German University in Cairo
Media Engineering and Technology
Prof. Dr. Slim Abdennadher

Computer Programming Lab,
Lab Assignment 0

This document aims at helping you get used to the submission procedure on our submission and grading system *Evaluator*. It also teaches you how to debug your own code, which will be helpful to you throughout the course.

1 Setup

First, you need to copy the portable eclipse from the intranet to your PC. You will find it in path:

```
V:\\Faculties\Media Engineering Technology\\
[CSEN401-Spring2011]Computer Programming Lab\\Eclipse JavaFX
```

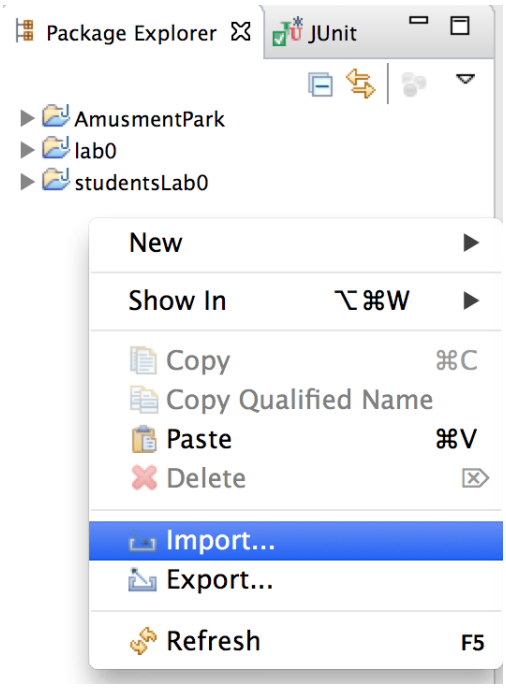
Then, download `BuggyLab0.zip` from the `met.guc.edu.eg` website.

2 Importing and Exporting in Eclipse

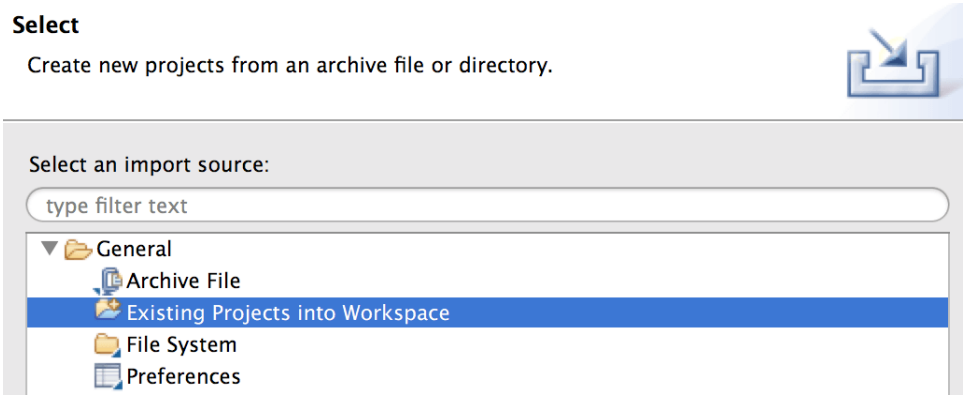
2.1 Importing a Project

a) Open Eclipse

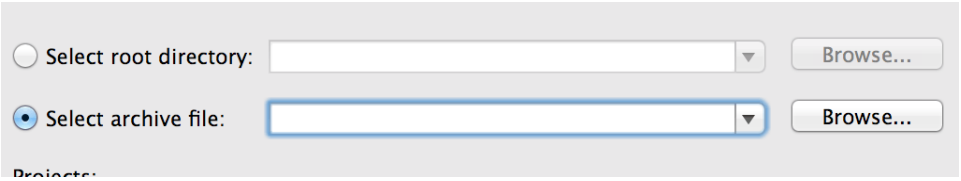
b) Right click in package panel and select import



