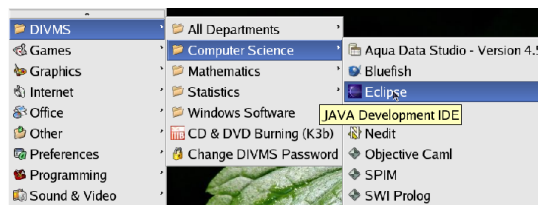


Introduction to Eclipse

Start Eclipse

- Click  and then click **Eclipse** from the menu:

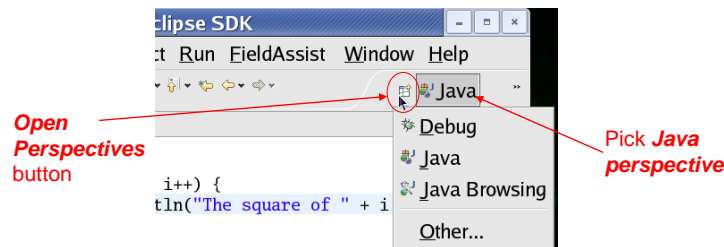


- Or open a shell and type **eclipse** after the prompt.

Initialize Eclipse

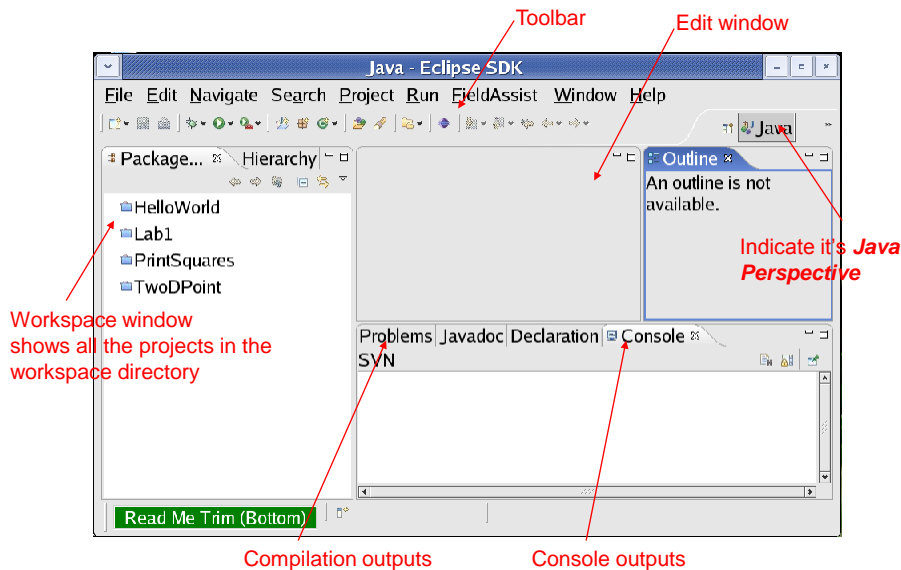
- Choose a workspace (a directory used by Eclipse to store your programs)
 - When you first start Eclipse, Eclipse will ask you to specify the workspace to use.
 - Accept the default workspace provided by Eclipse or specify an existing directory as the workspace.

- Choose a perspective (the layout of Eclipse user interface).
 - Open **Java perspective** (an interface for editing java source code): click **Open Perspective** button > click **Java**.



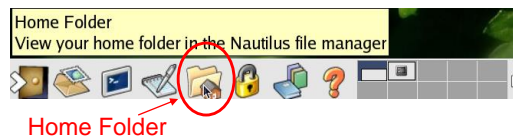
- **Debug Perspective** (an interface for debugging the program).

Overview of Eclipse Java Perspective



Load an Existing Java Program

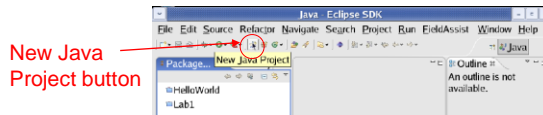
1. Open **Home Folder** and find the Workspace directory you use for Eclipse.



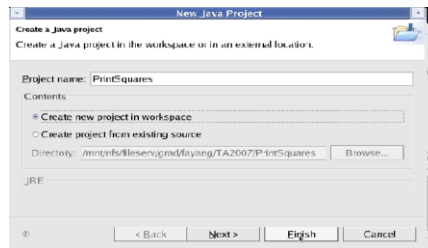
2. Create a folder named **PrintSquares** (or any other name you prefer) under the workspace directory.
3. Download **PrintSquares.java** from [Lab1 Document](#) to **PrintSquares** folder you just created.

