

How to Run a Java Program

CSE 1310 – Introduction to Computers and Programming
Vassilis Athitsos
University of Texas at Arlington

Initial Steps

- You must install Java SDK.
- You must install Netbeans.
- These are steps that you do just once. Once Java and Netbeans are installed, you can run any Java program you like on your computer.

Installing Java SDK

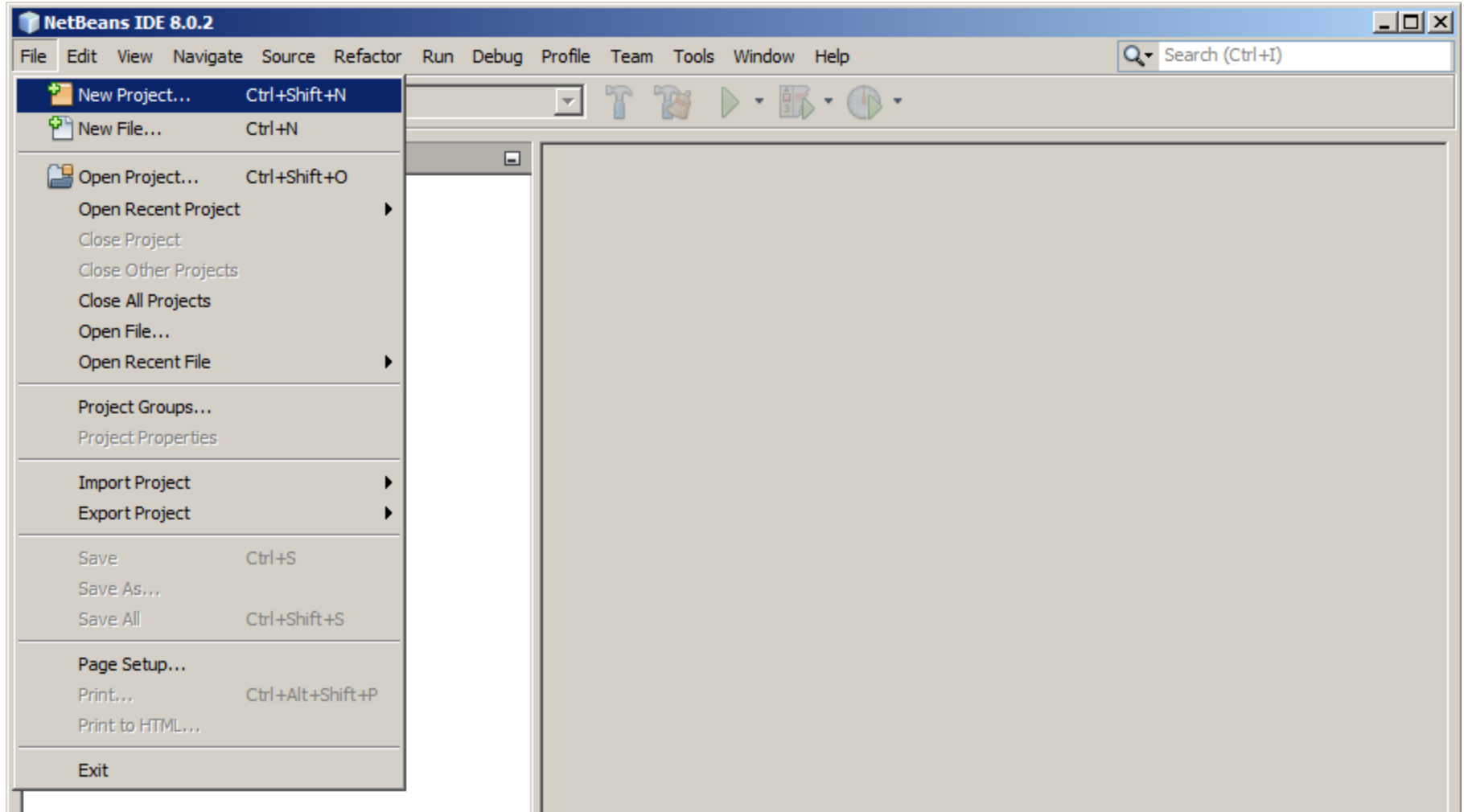
- As of August 24, 2016, the current version of Java SDK is Version 8, Update 102 (Java SE Development Kit 8u102).
- To install the current version of Java SDK:
 - Go to:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
 - Click on "JDK Download"
 - Follow the instructions on screen to complete the installation process.
- If you have trouble installing, ask a TA for help during lab hours.

Installing Netbeans IDE 8.1

- As of August 24, 2016, the current version of Netbeans is 8.1
- To install the current version of Netbeans:
 - Go to <https://netbeans.org/downloads/index.html>
 - Download the "Java SE" version
 - Follow the instructions on screen to complete the installation process.
 - You will be asked about JUnit: choose to install it.
- If you have trouble installing, ask a TA for help during lab hours.

