Debugging

Debugging

- Programming is difficult!
- When you have errors, how do you find them?
  - Use System.out.println (slow, painful way)
  - Use a debugger (much better way)

Debugging

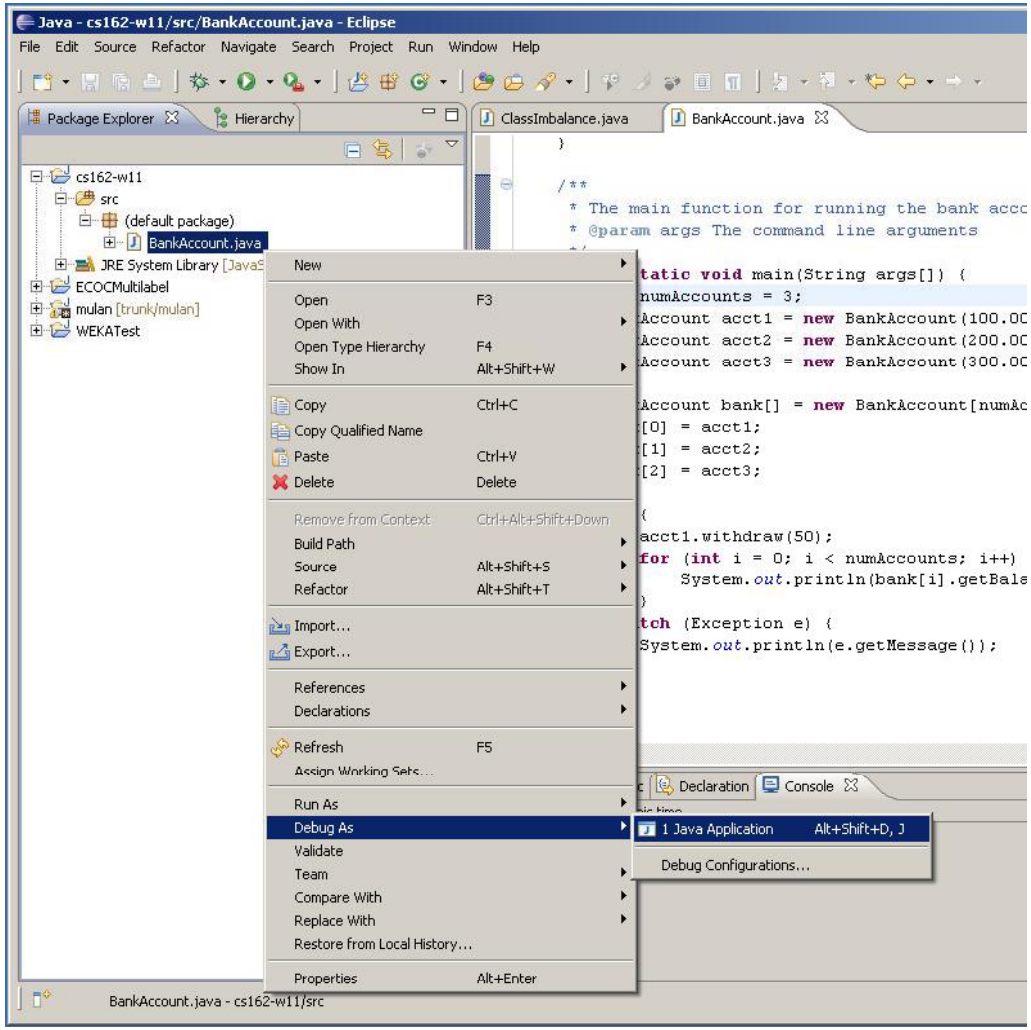
Things to learn:

- Breakpoints
- Commands: Resume, Terminate, Step over, Step into, Run to line
- Variables / Expressions Window
- Call stack

How to Debug

To start the debugger

- Right-click on the Java file (BankAccount.java) with the main function
- Select Debug as…→ Java Application
- You may be asked to switch to the Debug Perspective. Just click on yes.



