

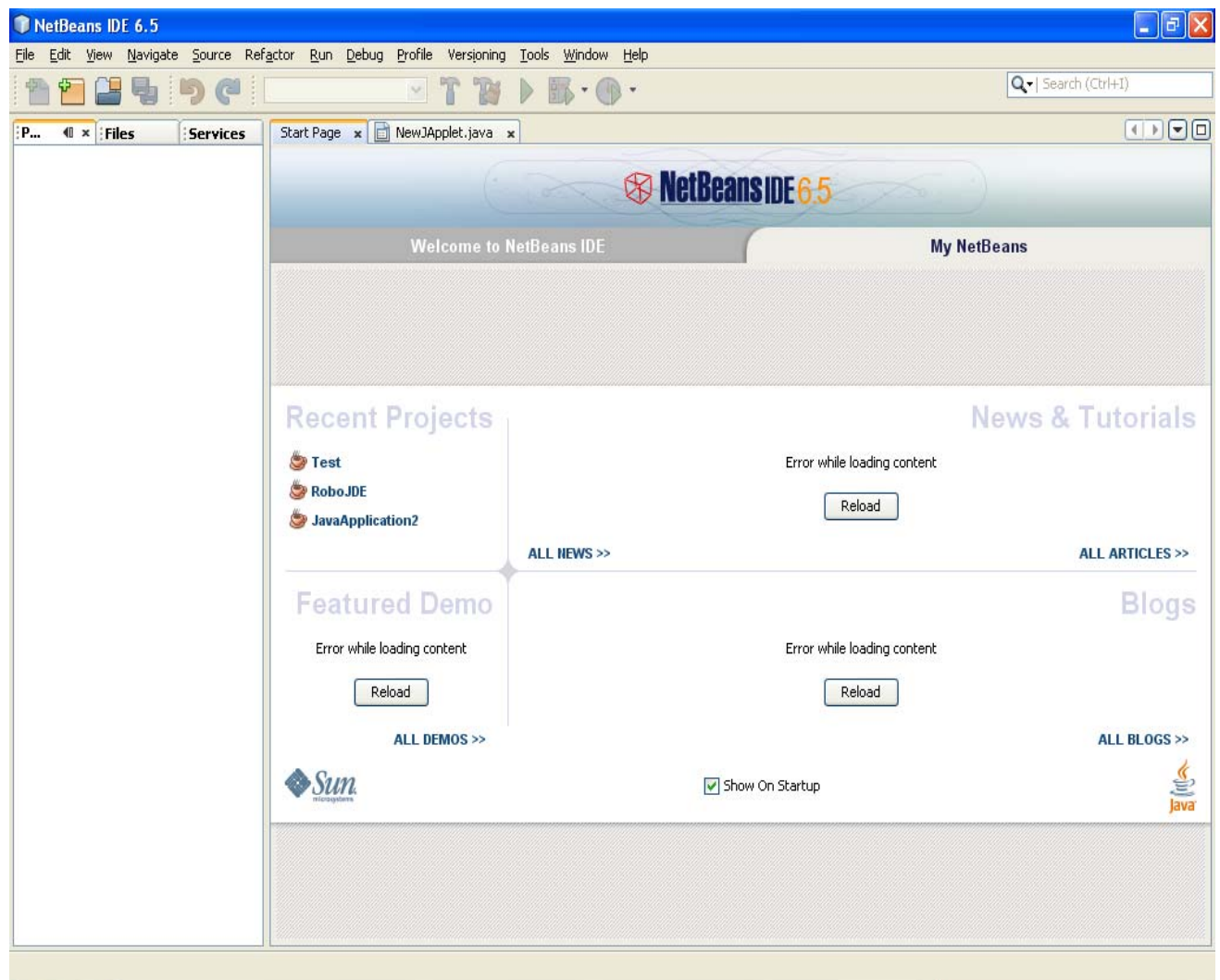
# CSCI 1301

## Lab 2 - NetBeans Basics

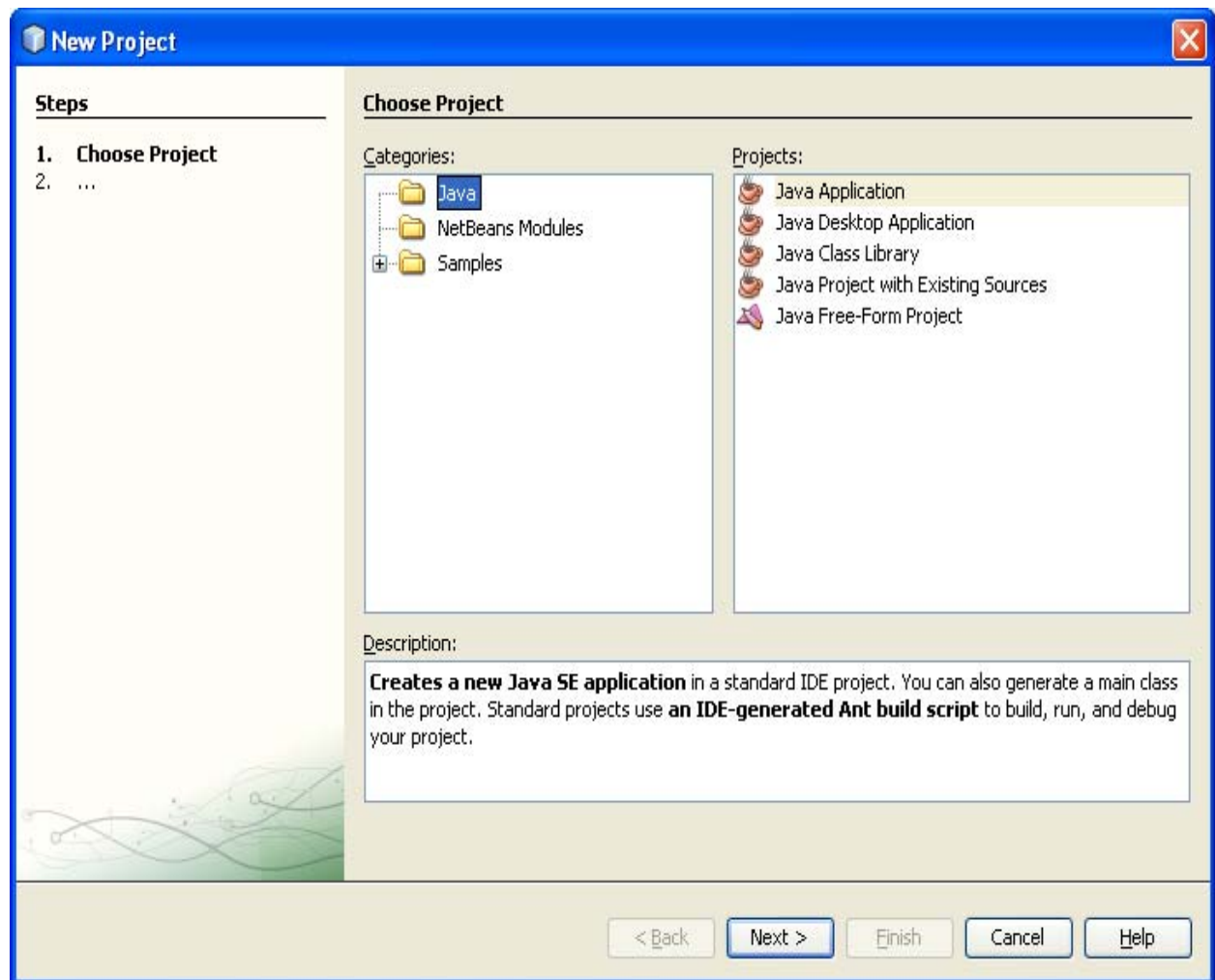
The purpose of this lab exercise is to become familiar with NetBeans, an Integrated Development Environment (IDE) that will assist you to develop your java programs. In this lab assignment, you will create, compile and execute a simple java program called HelloWorld.java in the NetBeans environment.

1. Before you start becoming familiar with NetBeans, create a folder on the desktop called CSCI 1301. Then click on **Start/All Programs/NetBeans/NetBeans 6.1**

You should now see this window.