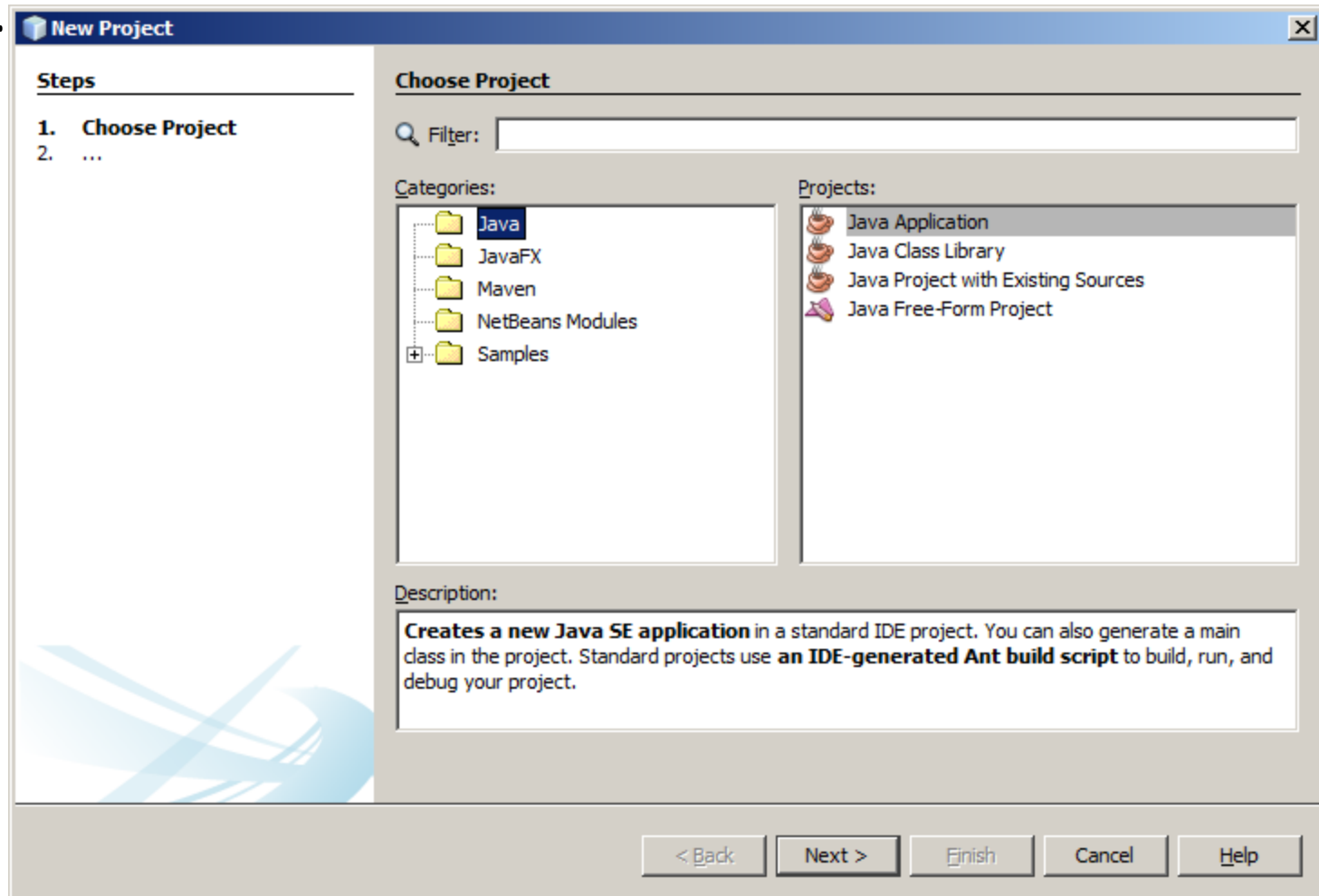
Creating a Netbeans Project (1)