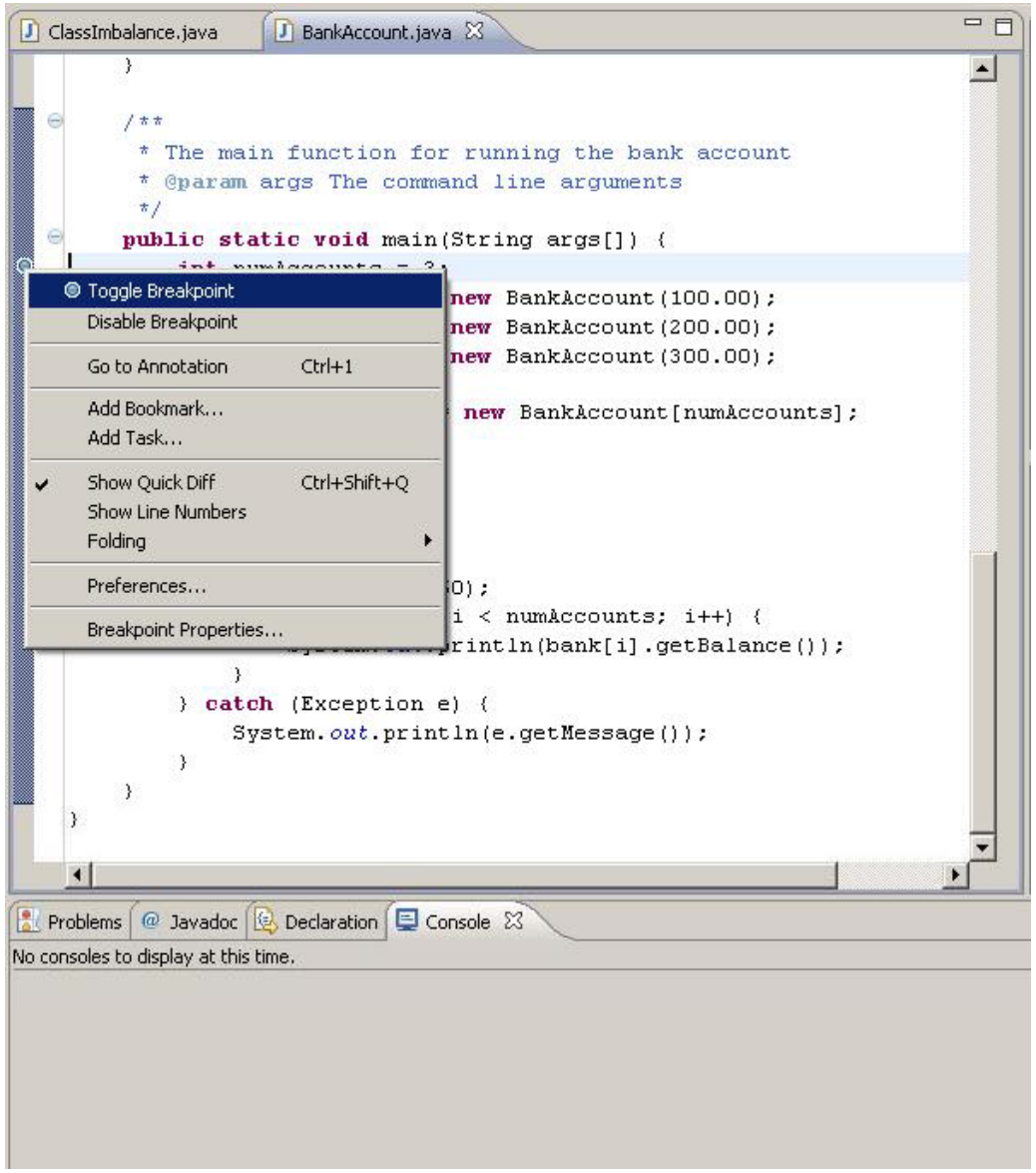
Breakpoints

- If you follow the steps from the previous slide, you probably will just run through the entire program and not stop
- You need to put a breakpoint in your code
- Breakpoint: A place in your code where you would like to pause while debugging

Breakpoints

To place a breakpoint, right click in the margin on the line you want to pause at. Select “toggle breakpoint”. You should see a blue circle.