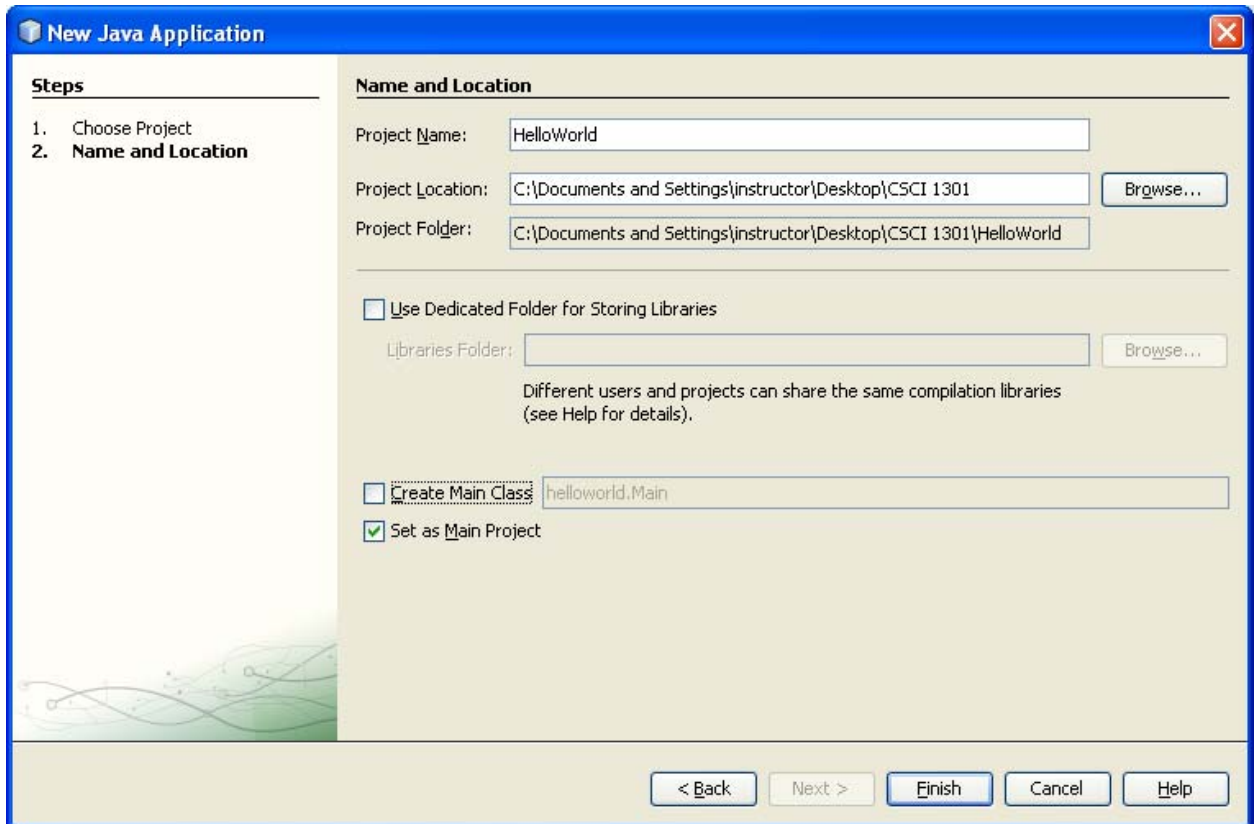


2. In NetBeans, a project is a collection of one or more java source code files with an extension .java. Let's learn how to create a simple java project having only one java source file. Click on "New Project" in the File. The following window will be displayed. Select Java Application and click on next.