- Start Netbeans.
- Go to File->New Project



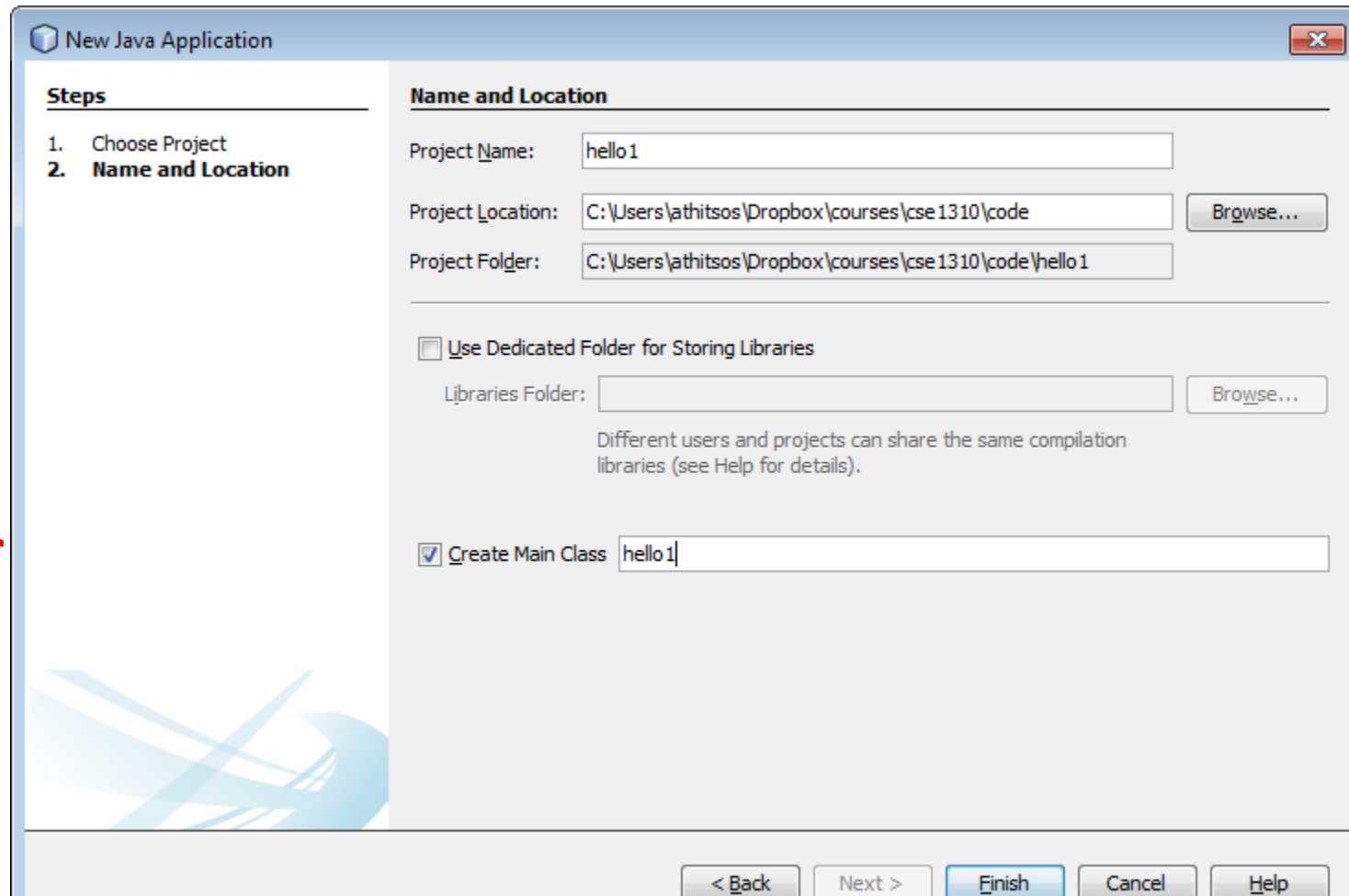
Creating a Netbeans Project (2)

- Under Categories, select "Java".
- Under Projects, select "Java Application".
- Click "Next".



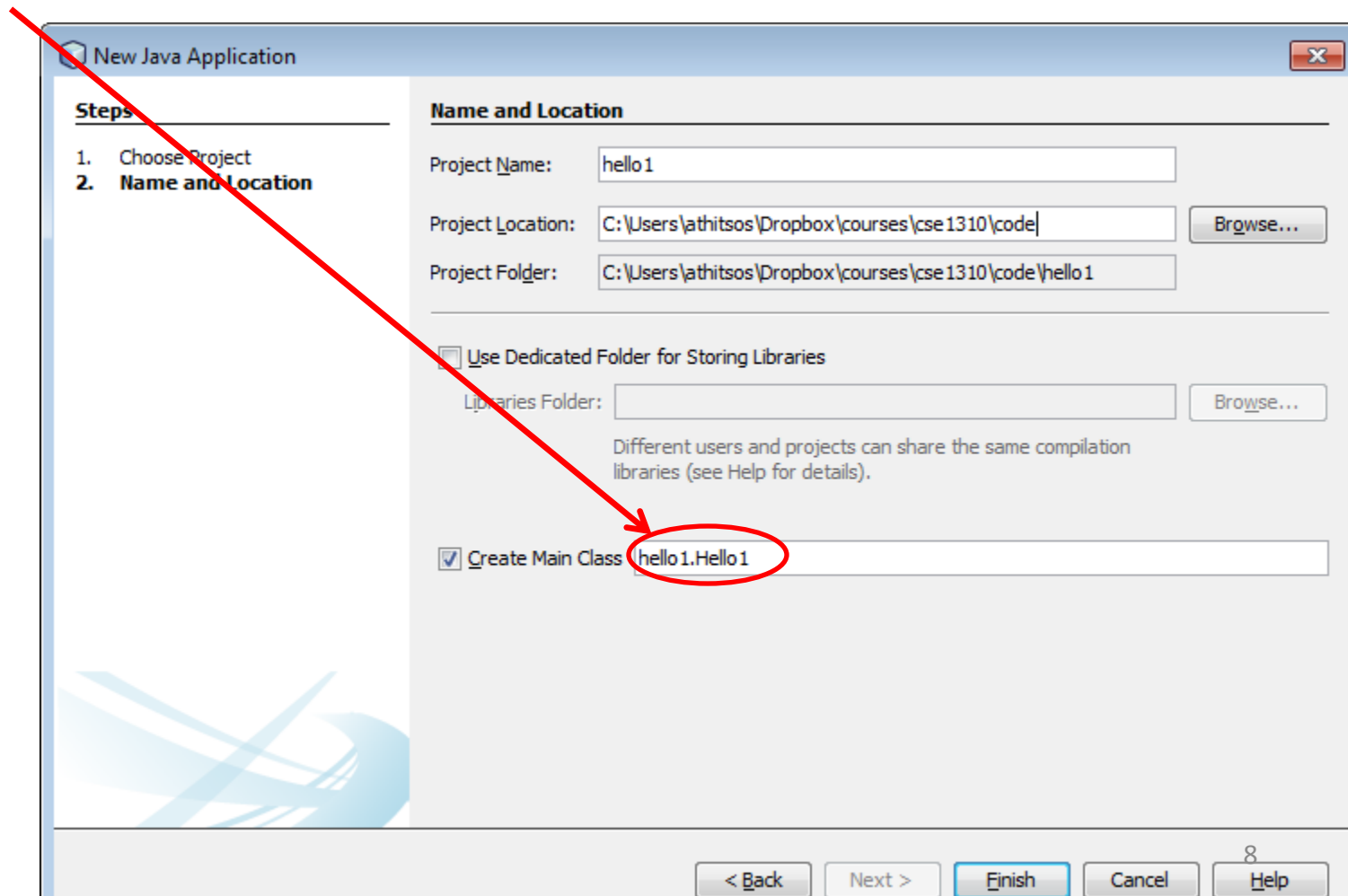
Creating a Netbeans Project (3)

