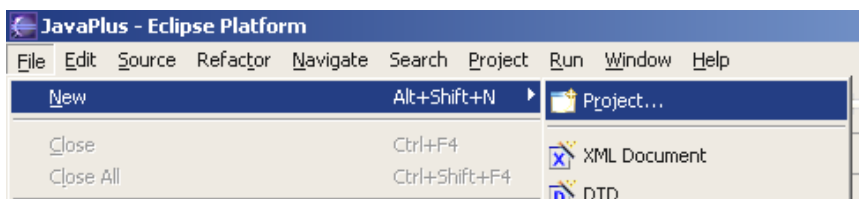


Using Eclipse

- Eclipse organizes code with what is called a *Workbench* or sometimes a *Workspace*.
- Within a workbench, multiple *Projects* are created. Each project is made up of source and build folders.
- This tutorial assumes that you already have a workbench setup and have applied any settings file(s) to that workbench.
- This tutorial goes through the steps to create a new project, to create Java source code files within that project, to edit the source, to compile the source, and to run the resulting bytecode.

Create a Project

- Go into File/New and select Project.



- Select "Java Project" (figure 1) and then press Next .
- For the "Project Name", type `HelloEclipse`, leaving the other settings as-is (figure 2), then press Next.

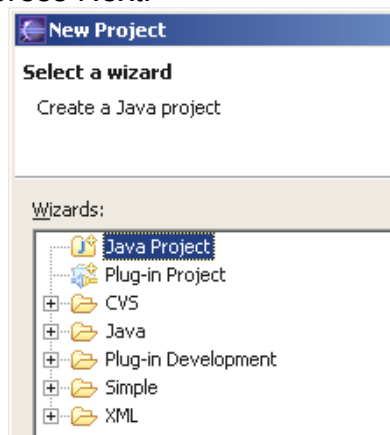


Figure 1

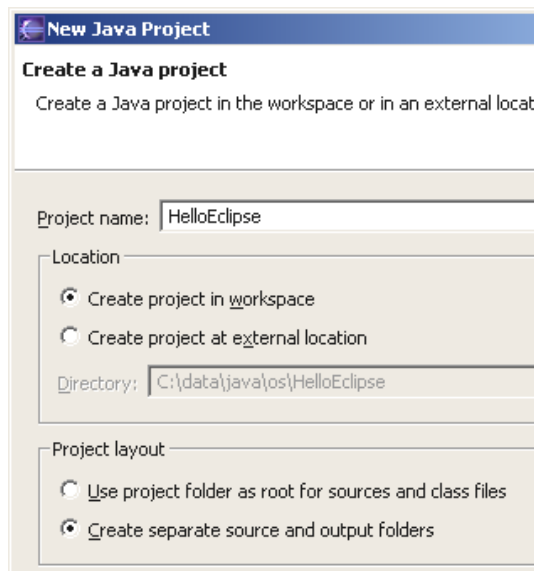
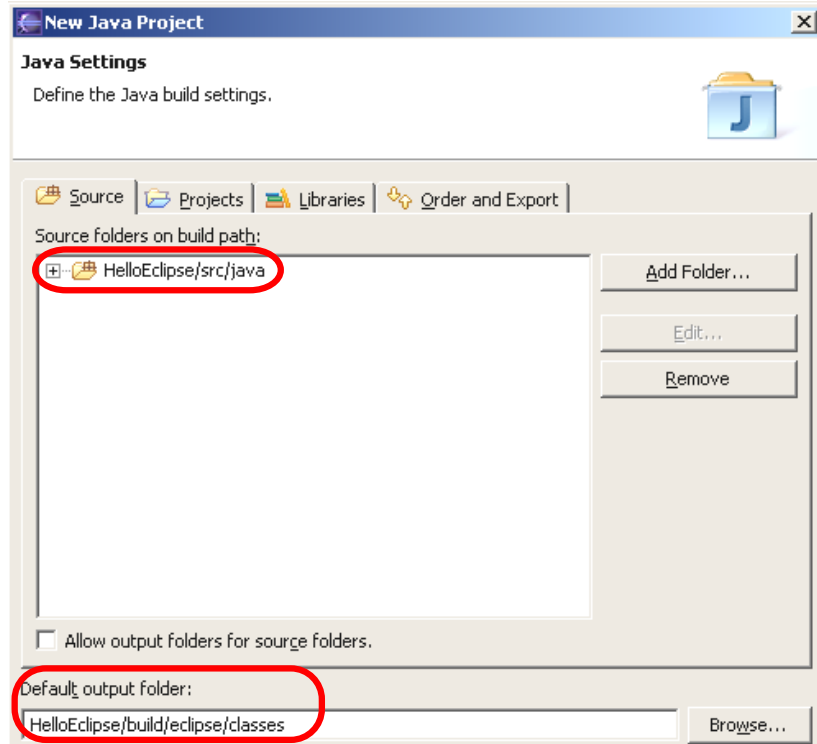
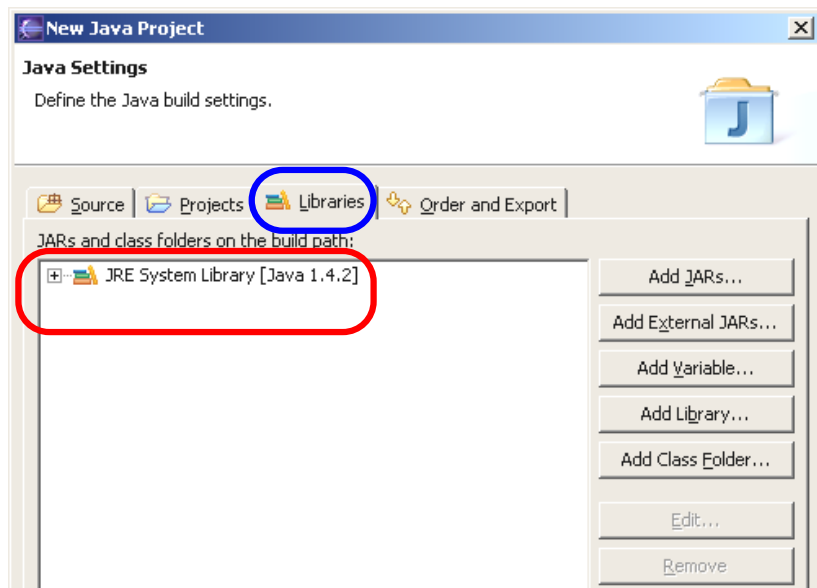