4. In Eclipse create a project named **PrintSquares**.

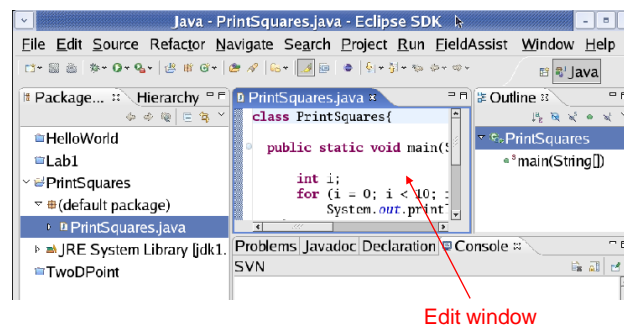
- Click **New Java Project** button.



- Type **PrintSquares** as the project name and then click **Finish** button.

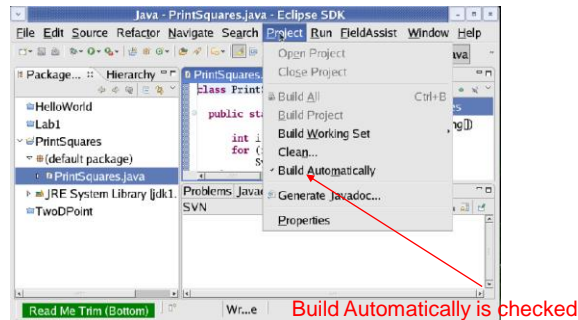


5. In Workspace window double click **PrintSquares**, then **(default package)**, and then **PrintSquares.java**. The source code of **PrintSquares.java** is shown in **Edit** window.



Compile the program

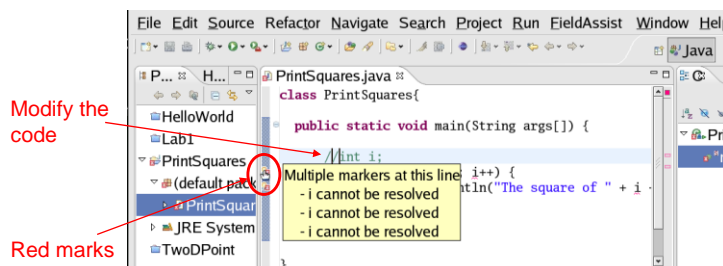
- If **Build Automatically** is checked, the program will be automatically compiled whenever you save the program.



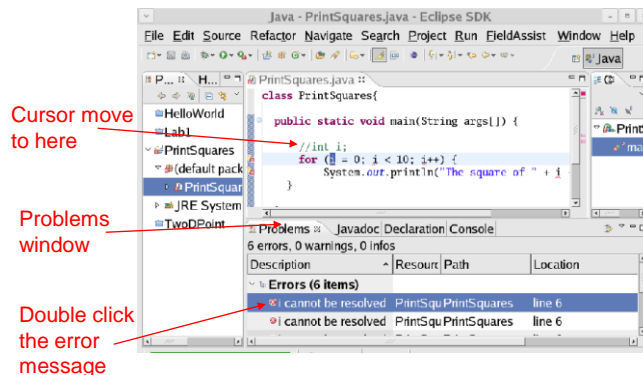
1. Modify "PrintSquares.java" source code as follows:

`int i;` Change to `//int i;`

2. The red marks on the left side of **Edit** window indicate that there are errors in **PrintSquares.java**. Move the cursor over a red mark to see the error message.



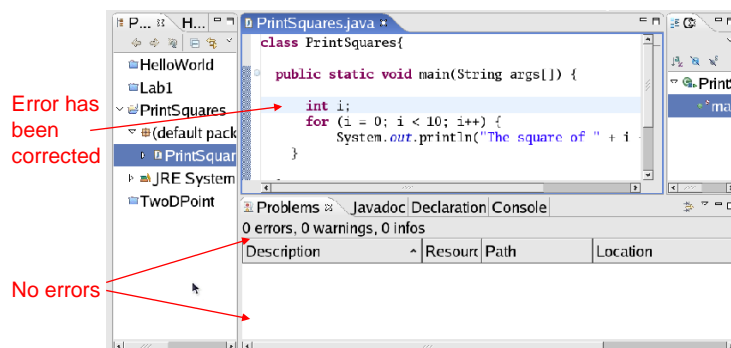
3. Click **Save** button on the **toolbar** to compile the program. **Problems** window shows the errors in the source code. Double click an error message and the cursor in **Edit** window will automatically move to the line in the source code where the error appears.



