



**Department of Electrical and Computer Engineering
University of Puerto Rico
Mayagüez Campus**

ICOM 4035 – Data Structures (CS2) Spring 2005

Laboratory 1: Using the Eclipse IDE and Debugger

1. Objectives

1. Practice the use of the Eclipse IDE and debugger with Java programs.
2. Learn about facilities such breakpoints, step over, step in and display variables.
3. Learn how to track the control flow of a program with the debugger.

2. Overview

In this laboratory exercise you will use the Point class previously discussed in the course to learn how to use a program called Eclipse. Eclipse is an Integrated Graphically Environment (IDE) used to develop Java Application. Eclipse supports editing, compiling, running and debugging Java programs. Our major concern is using the Eclipse Debugging Facilities. With a debugger, can better debug your code by using facilities that let you:

- Set breakpoints at which you a stop the control flow of you program.
- Set display variables to observe how a variable changes as program execution moves forward.
- Run the program step by step, and being able to print values of variables.
- Run the program step by step, and enter into the functions being called by the program.

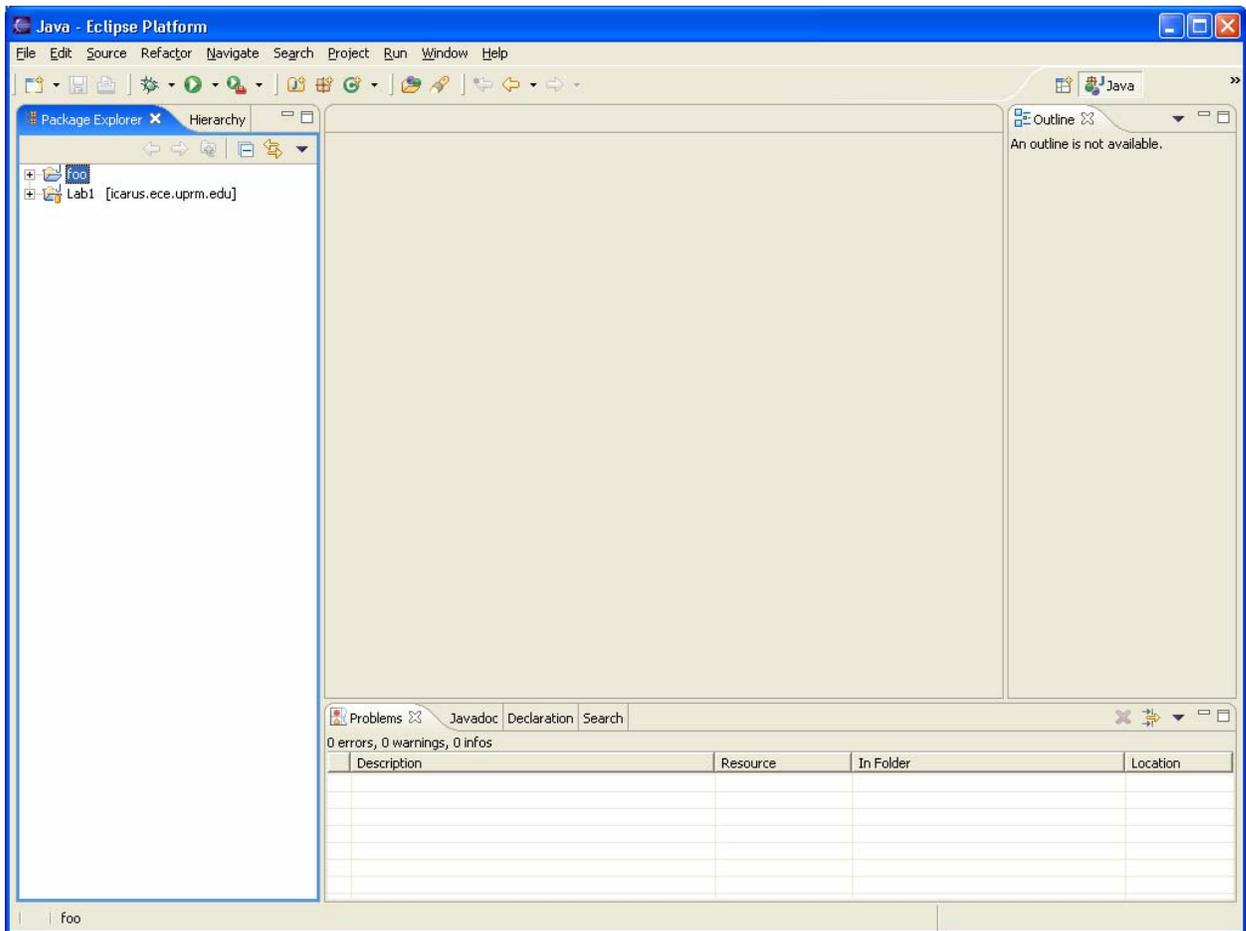
In this laboratory will learn how to perform these actions, and it will serve to help you debug the projects and remaining laboratory exercises.