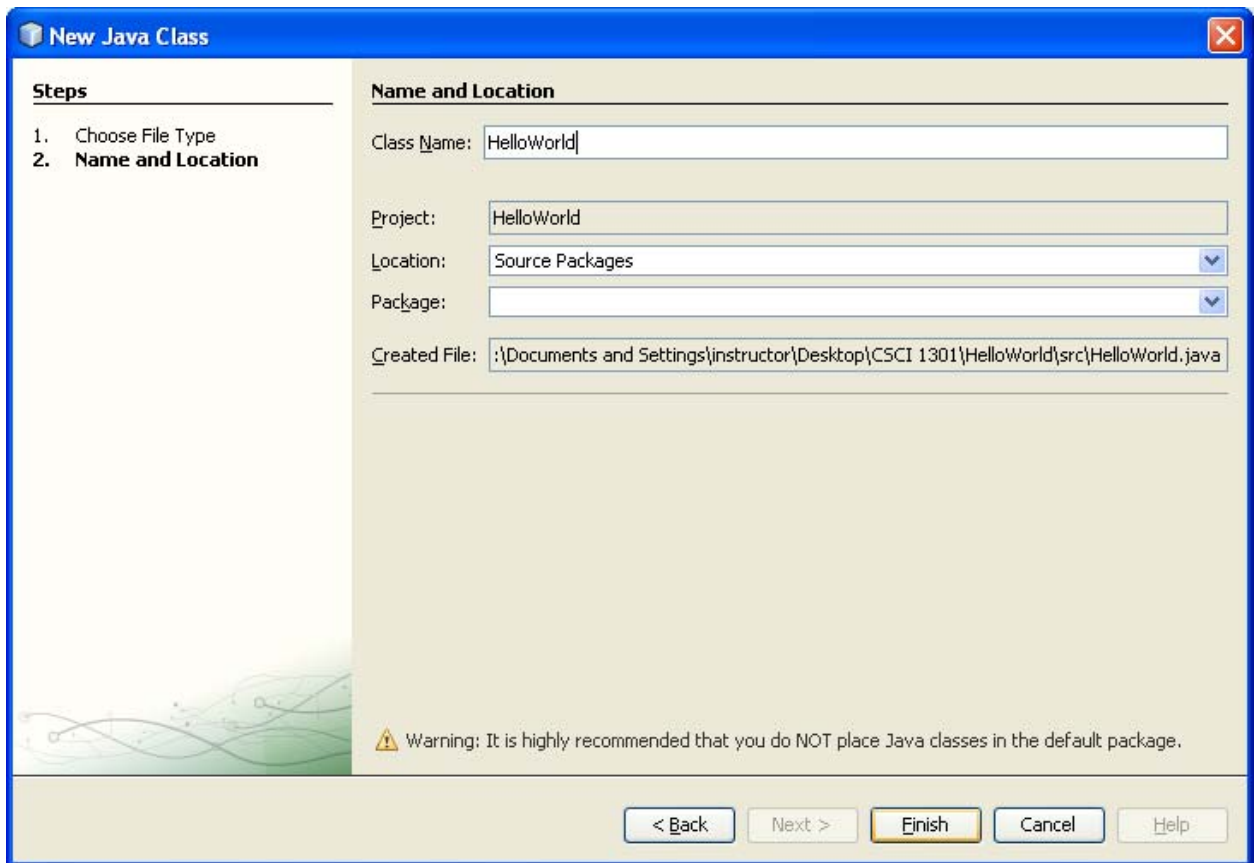
3. In the New Java Application window of the wizard, do the following (as shown in the figure below):

- In the **Project Name** field, type HelloWorld
- Select the folder CSCI 1301 you just created in Step 1 to save your first CSCI 1301 project. Use the **Browse** button at the right of the Project Location to locate such a folder.
- Uncheck “Create main class”
- Leave the Set as Main Project checkbox selected.



4. Click on the “**Finish**” button. When you instruct NetBeans to create a new project, it creates a folder with the name of your project in the **Project Location** you specified. In this example, NetBeans creates the folder HelloWorld in the folder CSCI 1301. In this folder, NetBeans also creates a folder called src that contains all the java source files of your project.

5. Right click on Source Packages, select New and then select Java Class. The following window will pop up.  
Write HelloWorld as in the *class name* field and press *finish*.



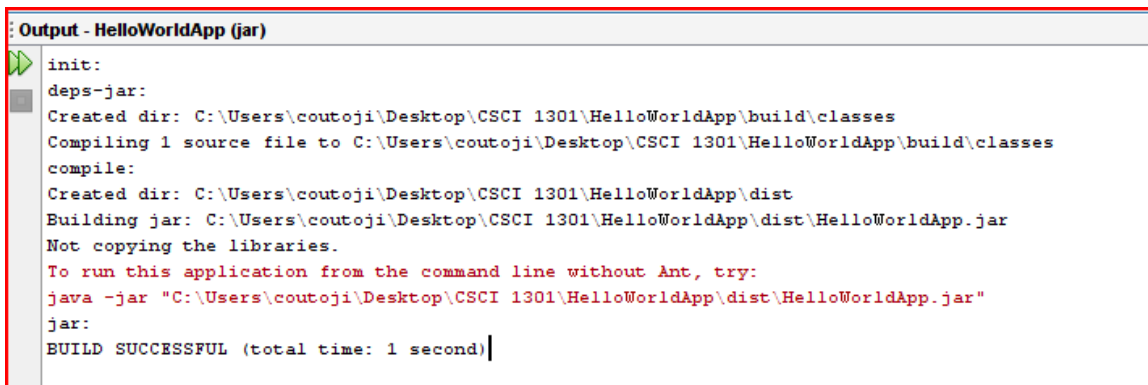
The Source editor window appears in the right window. You can enter and modify your Java source code using the source code editor. As you can observe, NetBeans has automatically generated a template of the class HelloWorld. The code of the new class is then displayed in the Source editor window. NetBeans source editor offers some helpful features that help you to enter and read Java source code in an easy manner. For example,

