Dr. Emily Hill CSCI 151

TUTORIAL: ECLIPSE DEBUGGING DEMO

This tutorial assumes you have downloaded the SortingB.zip file from BB and imported the SortingB project into Eclipse. To import the project from the zip file by selecting File > Import and then General > Existing Projects into Workspace. Hit "Next", then "Select archive file," browse to SortingB.zip, and hit "Finish".

The steps below will show you how to use the Eclipse debugger to step through an execution of IterativeSorter.

Displaying Line Numbers

1. Before we begin, it is helpful to have Eclipse display line numbers. Turn on line numbers by right-clicking in the editor window and selecting "Preferences"

😑 🔿 🕐 Java – SortingB/src/IterativeSorter.java – Eclipse – /Users/EmilyHill/Dropbox/In progress/CMPT 184/workspace 💬					
] 🕆 + 📄 🗟 📄] 🏊 + 🍫 + 🕗 + 🎥 🞯 +] 😕 😂 🖋 +] 🍄 🅖 🐲 🗐 🗊] 💱 + 🏹 + 🏷 + 🗇 +					
Package Explorer S Package Explorer S BatchCalculator RectangleTesting Sorting SortingB SortingB (default package) IterativeSorterTest.java RecursiveSorterTest.java RecursiveSorterTest.java RecursiveSorterTest.java RecursiveSorterTest.java RecursiveSorterTest.java RecursiveSorterTest.java RecursiveSorterTest.java Stream Library (JavaSE-1.6) JUnit 4 TestingHW.doc	<pre></pre>	 I rectangle Test.java I Rectangle Test.java I Rectangle Test.java I Rectangle Test.java Open Declaration Open Type Hierarchy Open Call Hierarchy Open Call Hierarchy Show in Breadcrumb Quick Outline Quick Type Hierarchy Show In Cut Copy Copy Qualified Name Paste Quick Fix Source Refactor 	"s □ #Z #Z #S F3 F4 ^\TH \C#B #O #C #C #X #C	Debug Java ** Dutline & Java ** Dutline & Java ** TextiveSorter a ray : int[] a ray : int[] a filterativeSorter(int[]) a contexter(int[]) a setArray(int[]) * void getSortedArray0 : in getOriginalArray0 : in getOriginalArray0 : in getOriginalArray0 : in sort0 : int[] a toString0 : String prettyPrint(int[)) : void S main(String[]) : void	
□ ◆	26 return sortedArray; 27 } 28	Local History References Declarations Market Add to Snippets Run As Debug As Coverage As Validate Team Compare With Replace With WikiText Preferences			

2. Next, select "Text Editors"

\varTheta 🔘 🔘	Preferences (Filtered)	
type filter text	Editor	(-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,-,
► Ceneral V Java ► Editor	Java editor preferences. Note that some prefere Celtors preference page. Show the shared text editor preferences mart carec posttooming in gava manues over Also see the compiler warnings and spell of Highlight matching brackets Light bulb for quick assists Only show the selected Java element Appearance color options: Matching brackets highlight Parameter hints background Parameter hints foreground Completion overwrite foreground Source hover background Source hover background	nces may be set on the Text rrides platform behavior) thecking preferences.
?	C	ancel OK

3. Make sure the check box next to "Show line numbers" is selected and hit "OK"

: 💭 💛 💟	Preferences (Filtered	d)
type filter text	Text Editors	↓ ↓ ↓
▼General ▼Editors	Undo history size:	200
Text Editors	Displayed tab width:	4
♥ Java ▶ Editor	Insert spaces for tabs	
	ighlight current line	
	Show print margin	
	Print margin column:	80
	Show line numbers	
	Show range indicator	
	Show whitespace characters	
5	Show affordance in hover on	how to make it sticky
	When mouse moved into hover:	Enrich after delay
	Enable drag and drop of text	t
	🗹 Warn before editing a derive	d file
	🗹 Smart caret positioning at lir	ne start and end
	Appearance color options:	
	Line number foreground	Color:
	Current line highlight	
	Find scope	
	Selection foreground color Selection background color	
	Background color	
	Foreground color Hyperlink	
	More colors can be configured o	on the Colors and Fonts preference page.
		Restore Defaults Apply
(?)		Cancel OK
<u> </u>		