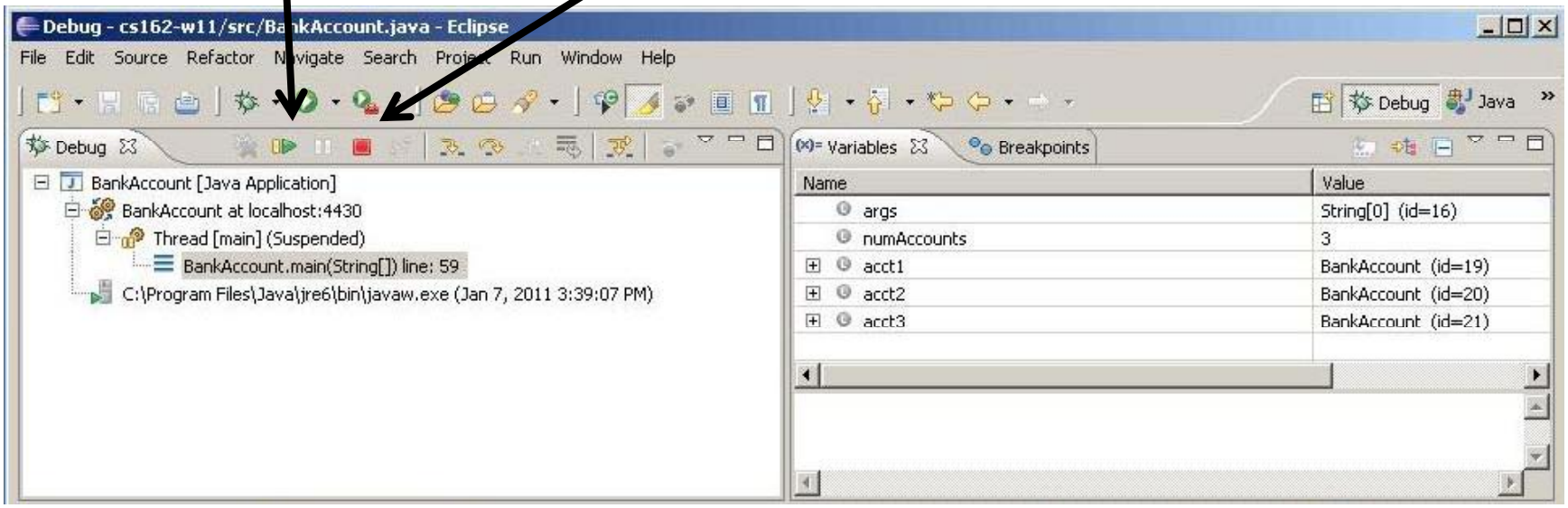
To remove a breakpoint, click on the blue circle and select “Toggle breakpoint” again.



