We need:

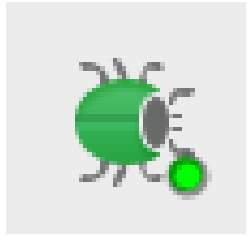
```
int max = list[0];
```



Some IntelliJ debugger buttons…

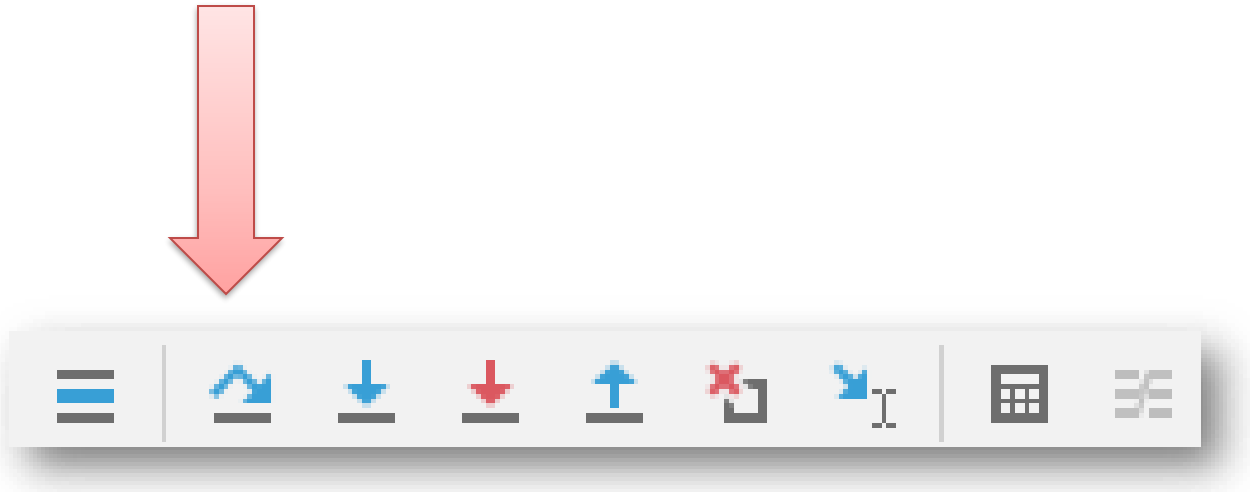


Some Debugger buttons…

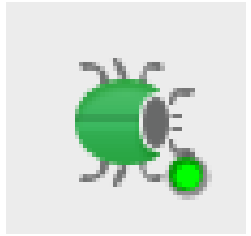


- **Step Over** 

  to step over the next method call (without entering it) at the currently executing line of code.

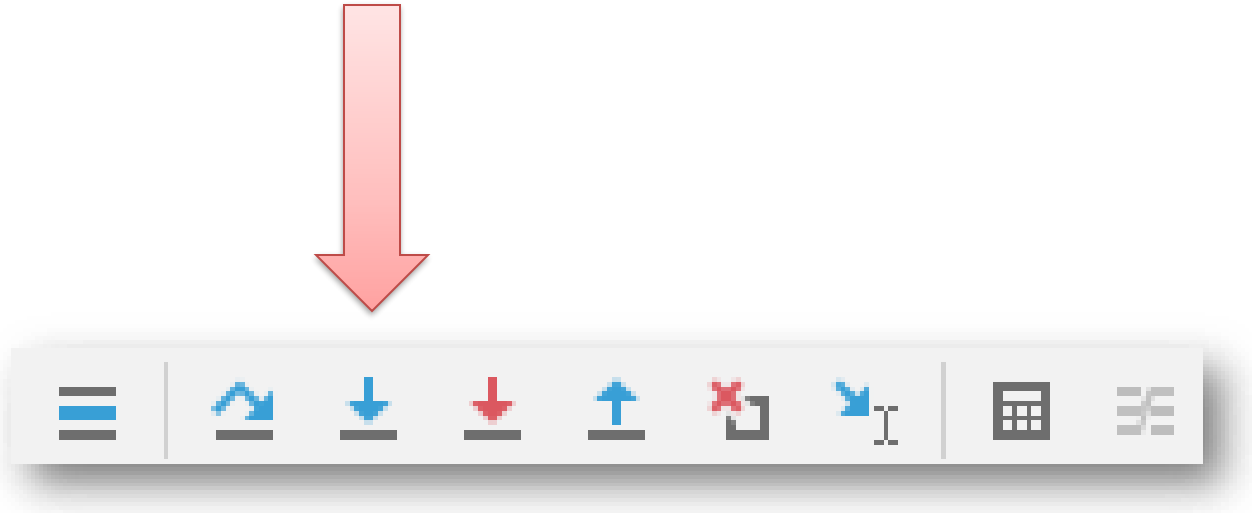


Some Debugger buttons…

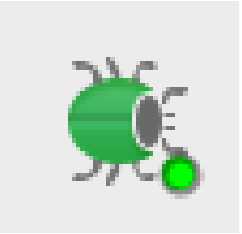


- **Step Into** 

  step into the next method call at the currently executing line of code.



