

# CS 1103-01 Programming 2 - AY2022-T5

[Dashboard](#) / [My courses](#) / [CS 1103-01 - AY2022-T5](#) / [7 July - 13 July](#) / [Lab 8 Unit 4](#)

## Lab 8 Unit 4

### Lab 8: Debugging

This week's lab will introduce you to two programming tools: The CVS version control system and the Eclipse debugger. The assignment (Part 2 of the lab) will be to submit a lab report about your experience with the debugger.

### Buggy Search and Sort

The code directory contains the file `BuggySearchAndSort.java`. Copy this file into an Eclipse project.

*BuggySearchAndSort* defines a search subroutine that is supposed to check whether or not a given integer occurs in a given array of integers. It also contains three different sorting subroutines that are supposed to be able to sort an array of integers into non-decreasing order. There is also a main program that tests the four subroutines (along with a sorting subroutine from one of Java's standard classes that is there to remind you that you don't necessarily need to write your own subroutines to perform common tasks).

If all the subroutines that are defined in *BuggySearchAndSort* were correct, then running the program would produce output that looks something like this:

The array is: 4 3 6 9 3 9 5 4 1 9

This array DOES contain 5.

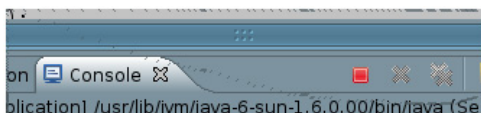
Sorted by Arrays.sort(): 1 3 3 4 4 5 6 9 9 9

Sorted by Sweep Sort: 1 3 3 4 4 5 6 9 9 9

Sorted by Selection Sort: 1 3 3 4 4 5 6 9 9 9

Sorted by Insertion Sort: 1 3 3 4 4 5 6 9 9 9

Unfortunately (or fortunately, for the purposes of this lab), each of the four subroutines in *BuggySearchAndSort* has a bug that can either produce incorrect results or an infinite loop. If you run the buggy program without modifying it, it will go into an infinite loop. Remember that to stop a running program in Eclipse, you can click the Stop button that looks like a small red square in the part of the window where the Console view appears:



(If this button is gray instead of red, it means that no program is currently running in the visible console. However, there might be other consoles that are still running programs; to avoid having a bunch of programs, all running in infinite loops at the same time, you should make sure that a program has been terminated before starting another program run.)

**Your job in this lab is to use the Eclipse debugger to find all the bugs in *BuggySearchAndSort.java*. In the written lab report that you will turn in, you should describe each bug, say how you fixed it, and explain briefly how you used the Eclipse debugger to find the bug.**




(Note: It's possible that you could find some or all of the bugs just by inspecting the code. However, the main purpose of this part of the lab is to get some experience with the debugger, so I would like you to do that rather than simply eyeball the code!)

To debug a program in Java, you have to start the program using a "Debug" command instead of a "Run" command. There is a "Debug As..." command in the same pop-up menu as the "Run As..." command. You can also click the "Debug" button to debug the program that you have run or debugged most recently. The "Debug" button is just to the left of the "Run" button in the toolbar, and it looks like a bug:



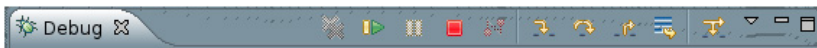
The main point of the debugger is that it allows you to pause the execution of a program and inspect the values of variables. Once the program has been paused, you can also step through the program line-by-line to see what is going on. The program will not pause, however, unless you have added a breakpoint to the program source code. If you run a program that has no breakpoints under the debugger, it behaves just as it would if it were run normally.

To add a breakpoint, simply double-click in the left margin of the source code, next to one of the lines of code. (By left margin, I mean the same part of the window where the error and warning "light bulbs" appear.) The breakpoint appears as a small dot: . You can remove the breakpoint by double-clicking this dot. There is also a "Remove All Breakpoints" command in the "Run" menu. I should also note that you can use the "Add Java Exception Breakpoint" command in the "Run" menu to set up a breakpoint that is triggered whenever an exception of a given type occurs.