- Give a name to your project, e.g., hello1.
- Give to your main class **the same name as for your project**, and click "Finish".
- **Names matter, see next slides.**
- **Students lose lots of points each semester for ignoring the instructions in the next 3 slides.**



Choosing a Name for the Main Class

- Netbeans suggests a default name for the main class.
- For example, if the project name is hello1, Netbeans suggest name hello1.Hello1 for the main class.



Choosing a Name for the Main Class

- Netbeans suggests a default name for the main class.
- For example, if the project name is hello1, Netbeans suggest name hello1.Hello1 for the main class.
- **You must override that**, with hello1 in this example.

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class

< Back Next > Finish Cancel Help

Choosing a Name for the Main Class

- Netbeans suggests a default name for the main class.
- For example, if the project name is hello1, Netbeans suggest name hello1.Hello1 for the main class.

- **You must override that**, with hello1 in this example.

- The main class name should be the same as the project name.

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name: hello1

Project Location: C:\Users\athitsos\Dropbox\courses\cse 1310\code Browse...

Project Folder: C:\Users\athitsos\Dropbox\courses\cse 1310\code\hello1

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder: Browse...

Different users and projects can share the same compilation libraries (see Help for details).

☒ Create Main Class hello1

< Back Next > Finish Cancel Help

Writing Code (1)

- If you create a new project as described, Netbeans will create a Java file, that looks like this:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author athitsos
 */
public class hello1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Writing Code (2)

- The place where you put your code is indicated with the text
`// TODO code application logic here`

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author athitsos
 */
public class hello1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Writing Code (3)

- For the first assignments, all you have to do is replace that "TODO" line with your own lines of code.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author athitsos
 */
public class hello1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Writing Code (4)

- Text that starts with `/*` and ends with `*/` is NOT PART OF THE PROGRAM.
- This text is called "comments", and is meant to be read by humans, not the computer.
- We will usually not care about (and not show) the comments that Netbeans generates. You can delete them if you want.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */

/**
 *
 * @author athitsos
 */
public class hello1 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

Writing Code (5)

- Without the comments, the initial code that Netbeans generates looks more simple:

```
public class hello1 {  
    public static void main(String[] args) {  
        // TODO code application logic here  
    }  
  
}
```

A First Example

- Remember: for a good part of the class, your code will go where the TODO placeholder is.

```
public class hello1 {  
    public static void main(String[] args) {  
        // TODO code application logic here  
    }  
}
```