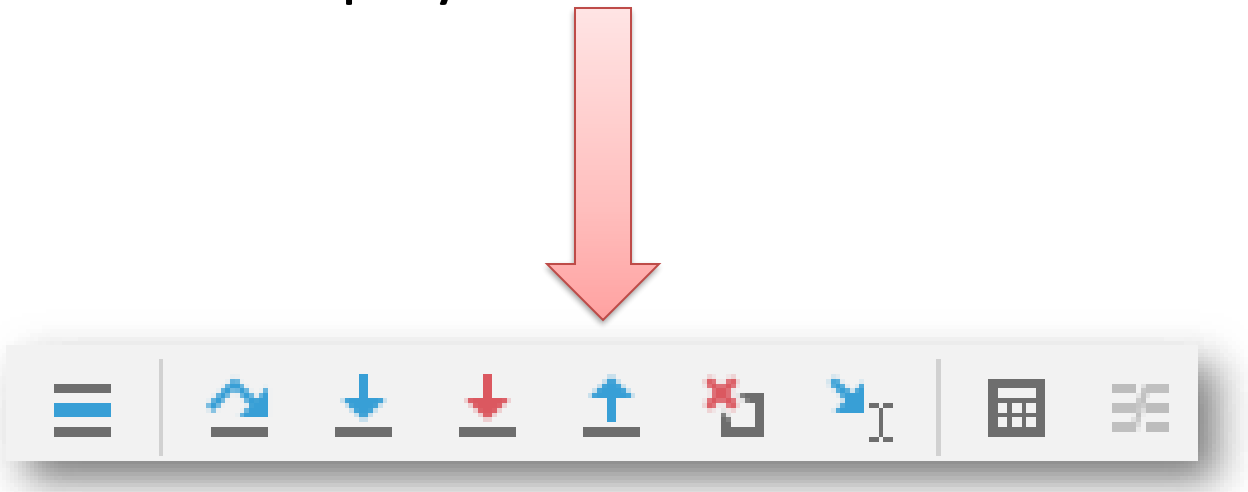
Some Debugger buttons…



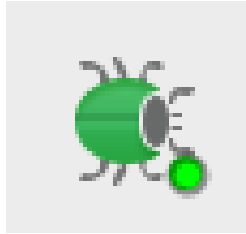
- **Step Out** 

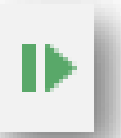
  executes the remaining lines of a method in which the current execution point lies.

  The next statement displayed is the statement following the method call.

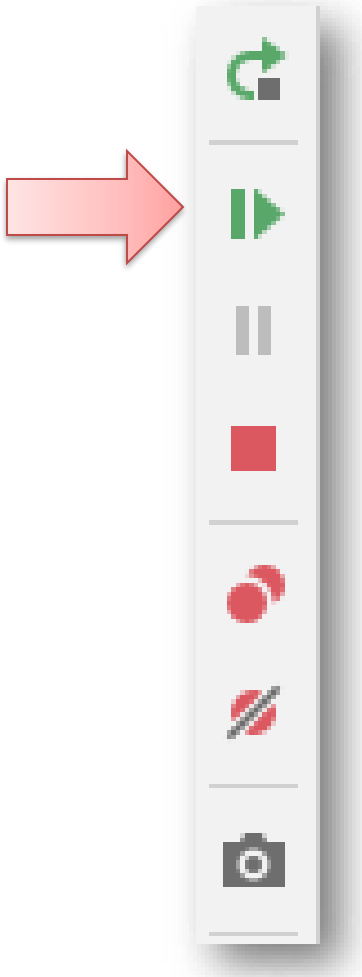


Some Debugger buttons…

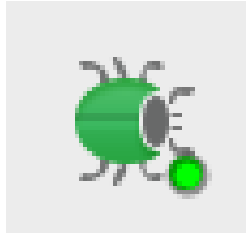


- **Resume Program** 

  resume the execution of the currently suspended program; it will stop again at the next available breakpoint (if any).