Figure 2

- **Source tab**—*no changes*.
- Note that Java source code files will be in the `src/java` subfolder.
- Note that the compiled `class` files will be under the `build/eclipse/classes` subfolder.
- **Projects tab**—*no changes*.



- **Libraries tab**—*no changes*.
- *This is where user-defined libraries are specified in more complex projects.*
- **Order and Export tab**—*no changes*.
- Press Finish to complete the project setup wizard.



Create a Class

- Right-click on the `src/java` folder and then select New/Class.



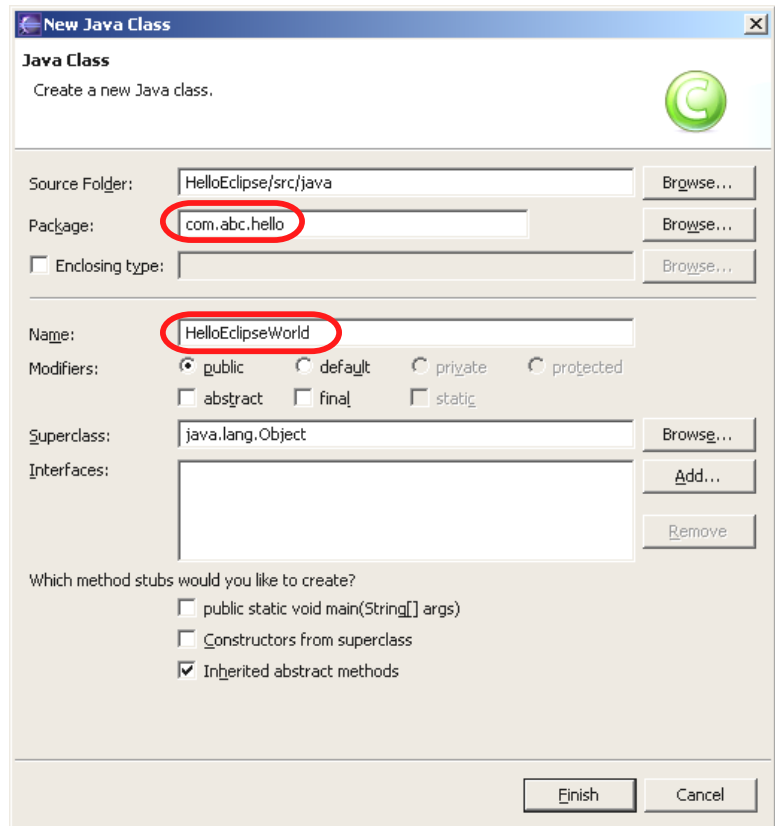
- Enter the package name:

`com.abc.hello`

- Enter the classname:

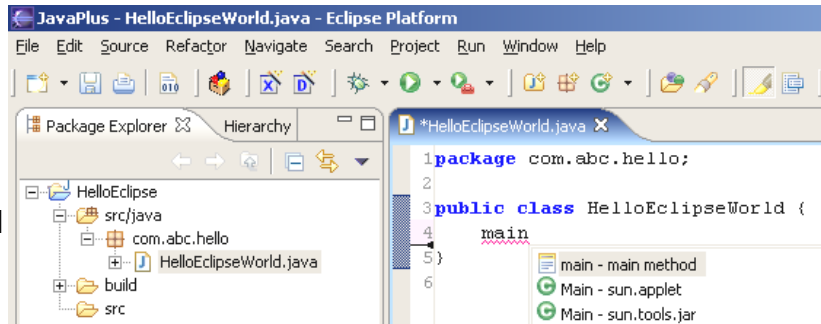
`HelloEclipseWorld`

- Press Finish.