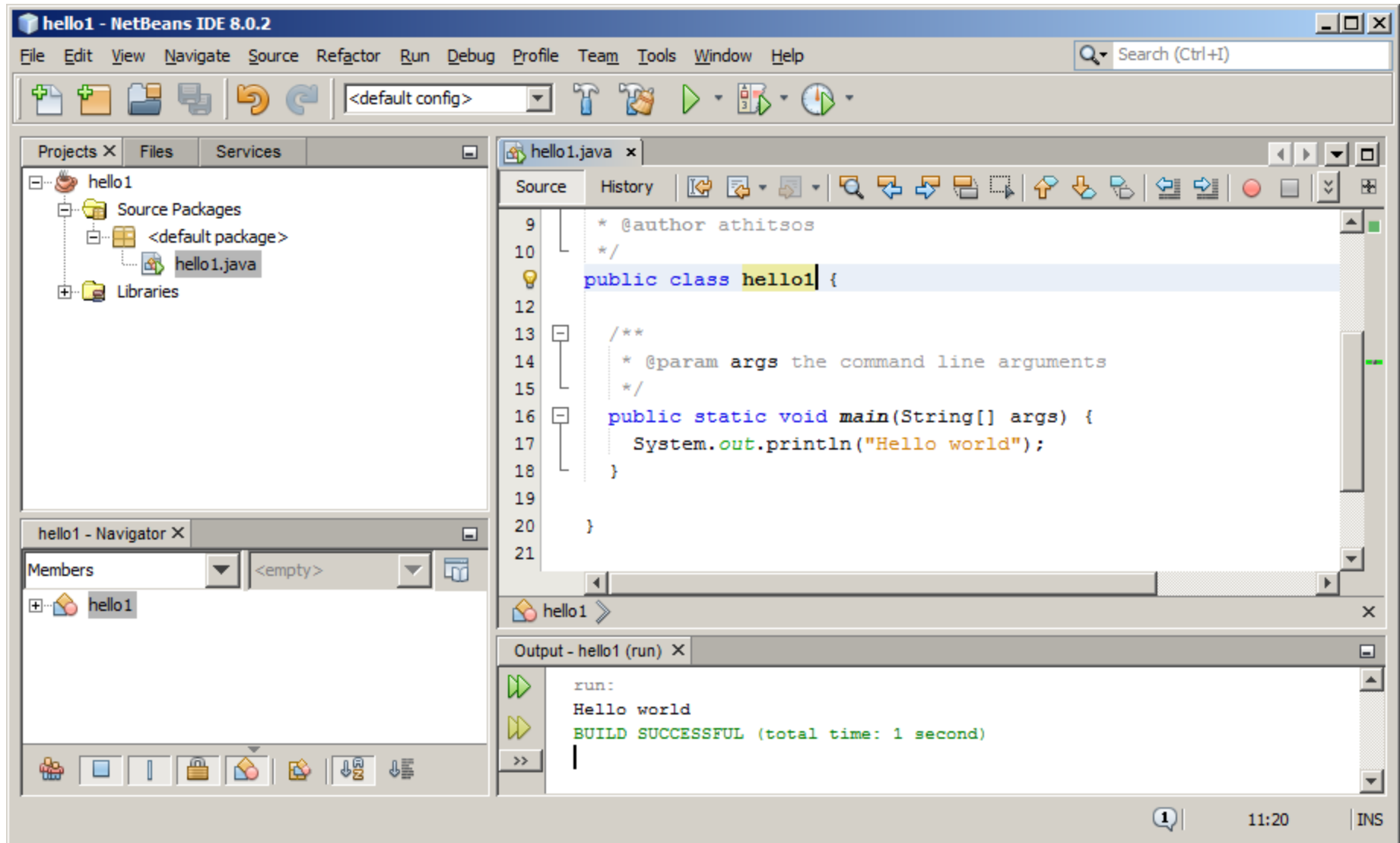

A First Example

- Here we have replaced the TODO placeholder with a line of code.

```
public class hello1 {  
    public static void main(String[] args) {  
        System.out.println("Hello world");  
    }  
}
```

- To run this code, go to Run->Run Project (or press F6) to compile the code.

- When you run your code, an "Output Window" appears at the bottom, that looks as shown.



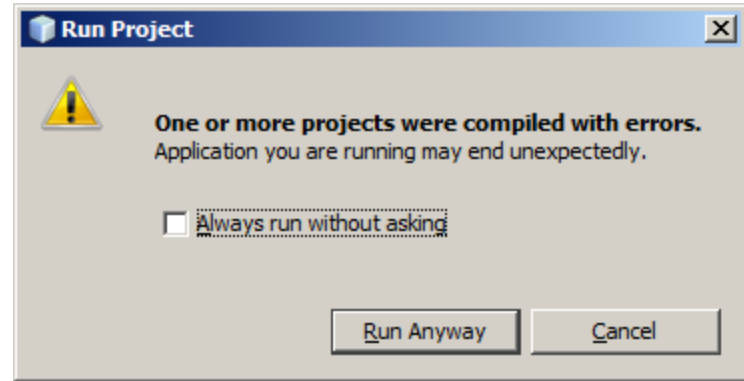
Failure to Run

- A lot of times you will get errors, and you will not be able to run the program.
- Do not panic, this is a very common thing to happen.
- For example, on this code we introduce an error on purpose, to see what happens (we delete the semicolon after "Hello world").

```
public class hello1 {  
    public static void main(String[] args) {  
        System.out.println("Hello world")  
    }  
}
```

- Now, when we try to run this code, we get what is shown on the next slide:

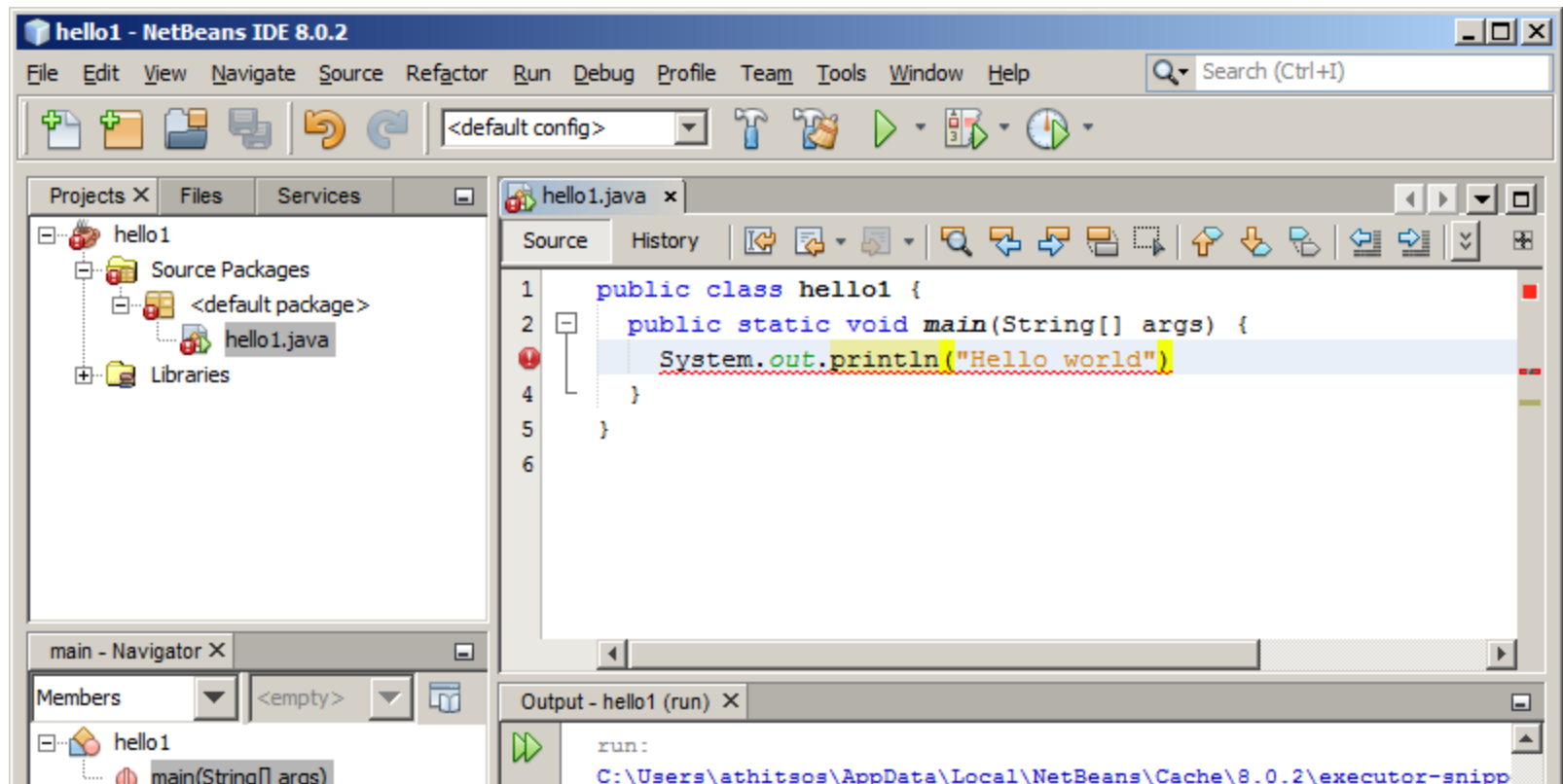
Error Message



- You will be seeing this window a lot.
- When you see it, click "Cancel".
- Then, click on "Run->Clean and Build Project".
- Now, the output window will show you an error message, which you should try to fix.
- You will learn how to fix these errors during the course.
- After you fix the error, you can try running the code again. If you get more errors, keep fixing them till the code runs.