3. Compiling the Point class and Point Tester Application

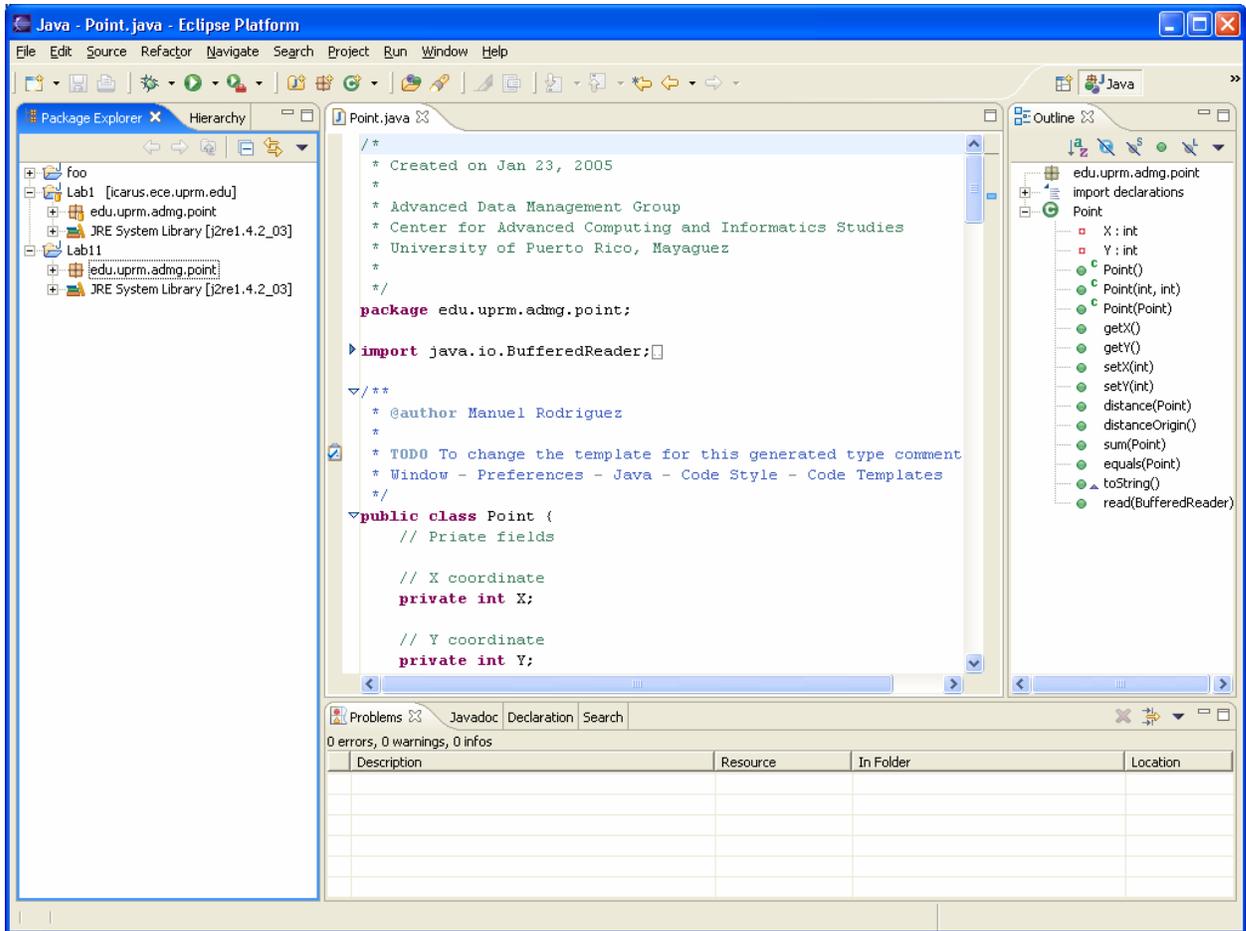
The first task to do is to login to your account at the Amadeus lab, run a Web browser, and download the source code for laboratory number 1. This file is called lab1.tar. Untar the file, and examine the files Point.java and PointTester.java. These files contain the implementation of the Point ADT that we discussed in the first lecture of the class. In addition, you will find the file PointTester.java, which has a main program to the test the Point class.