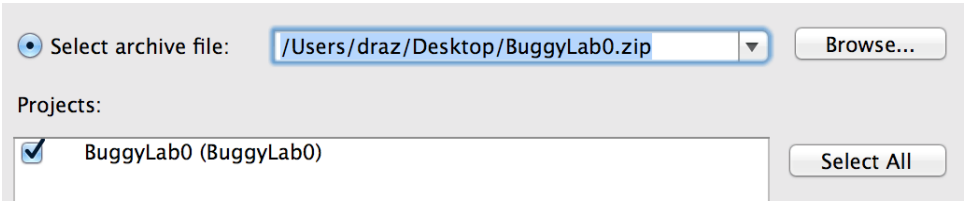
c) Select import existing java project



d) Select archive file



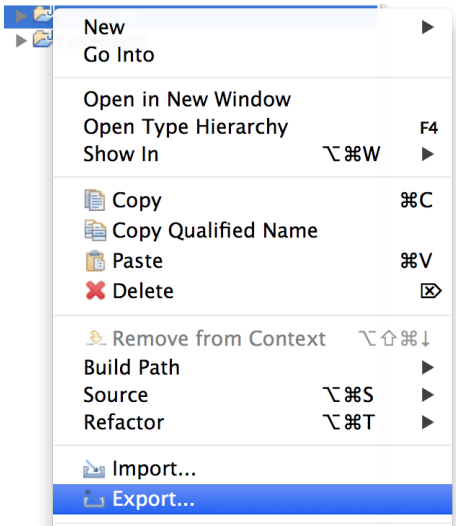
e) Select `BuggyLab0.zip` from the location you save it in



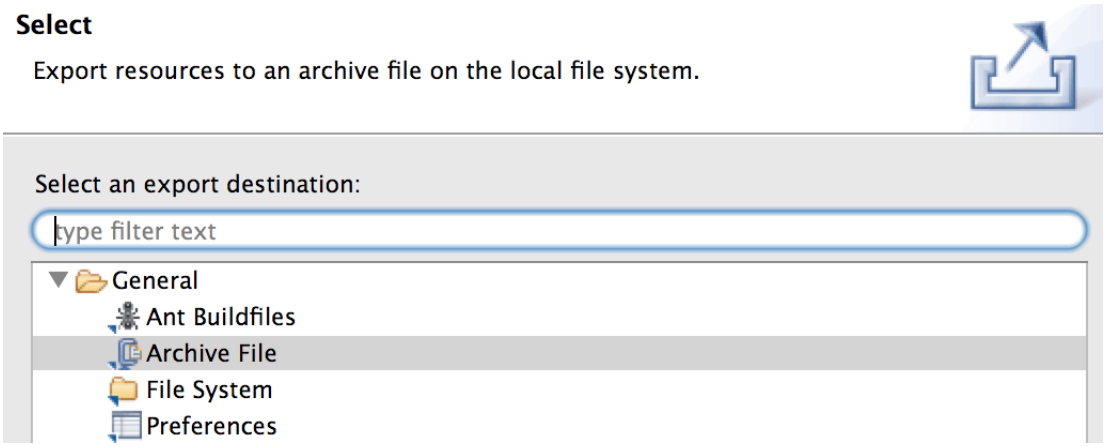
f) Finish

2.2 Exporting a Project

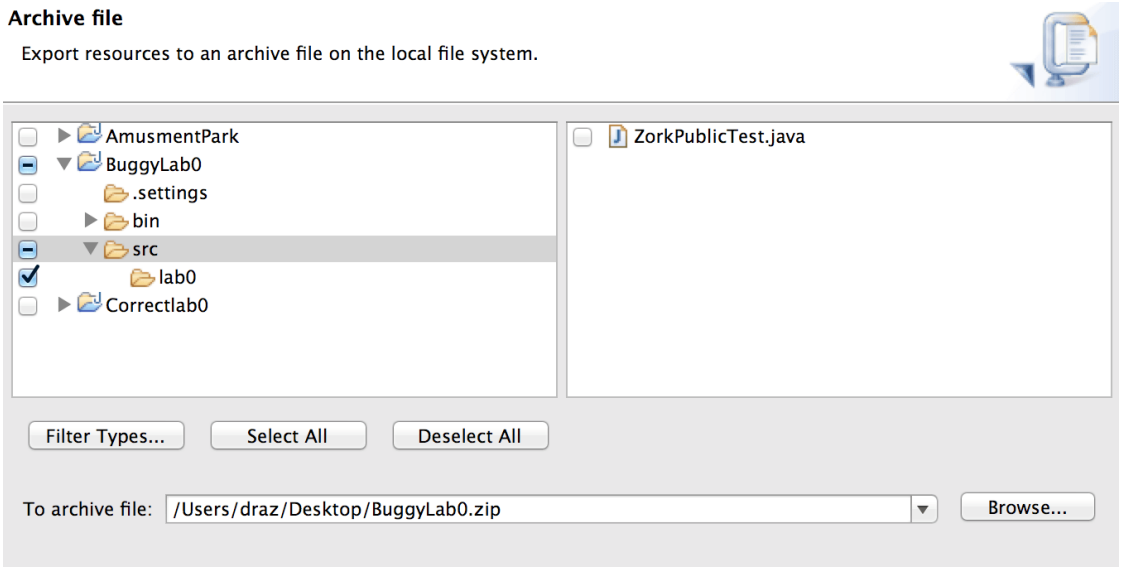
a) Right click on project `src` folder and select Export