Debugging Demo

1. Open the IterativeSorter class and scroll down to its main method



2. In the gray margin to the left of the line numbers, right click next to line 63 and select "Toggle Breakpoint":

Java - SortingB/src/IterativeSorter.java - Eclipse - /Users/EmilyHill/Dropbox/In progress/CMPT 184/workspace				
] 📬 • 🔛 🗟 👜] 🎴 • 🕸 • 🔘 • 🤇	L •] 😫 🞯 •] 🥙 😂 🖋 •] 🍄 🕖 😂 🖩 🗊] 🖢 • 🖓 • 🏷 • 🔶 •	😭 🏇 Debug 🐉 Java 🔌		
📕 Package Explorer 🕱 🛛 🗖	🚺 RecursiveSorter.java 🚺 IterativeSorter.java 🕱 🚺 RectangleTest.java 🎳 🏷 🗖 🗖	🗄 Outline 🛿 🗖 🗖		
Package Explorer S Package Explorer S Package Explorer S Package Explorer S Package Sorting Sorting Sorting Package Sorter, Java Pack	<pre>2 Rectangue is a in the interview of the interview o</pre>	<pre>butine &</pre>		
] 🗗	Writable Smart Insert 35 : 46			

When the breakpoint is set, you should see a blue circle next to line 63:



or by clicking on the green bug icon to the left of the green play arrow



4. Eclipse will ask if you'd like to open the debugging perspective, select "Yes" (I recommend selecting the check box so this becomes the default decision.

5. The debugging view has a number of panes:

Debug – SortingB/src/IterativeSorter.java -	Eclipse – /Users	s/EmilyHill/Drop	box/In progress/C	MPT 184/workspace
] 😷 + 🖫 🐚 🍙] 🂁 + ॐ + Ø + ௸ -] ॐ 🖉 🔗 +] ॐ 🖉 🋐] 灯 + 🖗 + ۞ + ↔ + 👔 🎊 Debug 🖏 Java »				
Image: Second	orter)) Iome/bin/java (F	🗣 Variables 🔎 Bi	reakpoints 🛿	¥ ⅔ ℛ ⊴ `
<pre> P RecursiveSorter.java I IterativeSorter.java I Rectangle S1 S20 private String prettyPrint(int[] a) { S3 S4 for (int i : a) S5 s += i + ""; S6 return s; S7 } S8 public static void main(String[] args) { </pre>	java 📝 Rectar	ngleTest.java	<mark>»4 □</mark>	E Outline ⊠ □ □
<pre>60 // Automate running, but not testing 61 int[] array = {5, 67, 12, 20}; 62 IterativeSorter s = new IterativeSorter(arr \$0.63 s.sort(); 64 System.out.println(s); // uses Sorter.toStr 65 } 66 67 }</pre>	ay); ing			 getSortedArray(): int[] getOriginalArray(): int[] sort(): int[] toString(): String prettyPrint(int[]): String ^S main(String[]): void
Image: Second State Image: Second State IterativeSorter (1) [Java Application] /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/Java (Feb 15, 2011 4:44:57 PM) 5				
] 0*	Writable	Smart Insert	63 : 1	

Pane 1: in the top left we have the debugging view that shows us the call stack and gives us buttons to control the execution

Pane 2: in the top right we can control what breakpoints are set, and by clicking on "Variables", see what values our variables have during execution

🕅 Variables 🖾 💩 Breakpoints	‱ ⇒ti 🕞 🗖
() args	String[0] (id=15)
Ø array	(id=17)
▶	IterativeSorter (id=19)
[5, 67, 12, 20]	

For example, we see that our array contains the values [5, 67, 12, 20] -- just as we set them in line 61

Pane 3: what line we are in the process of executing is highlighted in the source code window. We can also edit our code on the fly (such as to fix bugs), and Eclipse will do its best to continue execution from the beginning of the current method call.