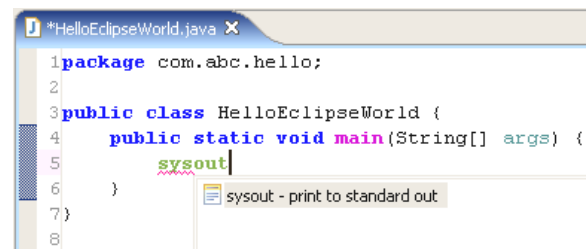


Edit the Java Source File

- Type in the contents of the source file.
- If you want to, you can use existing code templates and *Content Assist* to help write the code.
 - In this case, type `main`, then **Ctrl+Space** to activate Content Assist.
 - Select the first item (`main – main method`) to have the `main` method written for you from a template.



- Next, type `sysout`, then **Ctrl+Space** to activate Content Assist.
- Select the first item (`sysout – print to stdout`) to have `System.out.println();` written for you from a template.
- To see more of the templates that are available, go to **Windows/Preferences**. Then go into **Java/Editor/Templates**. You can also create your own templates to expand the power!
- The completed contents of `HelloEclipseWorld.java` should be:

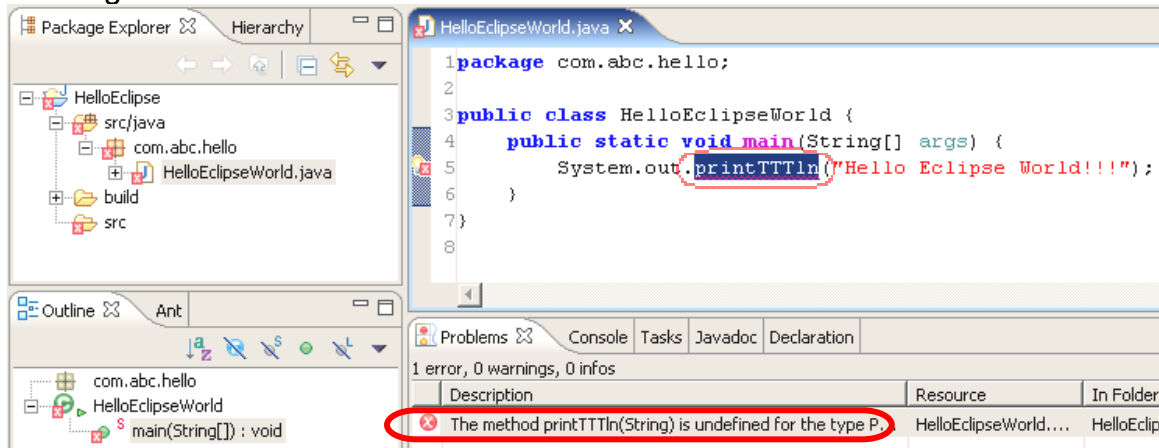


```
package com.abc.hello;

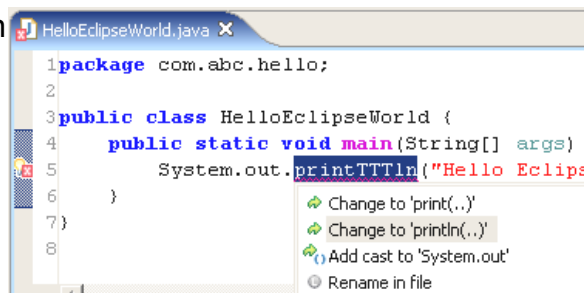
public class HelloEclipseWorld {
    public static void main(String[] args) {
        System.out.println("Hello Eclipse World!!!");
    }
}
```

Compilation

- To compile your Eclipse *project*, use **Ctrl+B** (menus: Project/Build All).
- Any compilation problems will show up in the *Problems* tab at the bottom of the screen. Double-clicking on a problem will open the associated file and jump to the offending line of code.



- When on the line with the error, the code can be manually corrected *or* the *Quick Fix* feature can be used to make suggestions by pressing **Ctrl+1** (menus: Edit/Quick Fix).
- Using Quick Fix, select the second suggestion to correct the code:

**Running the Java Application**

- For the *first* run, you need to let Eclipse know which of your classes should be used as the starting point (the class with the `main` method).
 - Right-click on the classname, select Run/Java Application.
- For *subsequent* runs, simply press **Ctrl+F11** (menus: Run/Run Last Launched). This step automatically:
 - Saves all modified files (prompted),
 - Compiles modified source files, and
 - Runs the application.