b) Select as Archive file



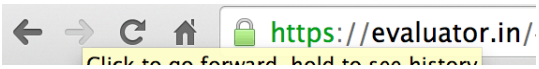
c) Make sure you uncheck everything except `src`. Also, uncheck the test files in `src`



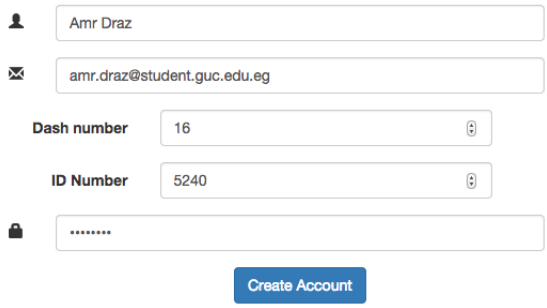
d) Name the zip file the same as the project name `BuggyLab0.zip` (any other naming convention would lead to incorrect test results)

3 Code Submission

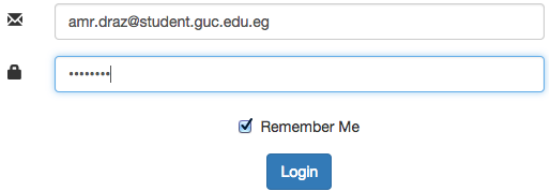
a) Go to `evaluator.in` using *Google Chrome* (any other browser will not work)



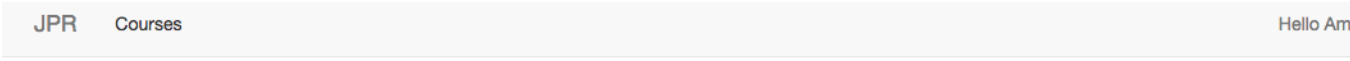
b) Register using your GUC student email (make sure to use '@student.guc.edu.eg')



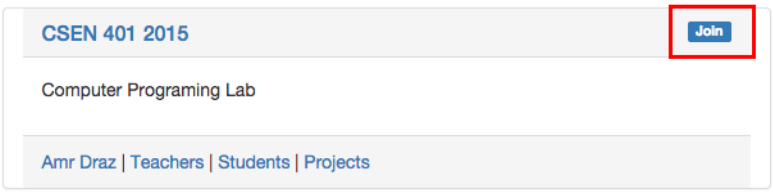
c) Login with your GUC student email and password