- Tabs are used to indent several parts of the program. This makes your code more readable and easier to debug.
- Words in blue are keywords in Java, words that belong to the Java's vocabulary. We will learn the specific function of this word throughout the course.
- Lines in grey are comments that provide documentation to your program but are not part of the Java code meaning that comments will not be executed by the Java VM.
- Braces and parenthesis come in pairs. If you place the cursor on either side of some of the braces (and parentheses) NetBeans will highlight its corresponding partner. This feature is useful when you need to find mismatched or unmatched braces or parenthesis in your code, which are common sources of syntax errors.

- Click on the editor window. Create a main method header and its body. In the main method add the line  

```
System.out.println("Hello World!");
```

  - Notice that the phrase “Hello World!” is in orange, which means it is a String literal. A String literal in java is a sequence of characters enclosed between double quotes. In this example, the String literal is used to display a greeting message to the user.
- In the **Build** menu, choose “Build Project”. This will run the *javac* compiler against all of the source code you have associated with this project. At this point, the Output window opens and displays output similar to this:



```
Output - HelloWorldApp (jar)
init:
deps-jar:
Created dir: C:\Users\couto\ Desktop\CSCI 1301\HelloWorldApp\build\classes
Compiling 1 source file to C:\Users\couto\ Desktop\CSCI 1301\HelloWorldApp\build\classes
compile:
Created dir: C:\Users\couto\ Desktop\CSCI 1301\HelloWorldApp\dist
Building jar: C:\Users\couto\ Desktop\CSCI 1301\HelloWorldApp\dist\HelloWorldApp.jar
Not copying the libraries.
To run this application from the command line without Ant, try:
java -jar "C:\Users\couto\ Desktop\CSCI 1301\HelloWorldApp\dist\HelloWorldApp.jar"
jar:
BUILD SUCCESSFUL (total time: 1 second)|
```

If the compilation step is successful, you will see the statement **BUILD SUCCESSFUL** at the bottom of the output window. If the compiler finds syntax errors in your program, the message **BUILD FAILED** will be displayed as well as a list of all the syntax errors found. Syntax errors are reported as hypertext link, clicking on an error hyperlink will navigate to the source of the error in your code. After you locate the error, you must fix it and build the project again until the program is successfully compiled.

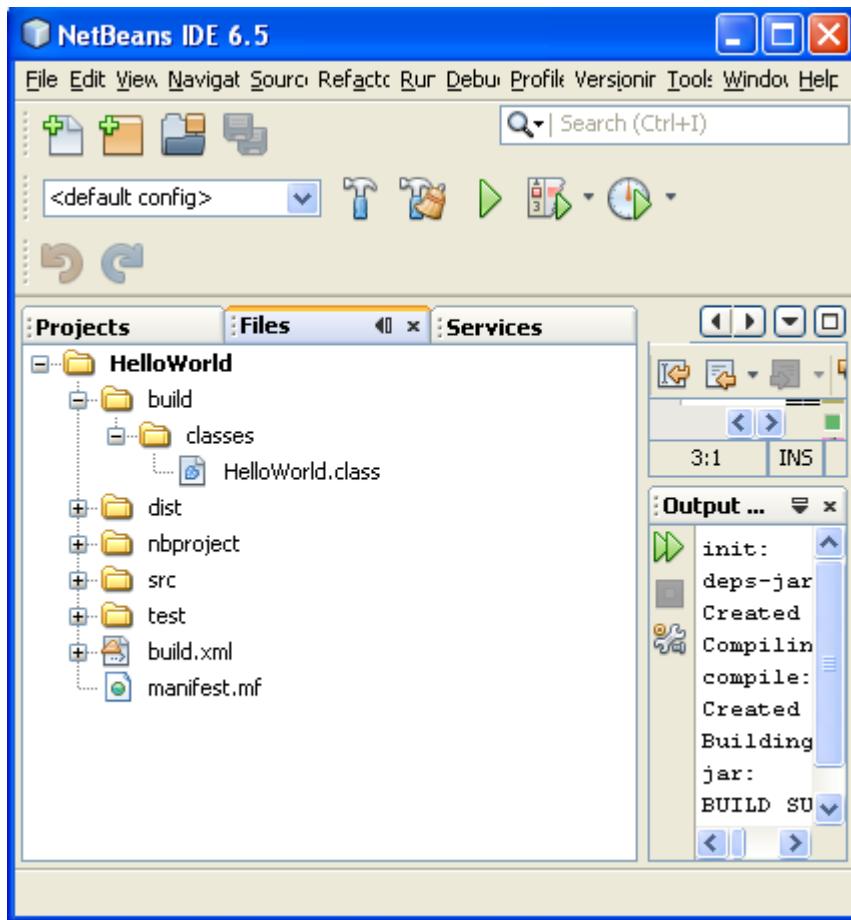
If your source code has no syntax errors, the Java bytecode file `HelloWorld.class` is generated in the folder *build/classes/* in the *classes* folder. You can see where the new file `HelloWorld.class` is generated by clicking on the *build/classes* folder in the *Files* tab.