Key Commands

Resume: run until you hit the next breakpoint or your program ends

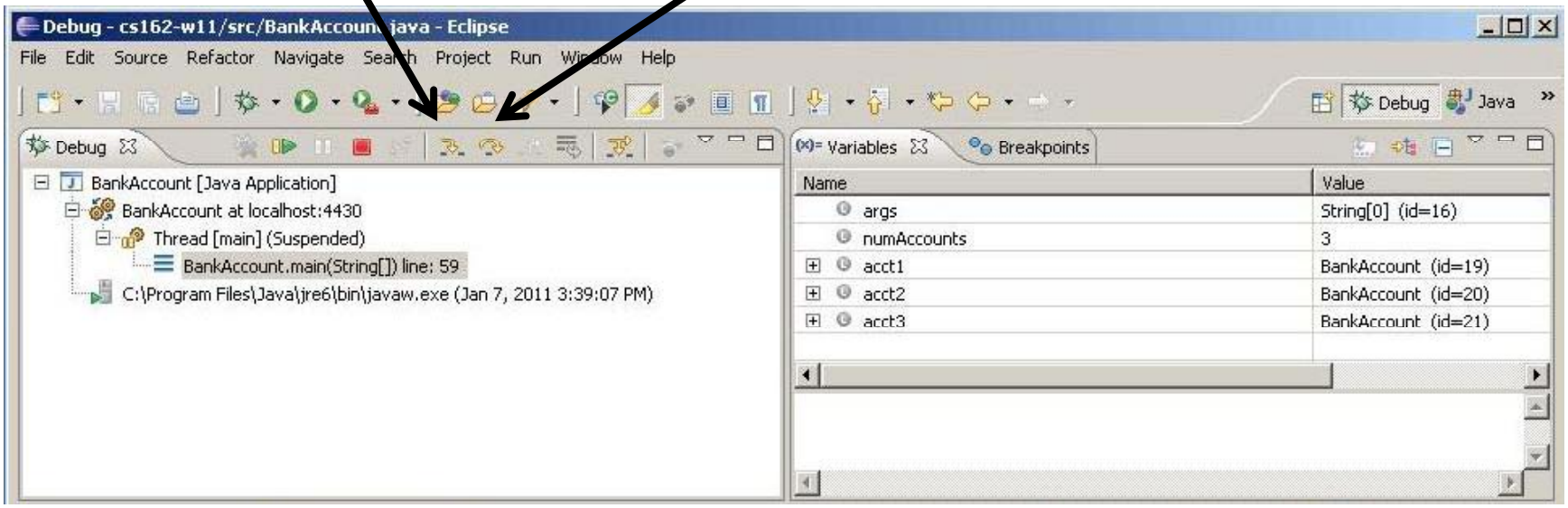
Terminate: stop the program



Key Commands

Step into: go into the function call on the line you are on

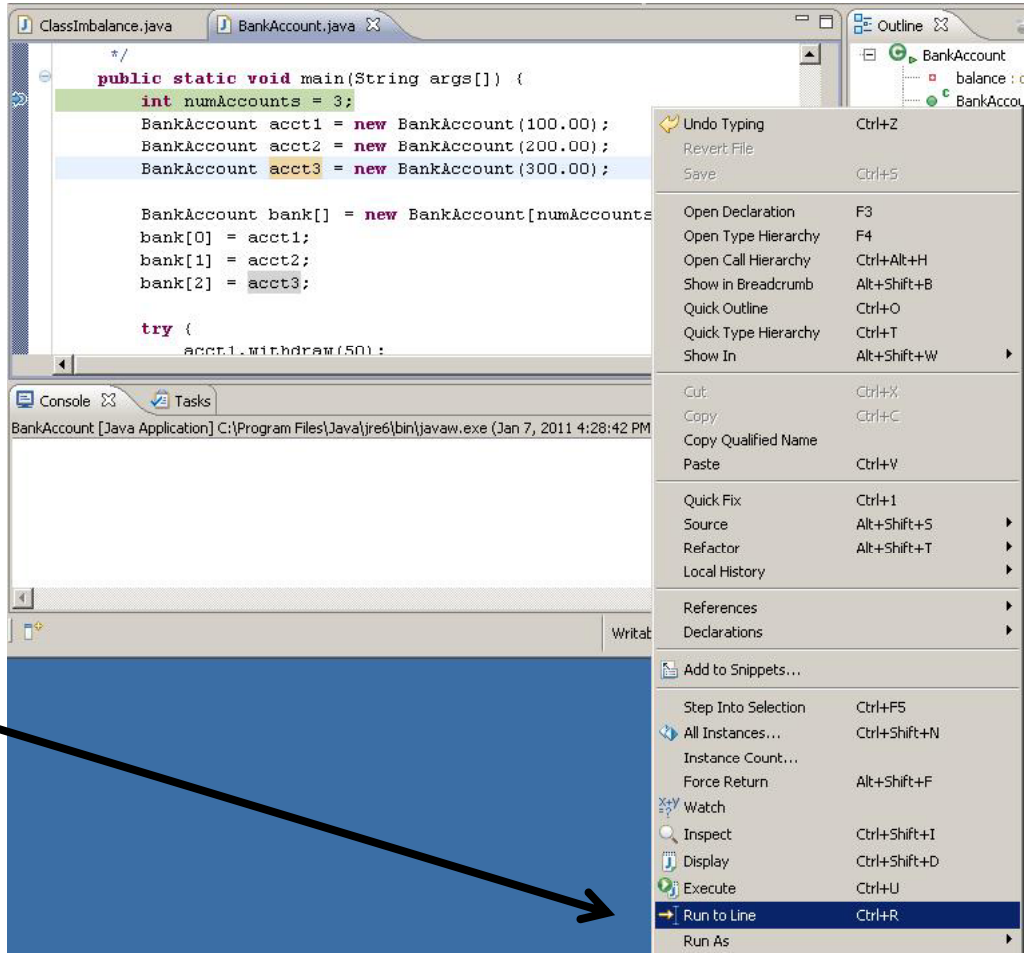
Step over: don't go into the function call on the line you are on