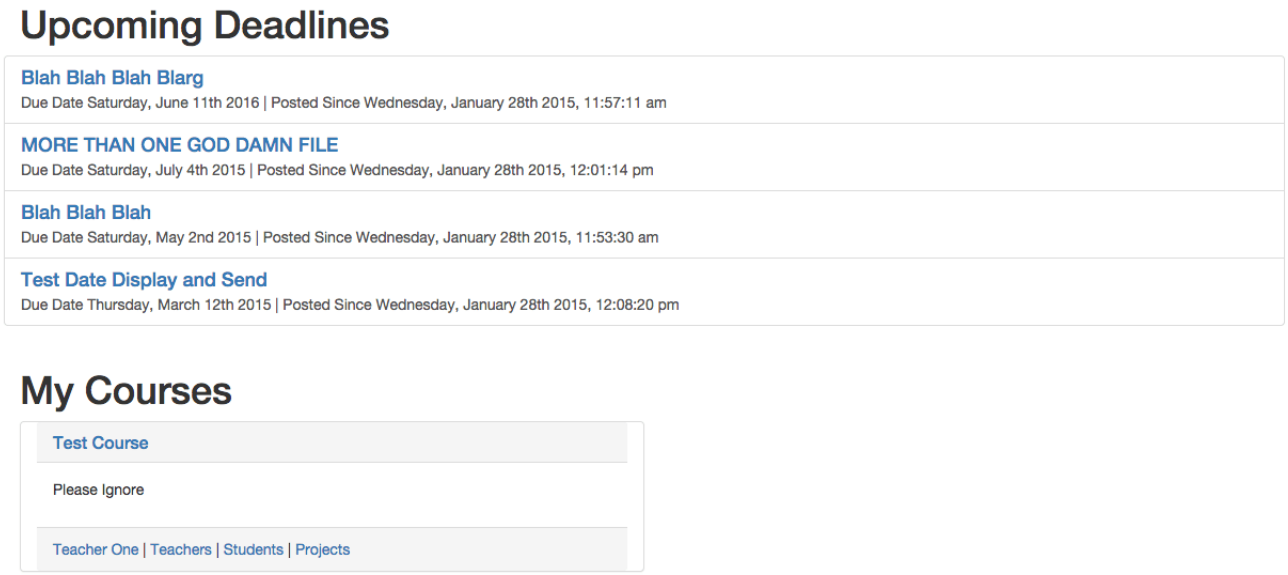
d) Navigate to courses



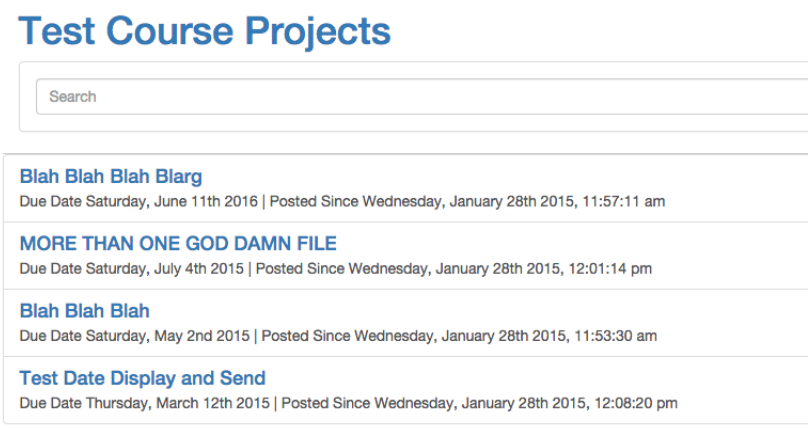
e) Join the CSEN 401 course



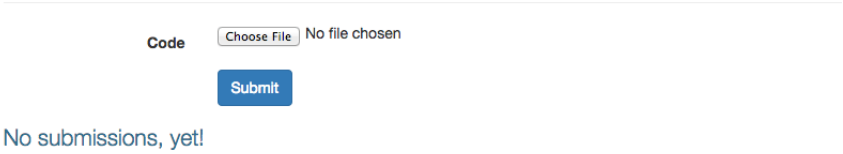
f) Go to the course projects



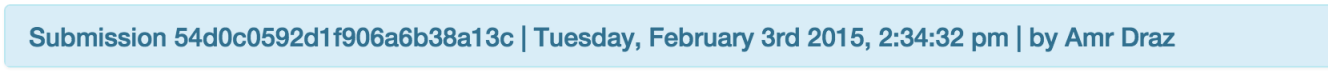
g) Select correct project



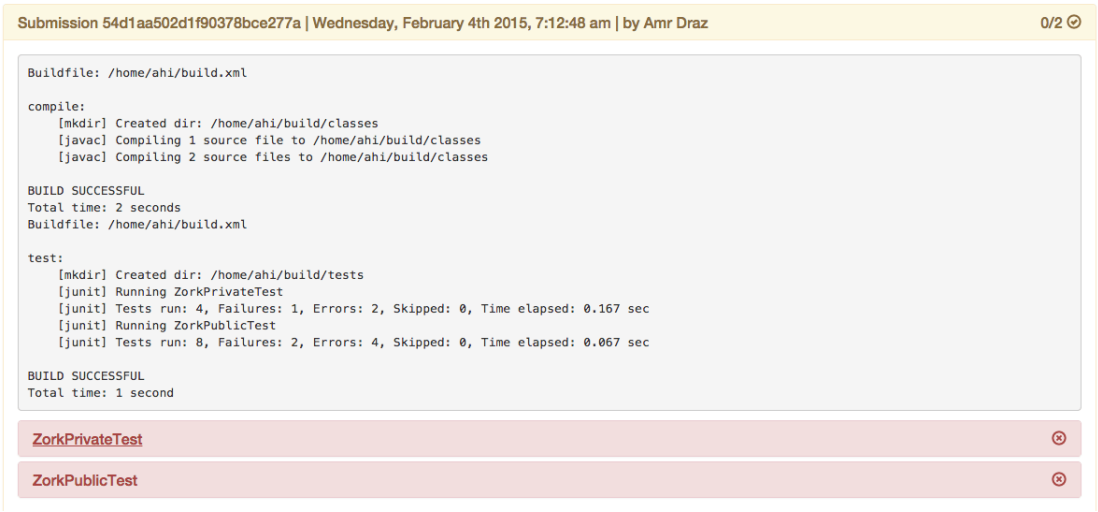
h) Submit `BuggyLab0.zip`



i) You should see a blue submission indicating submission is in progress. Refresh the browser until the blue progress bar is gone. Your submission will be highlighted in one of the following colors:



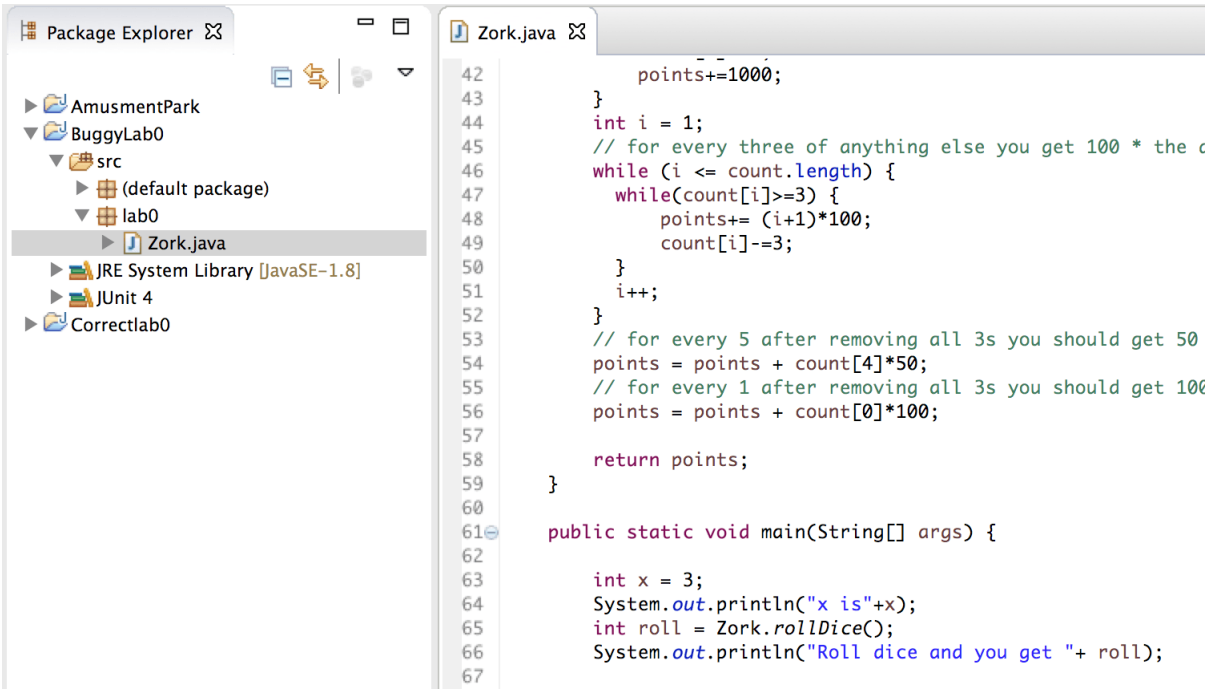
1. Red means that your project has a compile error
2. Green means that your submission is correct
3. Yellow means that some tests failed and you have some errors in your code
   - In this case, you can review the details of your submission by expanding it and looking at the tests that failed



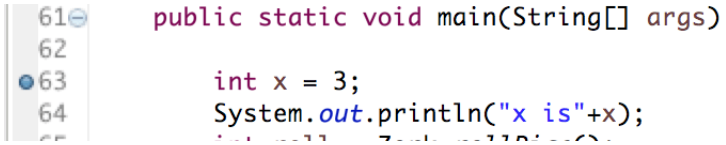
4 Code Debugging

After finding out that your code has some errors, you need to try and fix them. Instead of doing so through printing as you are used to from previous courses, you will learn how to make use of Java's debugging environment. The debug feature can be used to trace the exact execution of your algorithms.