- To run your first Java program, click on **Run Main Project** in the **Run** menu. The output window will display something like this

```
Output - HelloWorldApp (run)
init:
deps-jar:
compile:
run:
Hello World!
BUILD SUCCESSFUL (total time: 1 second)
```

Note that the message Hello World! is displayed in the Output Window.

9. Click on the *Files* tab on the left to see a folder-based view of your project, including files and folders. Expand the folders *build* and *src* until you see something like this:



10. Before you exit NetBeans, click Close “HelloWorld” on the File menu. If NetBeans prompts you to save the modifications, click on Save All. Finally, click on Exit in the File menu.
11. Finally, you should submit the file HelloWorld.java in WebCT to get credit for this lab exercise. To do so:
  - Login into WebCT and click on the Project Assignments link in the Course Menu or click on Course Content/Project Assignments.
  - Click on the Lab Assignment #2 link.
  - Click on the **Upload file** button.
  - Click on the **Browse** button that appears in the **Upload File for Lab Assignment 2** window.
  - Click on the file you want to upload and afterwards click on **Open**.
  - Click on the **Upload** button and locate the **HelloWorld.java** you have created in the previous steps. The file **HelloWorld.java** should be in the folder *HelloWord/src/* in the CSCI1301 folder you have created in your computer.

**Important:** You should submit the **JAVA** file (HelloWord.java) and **NOT** the **CLASS** file (HelloWord.class).

- Type your e-mail address under **Notification** to receive an e-mail confirming your assignment has been submitted.
- Click on the **Submit Assignment** button.  
An assignment confirmation page will be displayed just after you submit your file.

**Important:** Remember, you must upload the file you want to submit and then click the **Submit** button. Otherwise, if you upload your file(s), but forget to click on the **Submit Assignment** button, the file(s) have not been submitted to be graded.

- If you want to check **the submission status of your lab assignment**, click on **Project and Lab Assignments**. **The submission status of a project assignment is shown under the project's link.**

## *Writing a Java Program using a Text Editor*

Another way to write java source code is by using a text editor such as Notepad, Wordpad or Crimson editor instead of an IDE such as NetBeans. If you wish to create your java program in this way, open a text editor of your choice (textpad , for example), and create a file called “HelloWorld.java”. Enter the following code into the file:

```
public class HelloWorld {
```

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

Save the file and open a command line interface. In Windows, you'll want to run the **cmd** command (*Start/Run*) or find the *Command Prompt* option in *Accessories*. Change into the directory in which you created your HelloWorld.java file using the *cd* command followed by the path to the directory in which you have saved your java file. Afterwards, run the following command:

```
> javac HelloWorld.java
```

This compiles the file you created using the *javac* compiler from your Java JDK. If there are no errors, there should now be a file called HelloWorld.class in your current directory. Now run the following command:

```
> java HelloWorld
```

This command runs the compiled HelloWorld bytecode through the Java runtime environment. You should see the following output to your command line:

```
Hello World!
```