Key Commands

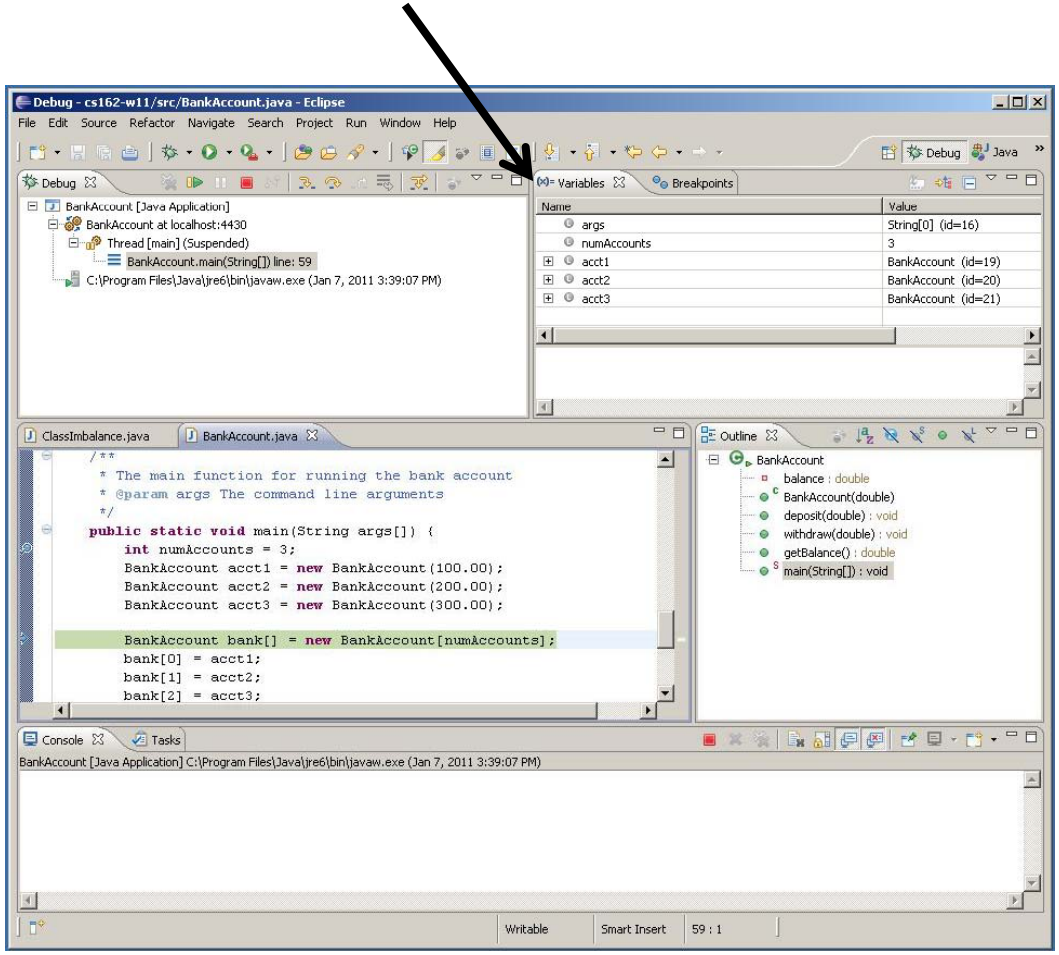
You can also execute up to a given line. To do so:

- Right click on the line you want to stop at (shown in light blue)
- Select **Run to Line**



Inspecting variables

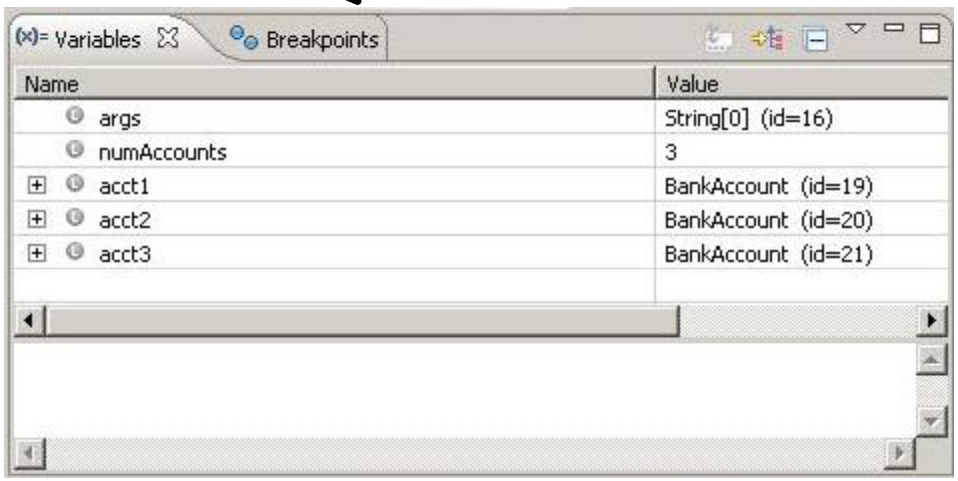
In debug mode, look at the window at the top right