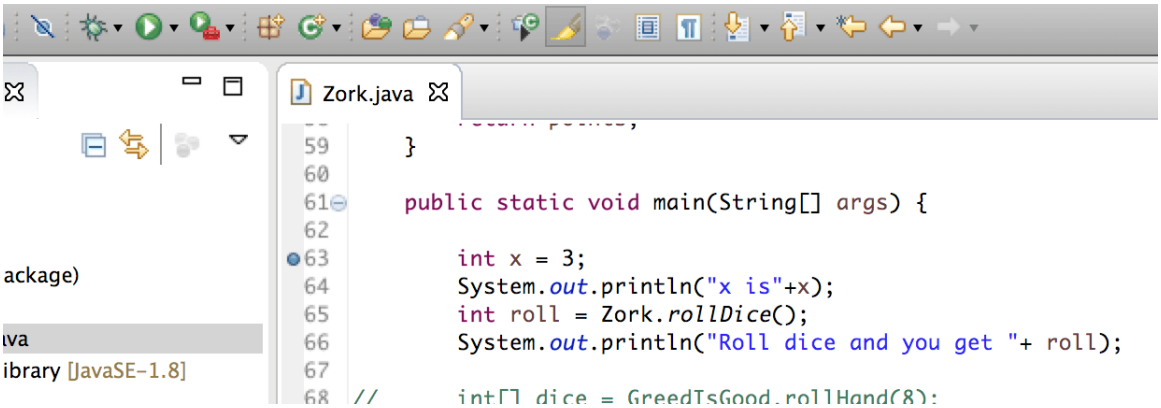
a) Go back to Eclipse and open `Zork.java`. Go to the main method



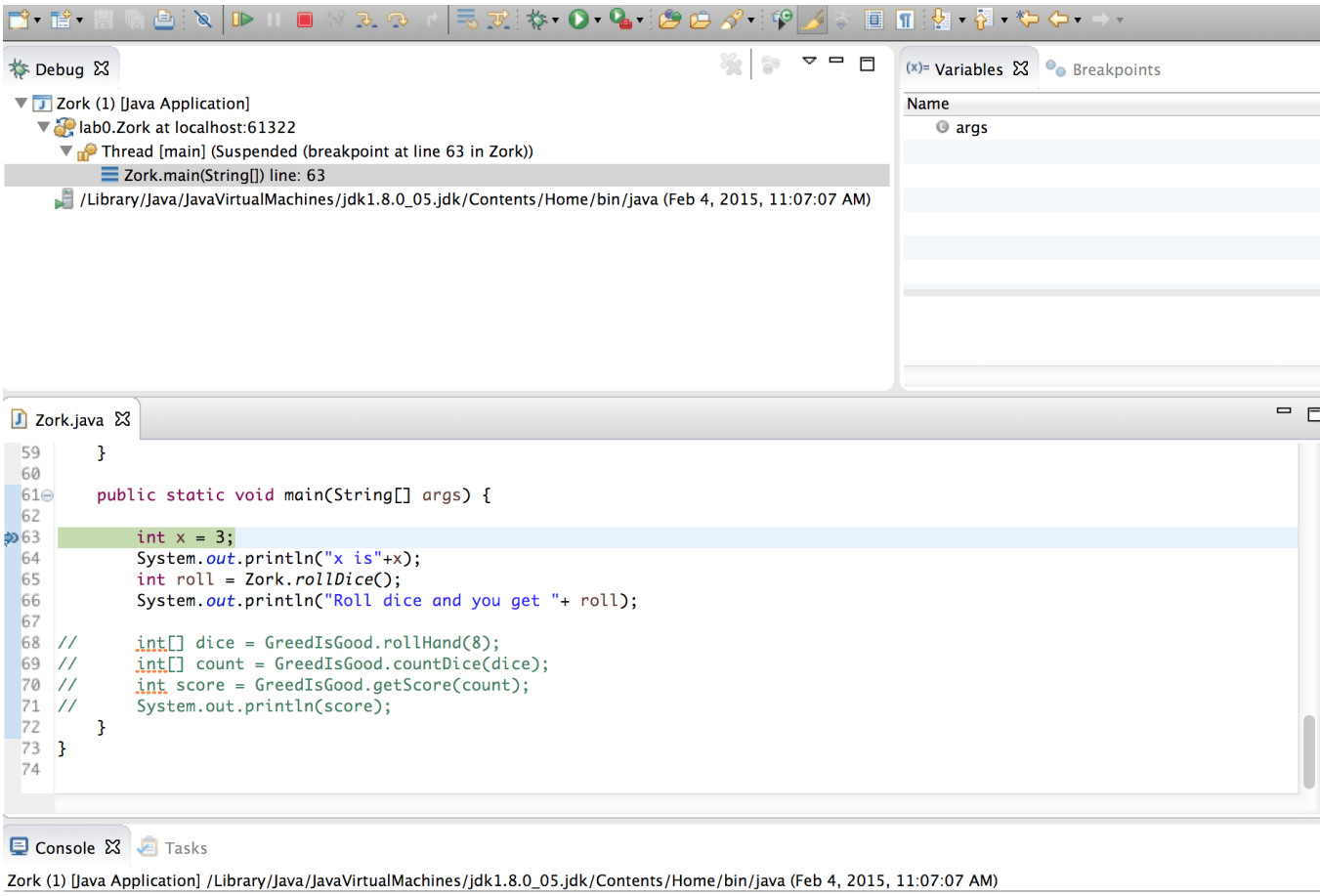
b) Add a breakpoint in next to int x by double clicking on the gutter next to the line number