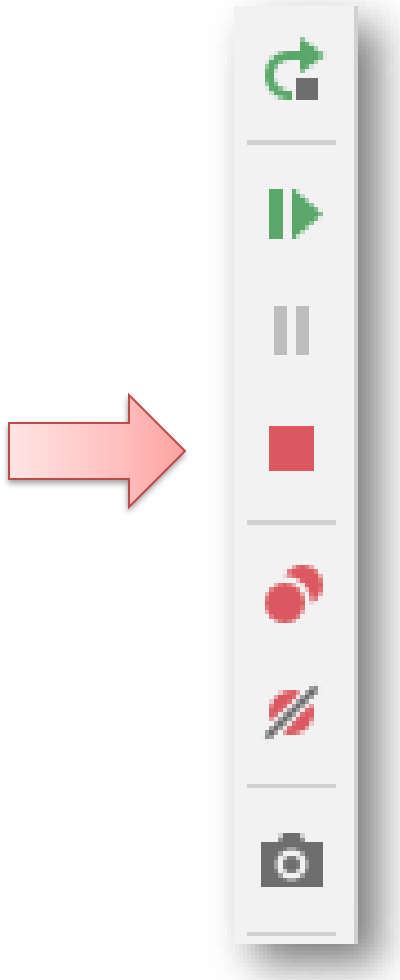


Some Debugger buttons…

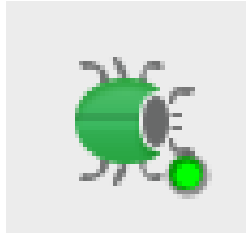


- **Terminate** 

  to terminate the launch associated with the selected debug target
  i.e. **stop** the program.

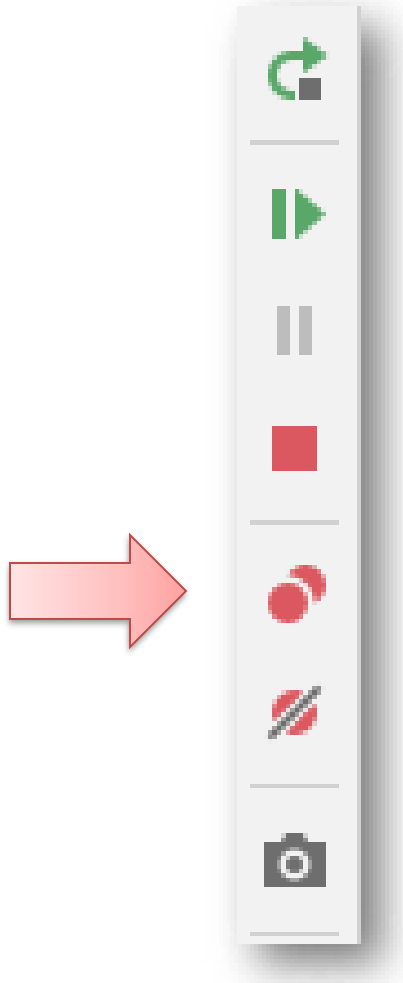


Some Debugger buttons…

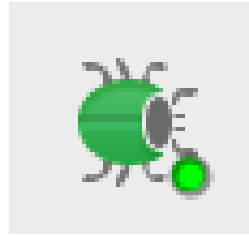


- **Show breakpoints** 

  show all the breakpoints (in a pop up window) in the program.

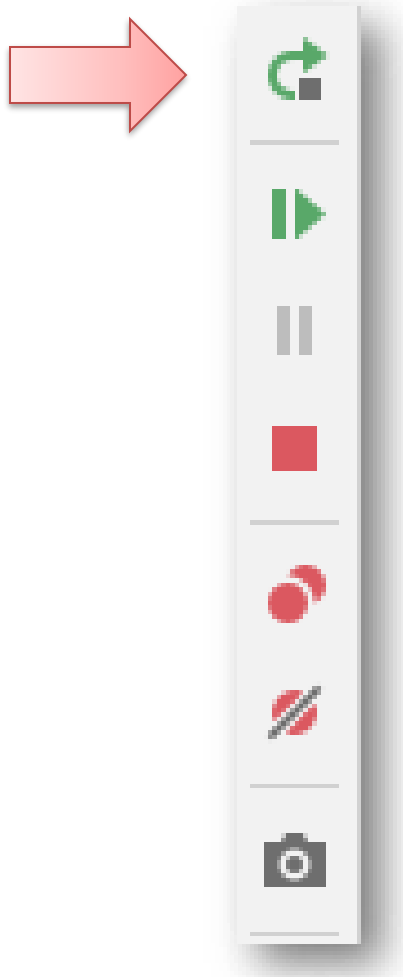


Some Debugger buttons…



- **Rerun program** 

  start the program again.



Now it’s your turn!

– practice using the debugger on this code (there is a step in your labs to provide support).

Any Questions?