Inspecting variables

Variables tab: can inspect the values of variables

Breakpoints tab: can see all the active breakpoints

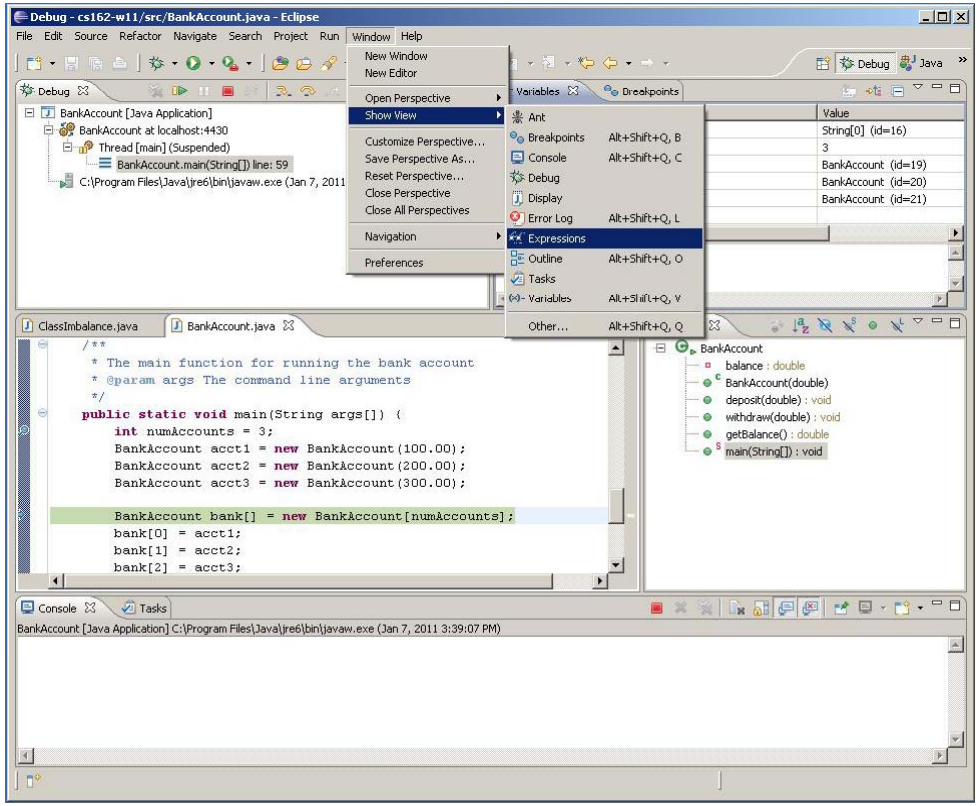


Expressions tab: (not shown here – may need to add this) can evaluate the results of certain expressions

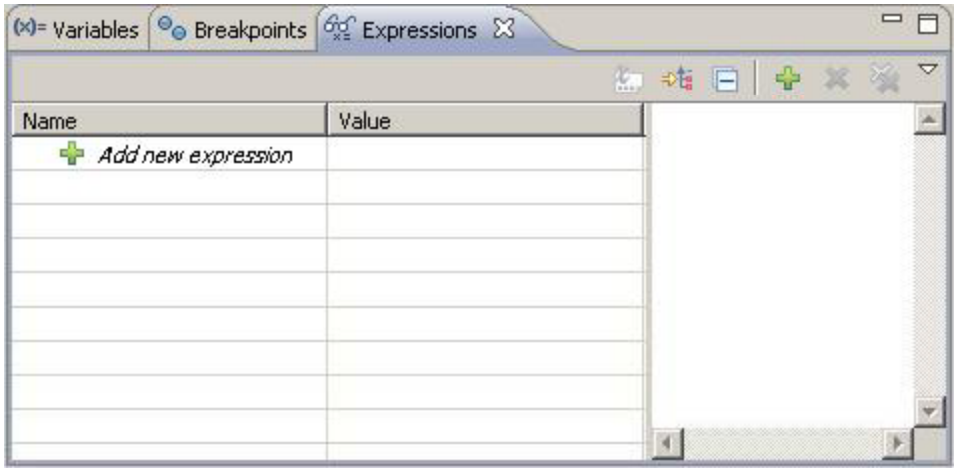
Inspecting Expressions

To bring up the Expressions tab:

- Click on Window -> Show View -> Expressions



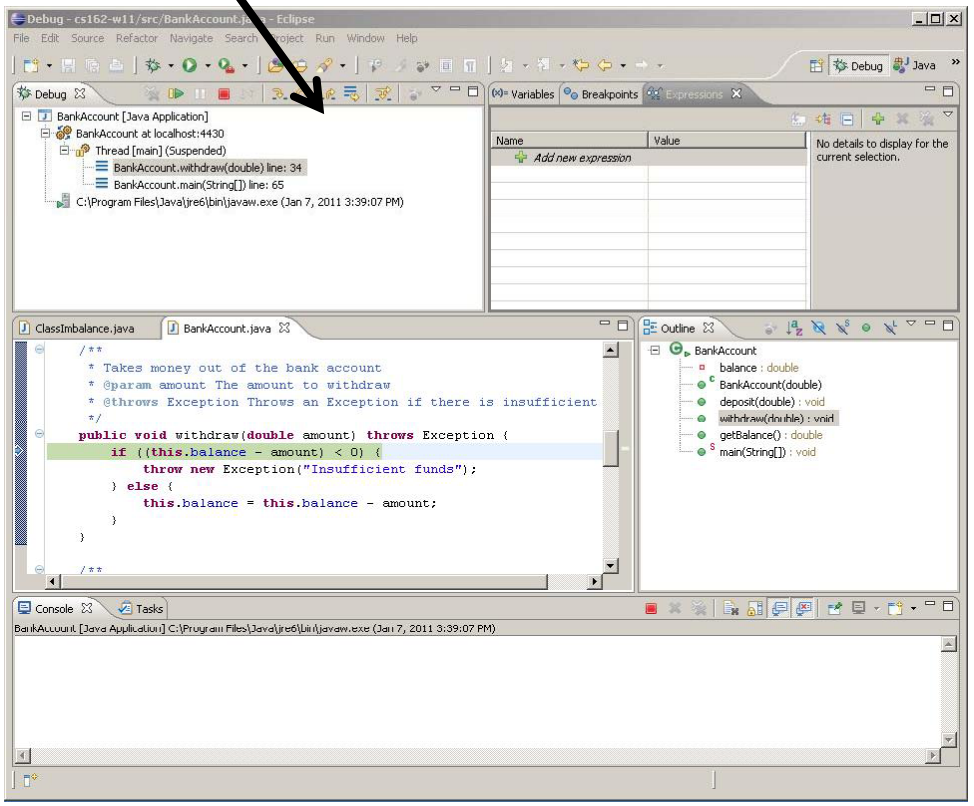
Inspecting Expressions



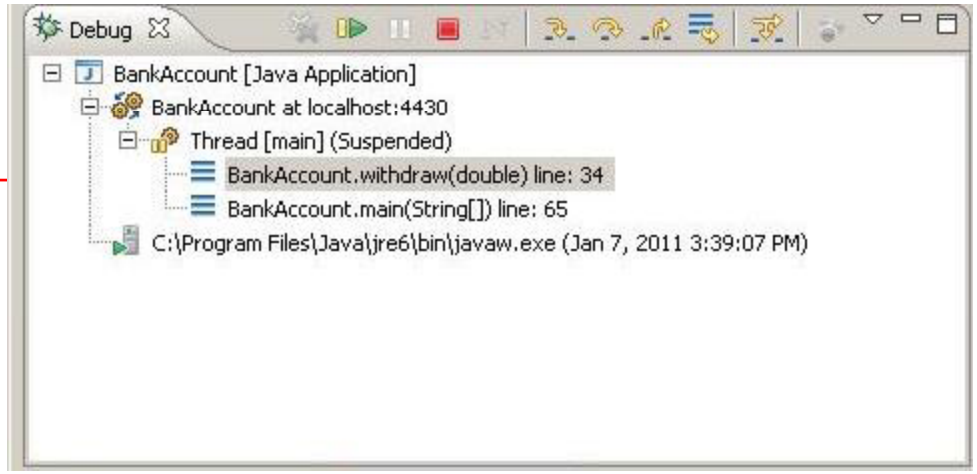
- If you click on the “Add new expression” part, you can add any expression you would like to evaluate as the code is running
- An expression can be a variable, a bunch of operations on variables, a function call, etc.
- This is a very useful tool!