Running Eclipse

1. Run the eclipse IDE by typing the command eclipse at the command prompt.
2. Press OK to all dialog boxes that appear.
3. You will need to go to the eclipse Workbench.
4. This will take you to a screen like this one.



5. Create a new project by pressing the Menu ->File->New->Project->JavaProject and Press Next.
6. Type Lab1 as the project and press finish.
7. You will see the Lab1 on the Package Explorer Tab. This is directory tree where you code will be organized.
8. Now, create a new package call edu.uprm.admg.point. To do so, press the following Menu->New->Package and type edu.uprm.admg.point as the package name and press finish. Notice how the package name appear under the Lab1 tree at the Package Explorer tab.
9. Create the class Point.java, by pressing Menu->File->New->Class. Type Point as the name of the class and press Finish. You will see a window like this one:



Notice the tabs:

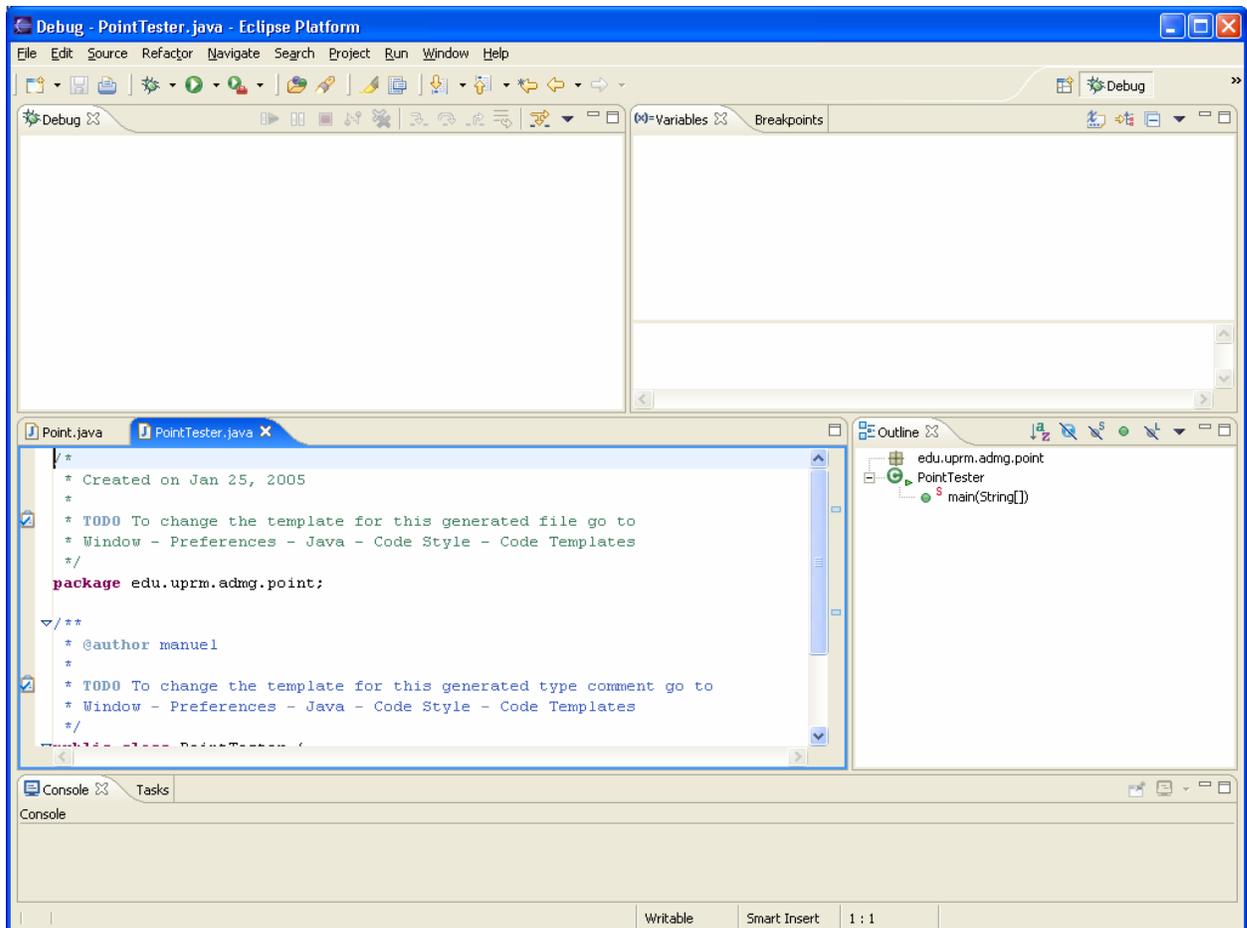
- **Package Explorer** – Shows the structure of your project, the packages available, and the Java classes available.
- **Source Code** – Shows the source code of your class.
- **Outline** – Shows the hierarchy of classes within the packages.
- **Console Tab** – Shows compilation problem, output of your program, etc.