4. Correct **PrintSquares.java** source code as follows:

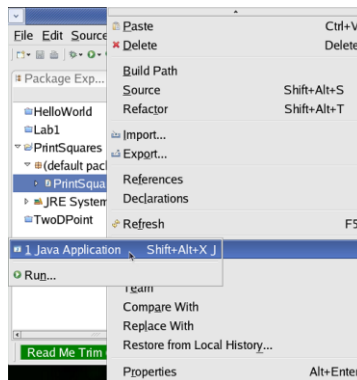
`//int i;` Change back to → `int i;`

5. Click **Save** button to compile the code again.

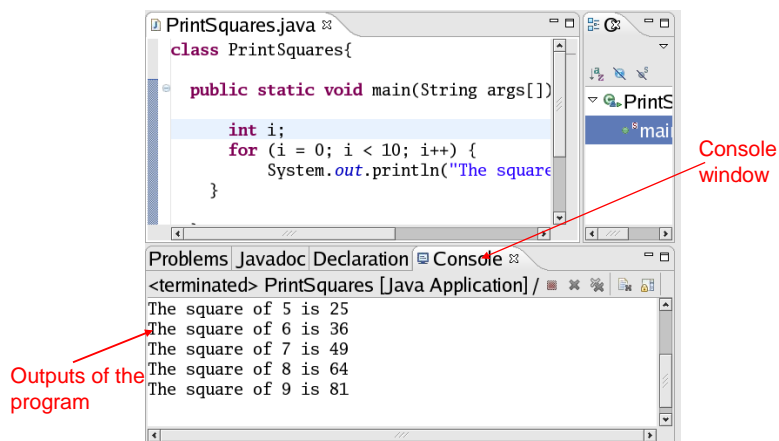


Run the program

1. Right click **PrintSquare.java** in **Workspace** window and select **Run As> Java Application**.

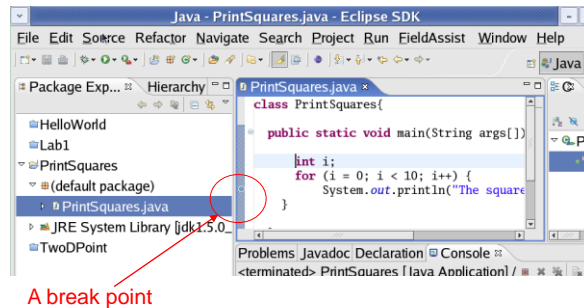


2. **Console** window shows the outputs of the program.

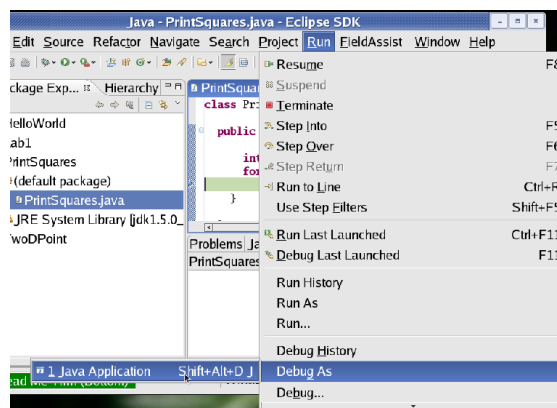


Debug a Program

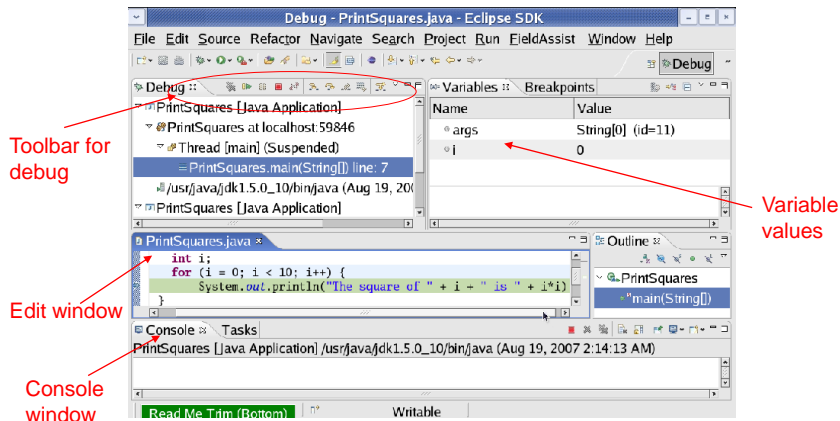
1. Add breakpoints: double-click the gray bar on the left of **Edit** window. A blue dot indicates a breakpoint. To remove a break point, double click the breakpoint.



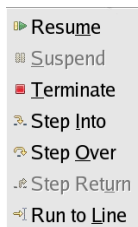
2. Select **Run->Debug as...->Java Application** to start the debugger.



3. Click **Yes** button in **Confirm Perspective Switch** window to switch Eclipse from **Java Perspective** to **Debug Perspective**.