c) Click on the bug icon to start debugging



d) You should see your workspace change do not panic these new windows are here to help

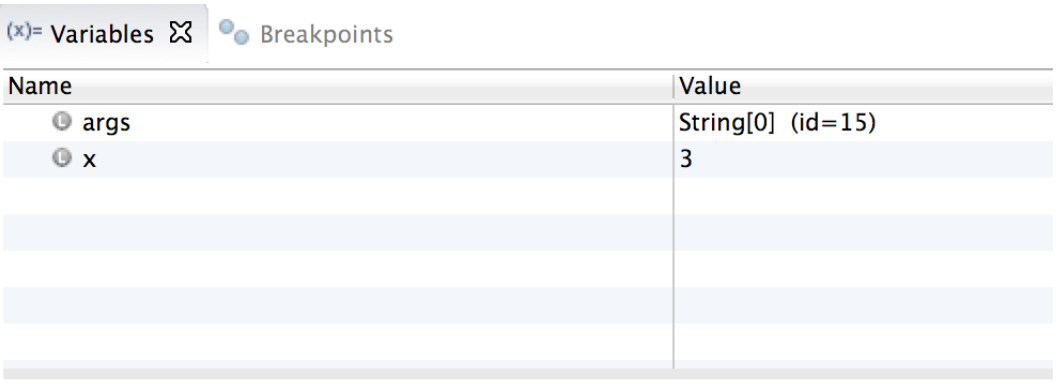


e) The controls at the top allow you to control the flow of your code

f) Step over using the left most control and you will notice the line changed from 57 to 58 (highlighted in green)

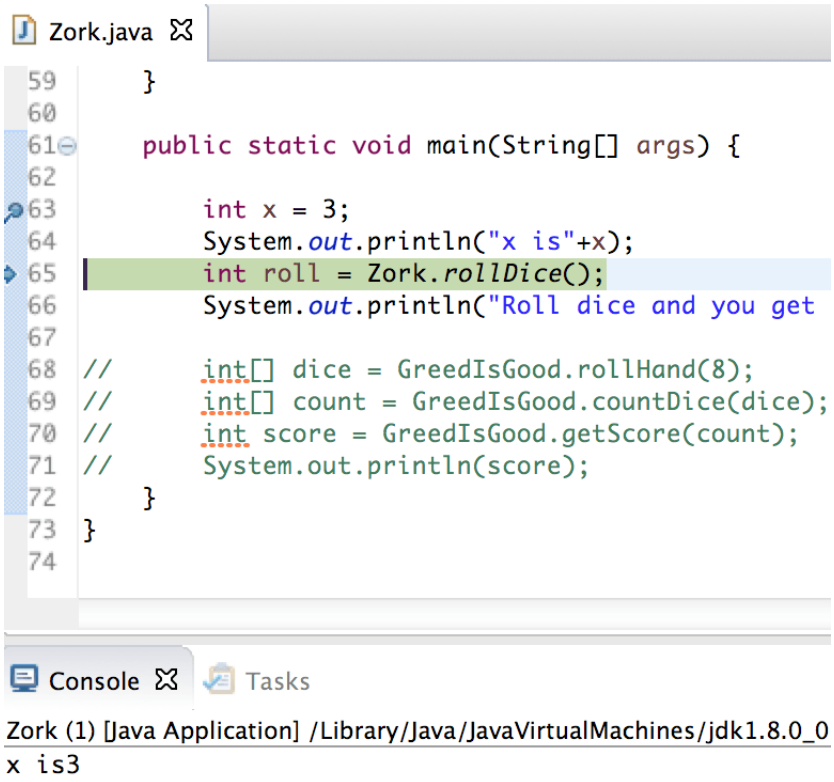
```
54
55    public static void main(String[] args) {
56
57        int x = 3;
58        System.out.println("x is"+x);
59
60        System.out.println("Roll dice and you get "+ Zork.rollDice());
61
```

