

CR Task 8.2 Debug This — Report

Name: Paul Watts**ID:** 569987

Issue:	[summarise the problem in your own words]
Program transcript or summary of its behaviour: (format user input in bold)	1 error found: File: /Users/paulwatts/Kit101 Java/8.2CR Debug This/GuessingGame.java [line: 112] Error: cannot find symbol symbol: class play location: class GuessingGame
Defective line number(s):	112
Summary of the cause:	The compiler cannot resolve the symbol 'play' because a GuessingGame object has not been correctly instantiated with the correct syntax
How cause was identified:	Have encountered same problem before in my java experimentation in this course
Replacement code:	<code>GuessingGame game = new GuessingGame(); game.play();</code>

Issue:	[summarise the problem in your own words]
Program transcript or summary of its behaviour: (format user input in bold)	Menu choices for "New Game" do not start a new game. Any number choice displays the menu again. A non number will cause an exception error. (I'm presuming we don't need to fix this exception error for this exercise).
Defective line number(s):	27
Summary of the cause:	The 'while' statement had the conditional tests for 'option' reversed so a valid choice would continue to loop the menu instead of exiting the menu and returning.
How cause was identified:	Tracing the value of 'option' in my head and evaluating the 'while' statement in my head.
Replacement code:	<code>while (option < 1 option > 3);</code>

Issue:	[summarise the problem in your own words]
---------------	---

Program transcript or summary of its behaviour: (format user input in bold)	Entered -100 to “give up”. Program returned me to ‘play again’ however it showed my efficiency as 100.0% when it should be 0.0% plus not good to tell me “congratulations” when I’m a quitter lol.
Defective line number(s):	77 Needs ‘break;’ inserted before this line
Summary of the cause:	Missing ‘break;’ for ‘CASE RAGE_QUIT:’ causing execution to drop into ‘case CORRECT:’
How cause was identified:	Setting debug breakpoint at line 43 and line stepping through the code.
Replacement code:	<pre>case RAGE_QUIT: System.out.println("Bad luck. You gave up after " + attempts + " attempts"); System.out.println("The secret number was " + secret); break;</pre>

Issue:	[summarise the problem in your own words]
Program transcript or summary of its behaviour: (format user input in bold)	Selected ‘Easy’ and did a guess of 2 and program displayed ‘Too low!’ however it showed the number of attempts as 25 instead of 1
Defective line number(s):	50
Summary of the cause:	Parameters for calling ‘showFeedback’ method in the wrong order
How cause was identified:	Searched for ‘Attempts so far:’ in code and zeroed in on “Too Low” section. This one did stump me for a few minutes after tracing the code and about to use the debugger since ‘attempts’ had not been modified before calling the ‘showFeedback’ method and then realised the only logical explanation was that parameters were in the wrong order and didn’t throw a compiler error because they were the same ‘int’ data type
Replacement code:	<code>showFeedback(result, attempts, secret, max);</code>

Issue:	[summarise the problem in your own words]
Program transcript or summary of its behaviour: (format user input in bold)	<p>New Game</p> <ol style="list-style-type: none"> 1. Easy 2. Medium 3. Hard <p>Select difficulty: 3</p> <pre>java.lang.ArrayIndexOutOfBoundsException: 3 at GuessingGame.selectDifficulty(GuessingGame.java:32) at GuessingGame.oneGame(GuessingGame.java:41) at GuessingGame.play(GuessingGame.java:108) at GuessingGame.main(GuessingGame.java:116) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)</pre>

	<pre> at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccess orImpl.java:62) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMeth odAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:498) at edu.rice.cs.drjava.model.compiler.JavacCompiler.runCommand(Jav acCompiler.java:259)</pre>
Defective line number(s):	32
Summary of the cause:	The DIFFICULTIES array has 3 elements which are at positions 0,1,2 therefore trying to access position 3 caused the error as it's out of bounds for the array.
How cause was identified:	Exception error shows the line number error occurred at. Was not difficult to resolve.
Replacement code:	<pre>return DIFFICULTIES[(option-1)] ;</pre>

Note: Tested multiple iterations of the program at each difficulty level once it was debugged. Used the debugger to determine the secret random number for each difficulty level and then tested values to show the logic was working correctly for a reasonable number of test cases.