In eclipse, the collection of tabs available you to perform a given is called a perspective. There are perspectives for writing code, debugging, making Web Pages, etc. Right now you are working on the Java Perspective.

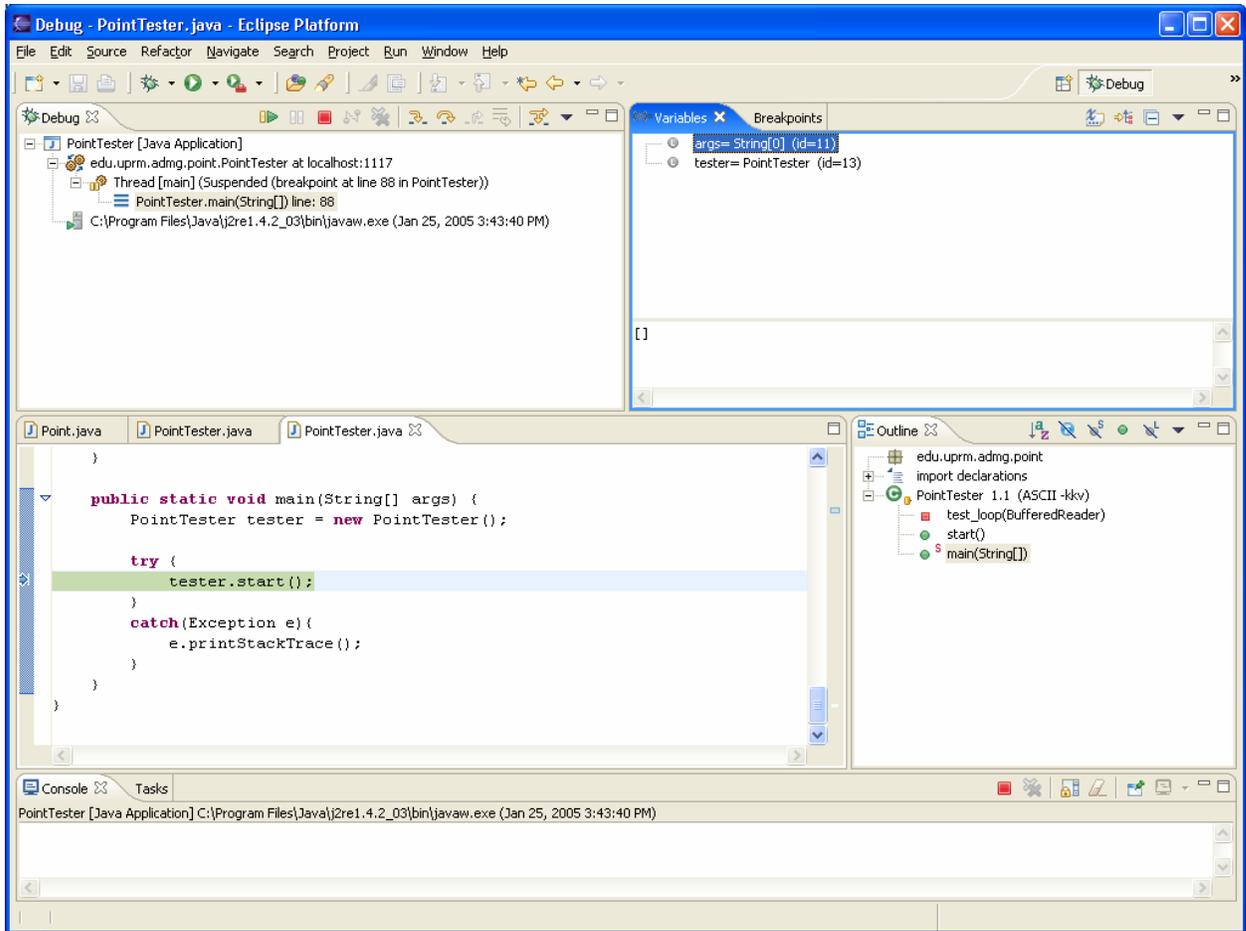
10. Take the code from the file Point.java you downloaded, and copy all its contents to replace the code that eclipse generated automatically. Press save to save the application. Notice that eclipse automatically compiles the program when you save it.
11. Repeat steps 9 and 10 to incorporate the class PointTester
12. At this point you will no error message on the Problems that. When errors do occur, you will see a red flag next to the line with the error and an error explanation on the Problems tab.
13. Now you program is ready for running and debugging. Next will debug the application.