g) The variables panel indicates the value of all variables of all the running methods. If you look at the variables panel, you will notice that x is set to 3 (you can hover over x to see it's value as well)



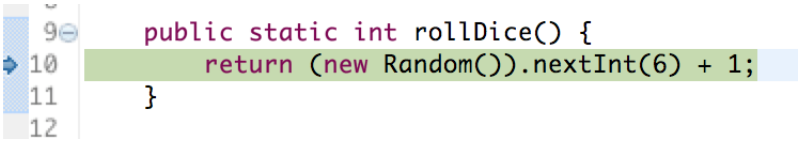
h) Step over again

i) You can see the output of the `System.out.println` in the console panel

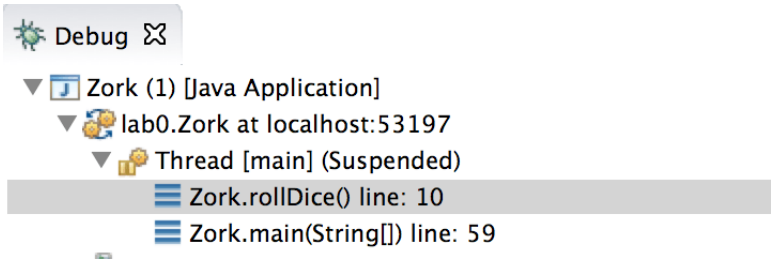


j) We can go inside methods using the step into control (the one before the step over control)

k) Note that we are now inside the `rollDice` method



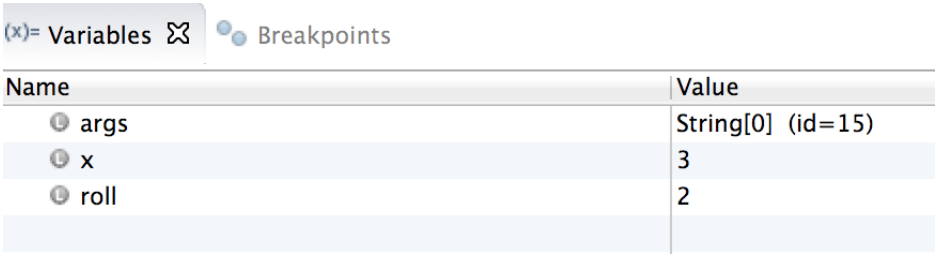
l) You will also notice that the stack frame changed. You can select the previous frame which will show you where you came from and use that to see the variables before going into the method.



m) Since there is not much to do within this method, let us step out. This will exit the method but continue running the code until the method returns

```
56
57          int x = 3;
58          System.out.println("x is"+x);
59          int roll = Zork.rollDice();
60          System.out.println("Roll dice and you get "+ roll);
61
```

n) If you step over again you will see in the variables panel that the variable `roll` was set by the method.



o) Step over again to see the output in the console

p) You can exit the debugging environment by choosing the Java logo

q) To practice more start trying to solve the first bug you found when you submitted your code online. To do so use line 69 as your input, insert a breakpoint and start debugging. When you solve the bug, submit your code online to make sure that you correctly resolver the issue.

r) After having solved your first bug, you can try uncommenting the final lines, stepping into the methods and tracing them to understand what they do. After that you can start debugging the rest of your code and submitting it to the system until you pass all the tests.