The Eclipse Debugger

Earlier, we explored debugging a buggy program (Debug.java) by adding temporary System.out.println() statements to check the value of variables at various points in the code. Another useful technique is to use a debugger to verify the execution “flow” and to see what the values of variables are at any point in the code.

We can explore using the Eclipse debugger to find and fix bugs by “stepping” through the code. Here are the general steps to get started debugging in Eclipse:

1. Open the file you want to debug, e.g., Debug.java.
2. Find the line where you want to place a breakpoint (a breakpoint is a point in your code where you want to temporarily stop execution of your program).
3. On the desired line, right-click near the left-margin of the edit window and select “Toggle Breakpoint”. Your program will now stop execution at that line when you run in Debug “mode”.
4. On the main menu, select Run -> Debug to start running the debugger.
5. Eclipse will now want to switch to the “Debug Perspective”. Agree to this and Eclipse will run your program and then stop when it reaches a breakpoint.
6. At this point, Eclipse will present you with several different windows designed to help you debug. The main windows that you’ll use are:
   a. the main “active” source code window that shows you where you are in the flow of execution (left center center),
   b. the call stack window (upper left window)
   c. the “watch” window where you can see the current values of variables (upper right window)
7. When your program is stopped at a breakpoint, you can take your time and examine current variable values in the watch window to see if they match what you expect them to be. You can also “hover” your mouse over a variable and see the current value.
8. If you want step execution one line at a time, you can select on the main menu:
   a. Run -> Step Into: this allows you to step into a method call
   b. Run -> Step Over: this allows you execute the method call without stepping into it
9. You can also just continue running until your program hits another breakpoint or exits by selecting Run -> Resume.
10. When you are done debugging select Run -> Terminate, and then you can switch out of the Debug Perspective and back to the regular Java Perspective by clicking the Perspective buttons in the upper right of Eclipse.
11. Note that there are shortcuts for all the menu choices so you can debug more efficiently.

If you need help, see one of the instructors or the tutors.