When a program is running in Eclipse under the debugger, and the execution gets to a line that contains a breakpoint, the execution is paused. If Eclipse is showing the regular Java perspective when this happens, Eclipse will want to switch to the Debug perspective -- another screen full of views (or panes) that are useful for debugging. The first time this happens, Eclipse will put up a dialog box that asks whether you want to switch to the Debug perspective. **I suggest that you check the box that says "Remember my decision" and say "Yes".**

Debugging is complicated, and so is Java's Debug perspective. The most important view is the "Variables" view in the top-right corner. Here, you can inspect the values of variables. An array or object variable has a small triangle next to its name; you can click this triangle to open and close the variable so that you can inspect its contents. If a variable is selected (by clicking it), its value is also shown at the bottom of the view.

The "Debug" view in the top right contains a list of currently active subroutines. Click on a subroutine to inspect its variables. The title bar of the "Debug" view contains some useful buttons:



The green triangle means "Resume" and can be used to resume normal execution of the program (until it next encounters a breakpoint). There is a red "Stop" button that can be used to terminate the program. Most important are the three "arrow" buttons for the "Step Into", "Step Over", and "Step Out" commands. "Step Over" is easiest to understand -- clicking it causes one line of code to be executed. "Step In" is similar, except that if the line of code contains a subroutine call, it takes you to the first line of the subroutine so that you can continue from there. "Step Out" continues execution until the current subroutine returns, and it stops at the line to which the subroutine returns. Note that these "Step" commands can also be found in the "Run" menu and that the **F5, F6, and F7** keys can be used as keyboard equivalents for the step commands.

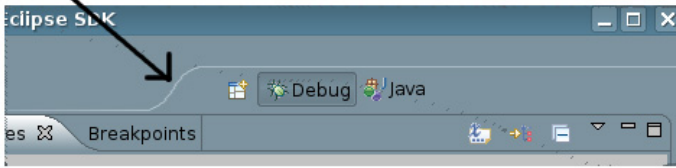
In the left-center area of the Debug perspective, you'll find an edit view where the source code files for the program are displayed. The current line is marked in this view with an arrow in the left margin, and you can watch this arrow move as you step through the program. **You can even edit your program in the Debug perspective.**

One final remark: There is a list of open perspectives on the right end of the toolbar at the top of the Eclipse window. You can move from one perspective to another by clicking on the name of one of the perspectives on this list. So, when you want to get out of the Debug



perspective and back to the normal Java perspective, just click "Java" in the list of perspectives:

Drag this separator left or right to change the amount of space available for perspective names



Last modified: Wednesday, 13 May 2020, 1:14 PM

## UoPeople Clock (GMT-5)

All activities close on Wednesdays at 11:55 PM, except for Learning Journals/Portfolios which close on Thursdays at 11:55 PM **always following the clock at the top of the page.**

Due dates/times *displayed* in activities will vary with your chosen time zone, however you are still bound to the **11:55 PM GMT-5** deadline.

◀ Lab 7 Unit 4

Jump to...

Solutions for Assignment Unit 3 ▶

[Disclaimer Regarding Use of Course Material](#) - [Terms of Use](#)

University of the People is a 501(c)(3) not for profit organization. Contributions are tax deductible to the extent permitted by law.  
Copyright © University of the People 2021. All rights reserved.

You are logged in as [Samuel Katongole](#) ([Log out](#))

🌐 [www.uopeople.edu](http://www.uopeople.edu)



### Resources

[UoPeople Library](#)

[Orientation Videos](#)

[LRC](#)

[Syllabus Repository](#)

[Honors Lists](#)

### Links

[About Us](#)

[Policies](#)

[University Catalog](#)

[Support](#)

[Student Portal](#)

### Instructors

[Instructor Portal](#)

[CTE Center for Teaching Excellence](#)

[Office 365](#)

[Tipalti](#)

### Contact us

[English \(en\)](#)

[English \(en\)](#)

[العربية \(ar\)](#)



[Data retention summary.](#)

[Get the mobile app](#)