Panes 4 & 5 are the same console and outline views as we have in the java perspective, where we edited and tested code previously

- 6. Let's focus for a moment on the buttons in the debugging view:
 - Resume: allows us to resume execution until the next breakpoint is reached
 - Terminate: immediately halts execution (same function as in the Java perspective)

• Step Into: allows us to step into the execution of a method, so we can watch which lines are executed and how the values of variables change

• Step Over: moves on to the next statement in the method we're executing; if the method is completed, step over behaves the same as step return

• Step Return: when inside method calls other than main, allows us to finish executing the current method and return back to the caller

7. Now let's see what happens when the sort method executes, by pressing "step into"
 Debug - SortingB/src/IterativeSorter.java - Eclipse - /Users/EmilyHill/Dropbox/In progress/CMPT 184/works

] 📬 🖫	ि 🔄] 🎦 र 🏇 र 🔕 र 隆 💋 🖉 🖉 🖉 🖉 🖉 🖉 🖉 🗐 🗍 🖢 र 🖗	▼		
🏇 Debug	X 🕺 🕨 🗉 🖬 🕺 👁 🗈 👘 🖉 💱 🖓 🖓	🗱 🕫 Variables 😫 🖓 Breakpoints		
🔻 🗊 lter	ativeSorter (1) [Java Application]	▼ ⊕ this	IterativeSorter	
	IterativeSorter at localhost:57674	🕨 🗉 array 🛛 🚽 🛶 🛶 🛶 🛶	(id=17)	
	P Thread [main] (Suspended)	sortedArray	null	
	IterativeSorter.sort() line: 34			
	IterativeSorter.main(String[]) line: 63			
	/System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (F			
		FF (7 12 207		
		[5, 67, 12, 20]		
<u> </u>				
Recurs	iiveSorter.java 🛛 🕖 IterativeSorter.java 🕱 🕖 Rectangle.java 📄 🕖 Rec	tangleTest.java 🔭 🖓 🖓 🖓	E Outline	
32		6		
330	330 public int[] sort() {			
⇒ 34 sortedArray = array.clone();				
35 for(int i = 0; i < sortedArray.length; i++) {			sortec	
36 int min = i;			array	
37 for(int j = i + 1; j < sortedArray.length; j++) {			⊜ ^c Iterati	
38 if(sortedArray[j] < sortedArray[min]) min = j;			🛛 🖉 🖉 🖉	
39	}		setArr	
40 int t = sortedArray[min];			getSo	
<pre>41 sortedArray[min] = sortedArray[i];</pre>			getOr	
<pre>42 sortedArray[1] = t;</pre>			sort()	
44 return sortedArray;				
45	1		S S S S	
40	nublic String toString() {	Ă	● ⁻ main(:	
48	return "Original: " + prettyPrint(getOriginalArray()) +	"\n" +		

We see a number of changes:

• Our editor has changed to the sort method, and the first line is highlighted, ready to be executed next

• We have added a call to sort in our call stack

• By expanding "this" (our IterativeSorter object 's'), we can see the current values of the array and sorted array fields

8. Watch what happens to sortedArray after clicking "step over"

() 🕫 🖻 🎽 🗖
Value
IterativeSorter (id=19)
(id=17)
(id=21)

We can see that sortedArray and array both contain the same list of 4 numbers.

9. Line 35 begins a loop that finds the index of the minimum element in the array, and then in lines 40-42 swaps the smallest number with the minimum unsorted value in the list. Looking at sortedArray, we can predict that the index of the minimum element is "0". Hit step over until you reach line 40. Is min set to "0" as we predicted? Which two numbers will be swapped in lines 40-42?

🕪= Variables 🔀 🔏 Breakpoints	
▼ ⊕ this	IterativeSorter (id=20)
🕨 🔲 array	(id=18)
🕨 🔳 sortedArray	(id=21)
0 i	0
min	0
[5, 67, 12, 20]	

10. Continue stepping over lines until you reach the beginning of the loop at line 35. Try to predict what will happen in this iteration of the loop. Looking at the 3 elements starting at index 1, what is the index of the next minimum element? Which two numbers will be swapped? Step over the lines, checking the value of min at 40 and verifying sortedArray when you reach line 35 again to see if you correctly predicted which two numbers would be swapped.