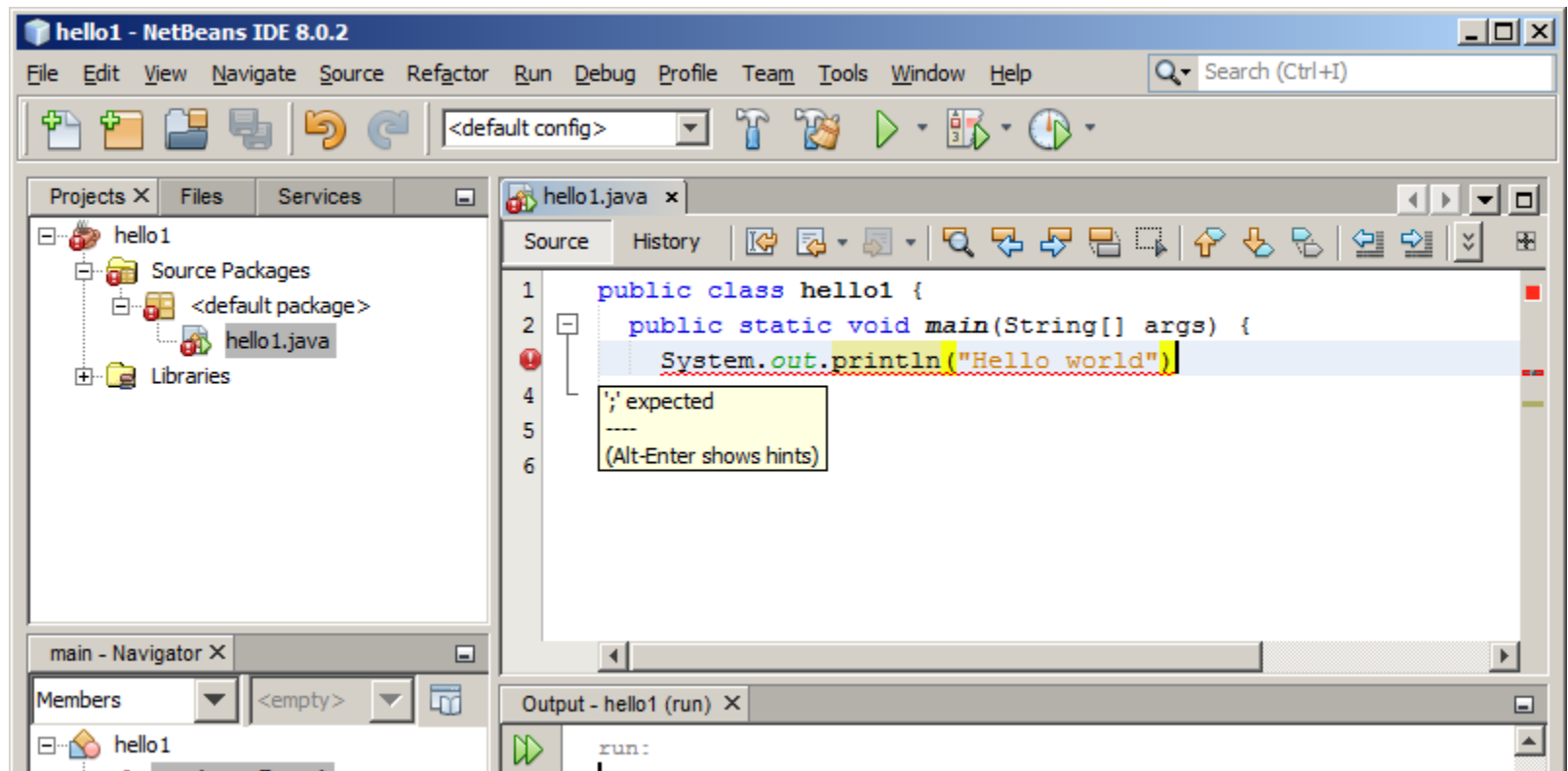
Netbean Real-Time Error Detection

- Note the ! sign on line 3 of the code.
- This is Netbean's way of telling you that that line is not correct.



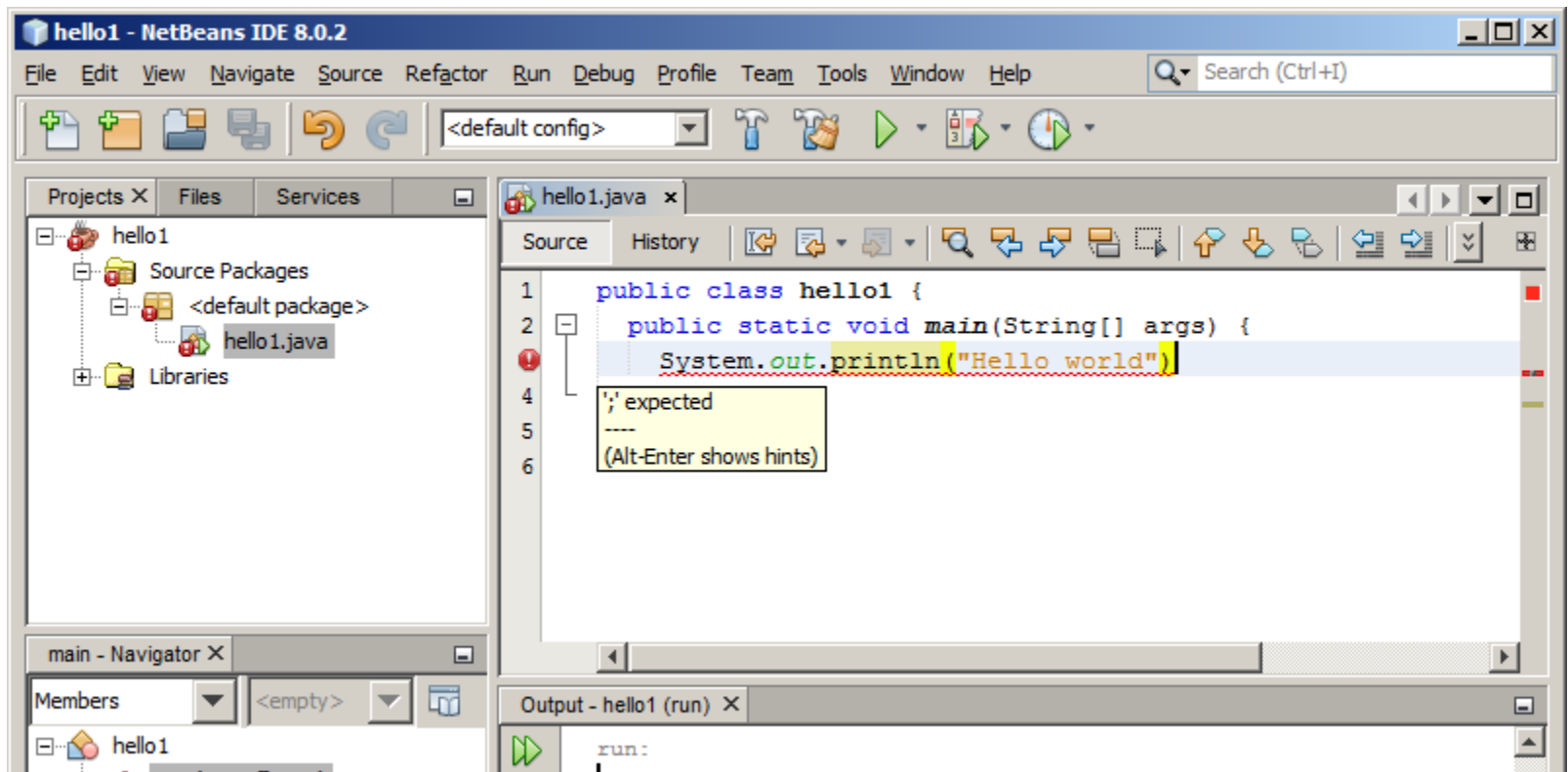
Netbean Real-Time Error Detection

- If you put the mouse arrow over the ! sign on line 3, you see an error message.
- In this particular case, it says: ';' expected.
- This means that you forgot to put the semicolon at the end.