4. Debugging the PointTester Program with Eclipse

1. First switch to the Debug Perspective by pressing Window->OpenPerspective->Debug.
You will see a screen like this:



2. Observe the various tabs in the Debug perspective. There is one area containing the Java code. The other areas are used to show program Input/output, trace variables and values, and see the current function being executed.
3. Let us set a break point in the PointTester Program. A break point tells the debugger to stop executing at some point in the program. Then you can run each instruction one-by-one to observe what is happening in your program. To set a breakpoint on the instruction tested.start() in the main() function, highlight the line and double click on the blue bar at the left of the window. You should a blue dot indicating that a break has been set.
4. Now run the application by pressing the button with the Bug icon and pressing PointTester. The program will start running and then stop at this line with the break point. You will see a little arrow and a green mark indicating the instruction to be executed next, as shown in the following screen.



(1) (2) (3) (4) (5) (6)



5. Now let us look at the button available to use
 - 1) Starts the execution of the debugging process.
 - 2) Pause the execution of the debugging process.
 - 3) Stop the debugging process.
 - 4) Step into the execution of a function, entering the body of the function.
 - 5) Step over the execution of a function , skipping how the function operates.
 - 6) Step return to go back to the body of the previous function.
6. To enter into the body of function of function start() press (4). **What happens?** Press (6). Type 1 2 at the input prompt, and then type 7 at the next input prompt. **What happens?**
7. Now run the PointTester program again from within the debugger. To enter into the body of function of function start() press (4). **What happens?** Observe how the values of the variable are shown on the Variable Tab (upper right-hand corner). Keep pressing (5), and see how the output of the program appears. When asked for input, first enter 12 2, and later on type 7. Continue pressing (5) until the program finishes.

8. Now run the PointTester program again from within the debugger. To enter into the body of function of function start() press (4). **What happens?** Observe how the values of the variable are shown on the Variable Tab (upper right-hand corner). Keep pressing (4), and see how the output of the program appears. When asked for input, first enter 12 2, and later on type 7. Continue pressing (4) until the program finishes.