🗱 Variables 🛛 💊 Breakpoir	nts
🔻 \varTheta this	IterativeSorter
🕨 🔲 array	(id=18)
🕨 🗉 sortedArray	(id=21)
0 i	1
[5, 12, 67, 20]	

11. Continue stepping through the execution until you understand how the algorithm for IterativeSorter works. When you've seen enough, press "step return" to go back to main

00	Debug - SortingB/src/IterativeSorter.java - Eclipse - /Us	ers/EmilyHill/Dropbox/In progre
] 📬 🛛 🖬 🖨] 🤇	🎦 • 🕸 • 🔕 • 隆 🖨 🥔 • 🛛 🕫 🖉 • 🗍 🛠 🖉 • 🖓 •	• 🖘 🗘 • 🗘 •
🏂 Debug 😫	🗍 🖓 😴 📩 📀 🕫 🕅 🔳 💷 📲 🖉	🕬= Variables 🔀 🔍 🗞 Breakpoints
▼ J IterativeSorter	(1) [Java Application]	() args
V 🖓 IterativeSort	er at localhost:57797	▶ @ array
🔻 🛷 Thread (r	main] (Suspended)	▼ @ s
📃 lterati	veSorter.main(String[]) line: 63	🕨 🔳 array
📙 /System/Lib	rary/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (F	🕨 🗖 sortedArray
		[5, 12, 20, 67]
🚺 RecursiveSorter.ja	va 🚺 IterativeSorter.java 🛛 🚺 Rectangle.java 🗍 Rec	tangleTest.java ³⁴
54 for	(int i : a)	
55	s += i + " ";	
56 ret	urn s;	
57 }		
58		
59 public	static void main(String[] args) {	
60 // int	Automate running, but not testing	
62 Tte	LJ urray = 15, 07, 12, 20;; rativeSorter s = new IterativeSorter(array);	
⇒63 S.S	ort():	
64 Sys	tem.out.println(s); // uses Sorter.toString	
65 }		
CC		

where we see that the sortedArray field in IterativeSorter variable 's' does indeed contain a sorted array of numbers

12. To complete your debugging session, click "resume" to execute line 64 and finish execution, or click "stop" to halt the execution.

13. To get back into editing mode, select the "Java" perspective in the top right

🔴 🔿 🖉 Debug – SortingB/src/IterativeSorter.java – Eclipse – /Users/EmilyHill/Dropbox/In progress/CMPT 184/workspace				
] █+ 🛛 🗟 🍐 🂁 🏇• Ø• •] 🤌 🖨 🖋•] 🍄 🥖 🐲 🗐 🕤] ᢓ• Ş	😭 🏇 Debug 🐉 Java 🔌			
🎋 Debug 🕱 🛛 🔌 🕪 🗉 🔳 💐 🍡 🗫 🕫 💆 🗊 🌄 🗖	🗱 🕫 Breakpoints	🖢 🏘 🖻 🗸 🕇 🗖		
V IlterativeSorter (1) [Java Application]	() args	String[0] (id=15)		
TerativeSorter at localhost:57674	🕨 🖲 array	(id=17)		
Thread [main] (Suspended (breakpoint at line 63 in IterativeSorter))	▶ @ s	IterativeSorter (id=19)		
IterativeSorter.main(String[]) line: 63				
📕 /System/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home/bin/java (F				
	[5, 67, 12, 20]			
📝 RecursiveSorter.java 🚺 IterativeSorter.java 🕱 🚺 Rectangle.java	tangleTest.java 🎽 🗖 🗄	🗄 Outline 🕱 📃 🗖		
<pre>>1 52 private String prettyPrint(int[] a) { 53 String s = ""; 54 for (int i : a) 55 s += i + " "; </pre>		<pre></pre>		