The Call Stack

- At the top left of your debug perspective, you'll see a window with "Debug" as the title.
- This is called the call stack window.



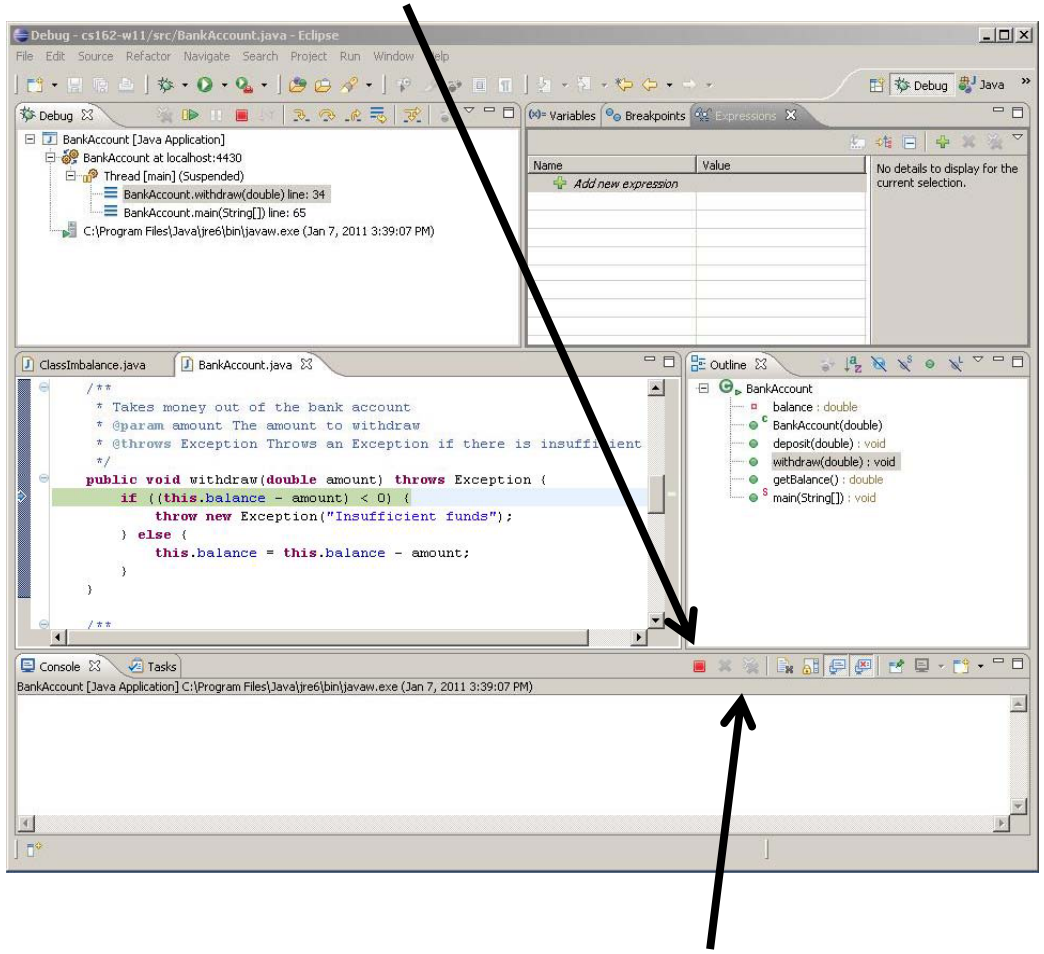
The Call Stack



- These are the current “stack” of functions.
- This example shows that you were in the BankAccount.main() function which then called the BankAccount.withdraw(double) function.

Finally

Remember to terminate (hit the red square)



And clean up your threads (hit the two black Xs)

Steps in Debugging

- Reproduce the error. Make sure you can show it repeatedly.
- Simplify the error. Find the simplest test case or set of values that show the error.
- Divide and conquer. Use breakpoints, print statements, to locate where the error occurs
- Know what your program should do.

Steps in Debugging, cont

- Keep an open mind. "When you have eliminated the impossible, whatever remains, however improbable, must be true"
- Look around. If you don't see anything at the point the error is manifest, the mistake is probably earlier.
- Look at the details. The simple errors are the hard ones to see.