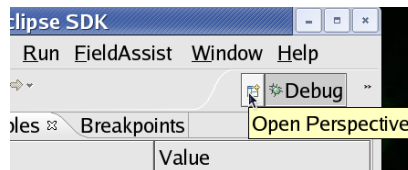
4. Play with the debug commands and watch the change of variable values in **Variable** window and the outputs in **Console** window.



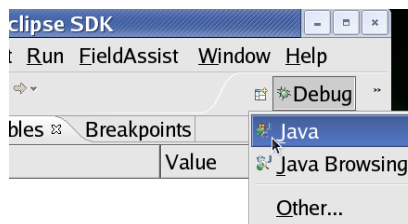
- Resume** resume the execution of a paused program.
- Suspend** temporarily pause the execution of a program.
- Terminate** end the current debug session.
- Step Into** execute a single statement or step into a method.
- Step Over** execute a single statement. If the statement contains a call to a method, the entire method is executed without stepping into the method.
- Step Return** execute all the statements in the current method and returns to its caller.
- Run to Line** runs the program, starting from the current execution point, and pauses at a breakpoint.

5. Switch Eclipse from **Debug Perspective** back to **Java Perspective**.

- Click **Open Perspective** button.



- Then click **Java**.

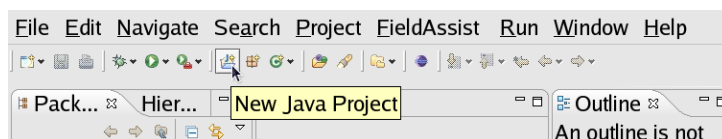


Create A New Java Application

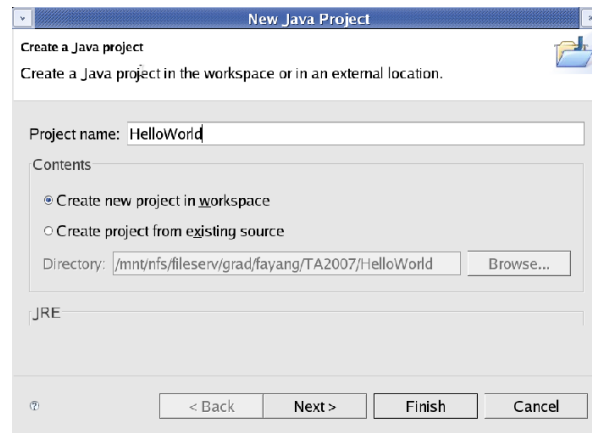
Example: create a HelloWorld java application

1. Create a new project named **HelloWorld**.

- First click **New Java Project** button.

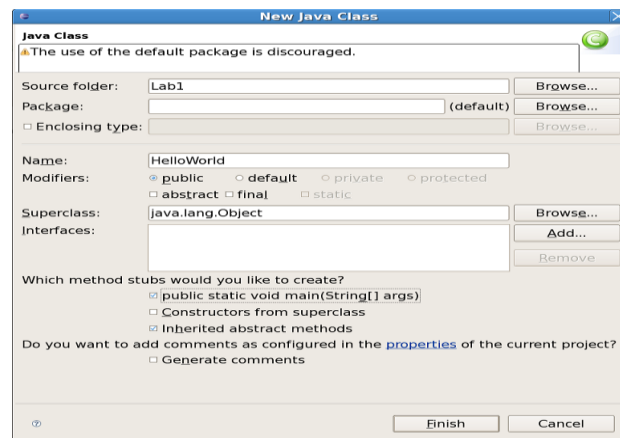


- Then in **New Java Project** window input the project name as **HelloWorld** and click **Finish** button.

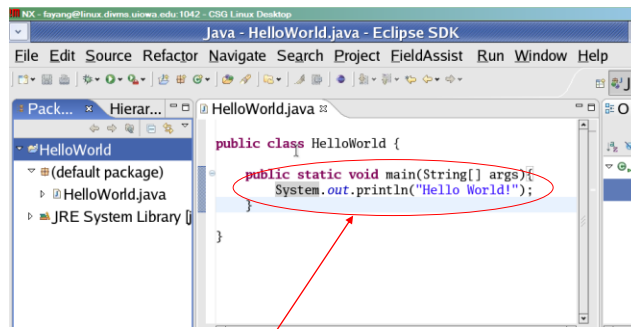


2. Click **New Java Class** button to create a Java class.

- In **New Java Class** window, input **HelloWorld** as the name and check the box "public static void main (String[] args)" if you want a main method.



4. Modify **HelloWorld.java** source code as follows:



Add **System.out.println("Hello World");**
inside **Main** method.

5. Follow the instructions in the previous slides to compile and run the **HelloWorld** program.