***Debugger Crash Course for Beginners***

**Break Points**

* We can set/toggle/disable break points by right clicking on the left margin of our code where the line numbers are
* When you set a break point, the debugger will stop the program there and wait for instructions from you before moving forward

**The Debugger Perspective**

* The Debug perspective opens automatically but it can also be done by selecting the Window -> Perspective -> Open perspective menu at the top of Eclipse (along with many other perspectives)
* We can toggle between the debug and Java EE perspectives in the top right corner
* Run the debugger by selecting the bug icon in the top bar of Eclipse to the left of the Run button.
* Note that the line highlighted in green is not executed yet where the breakpoint was set is not executed yet
* The top left pain is where we will control the debugger by default.
* If you don’t see the buttons along the top this top left pane in line with the tab that says debug click the downward facing arrow near the top right of this pane and select “Show debug toolbar”
* In the top right pane we can see all of our variables and information about their current state
* The three tabs in the top right pane we will worry about are Variables tab which displays information about the current state of the variables of our program, Breakpoints which contains a list our breakpoints, and Expressions which contains all the states of the expressions you’re currently watching. Note that if you are watching an expression and you are currently not in the class where it is defined you will see an error in the expression tab where its value normally is displayed.

**Debugger Functions: Watch, Step Into, Step Over, Step Return**

* If we select step into from the debug toolbar we will be taken inside the method/constructor being called
* If we right click some code we can watch the value of that expression
* If we are confident that something will work we can step over the constructor which won’t take us inside the method but to the state of the program after it’s execution
* If you not interested in watching the constructor/method do it’s work we have the step return option that will takes out of the current block skipping forward to the place that originally called it