Netbean Real-Time Error Detection

- You should look for these ! signs, and fix the errors, before you try to run your code.



Running Existing Code

- Oftentimes you may want to run code from somewhere (the slides, the textbook, the course website, etc.).
- To do that, you need to make sure that your project and main class are named appropriately.
- Step 1: Look at the code that you want to run, and find the name that shows up right after "public class".
- Step 2: When you create your project on NetBeans, use that name for your project and your main class.
- Step 3: Copy the code that you want to run to the .java file that NetBeans created.

Running Existing Code – Step 1

- Step 1: Look at the code that you want to run, and **find the name that shows up right after "public class"**.
- For example, in this piece of code, what is the name that shows up right after "public class"?

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

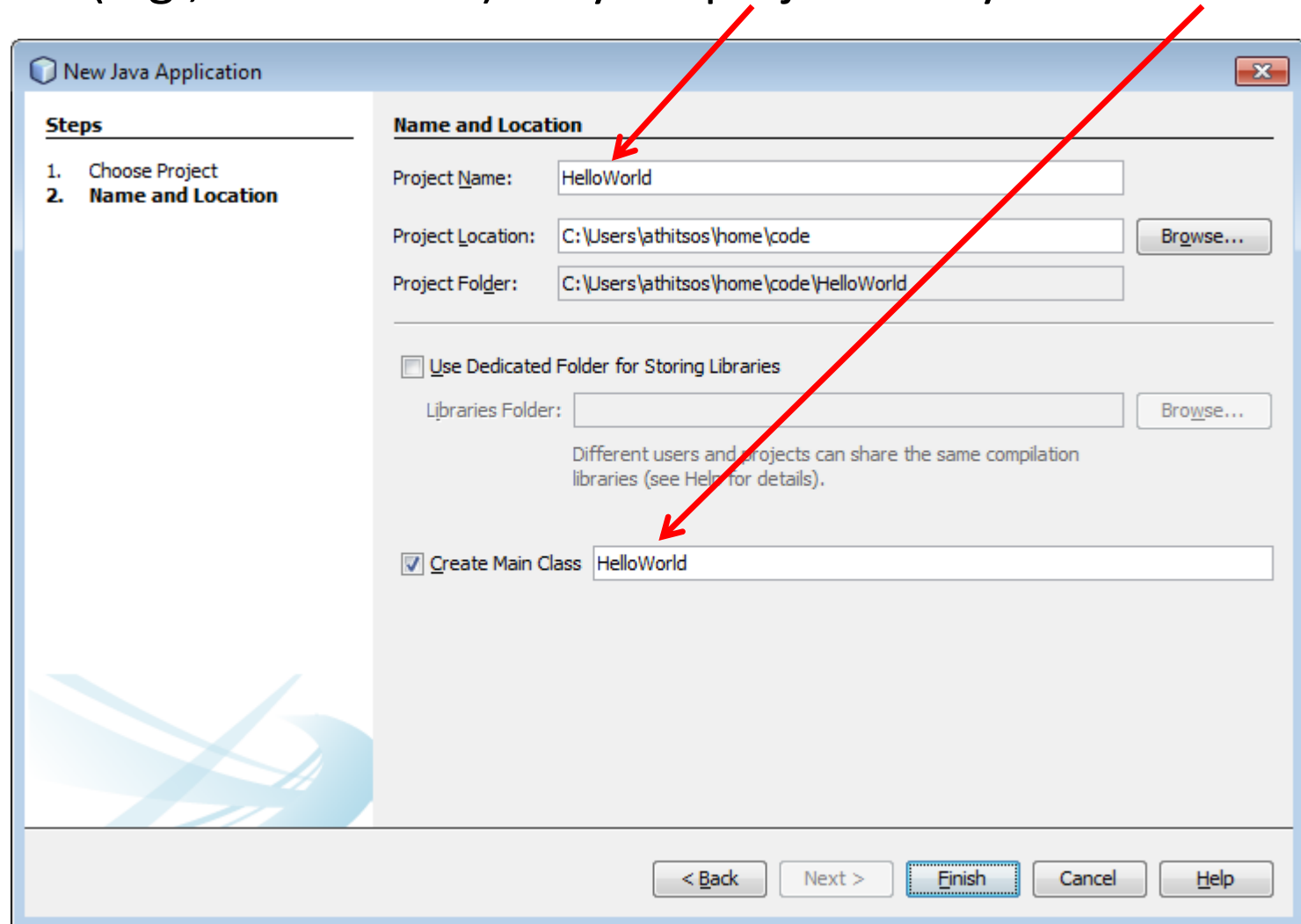
Running Existing Code – Step 1

- Step 1: Look at the code that you want to run, and find the name that shows up right after "public class".
- For example, in this piece of code, the name that shows up right after "public class" is **HelloWorld**.

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

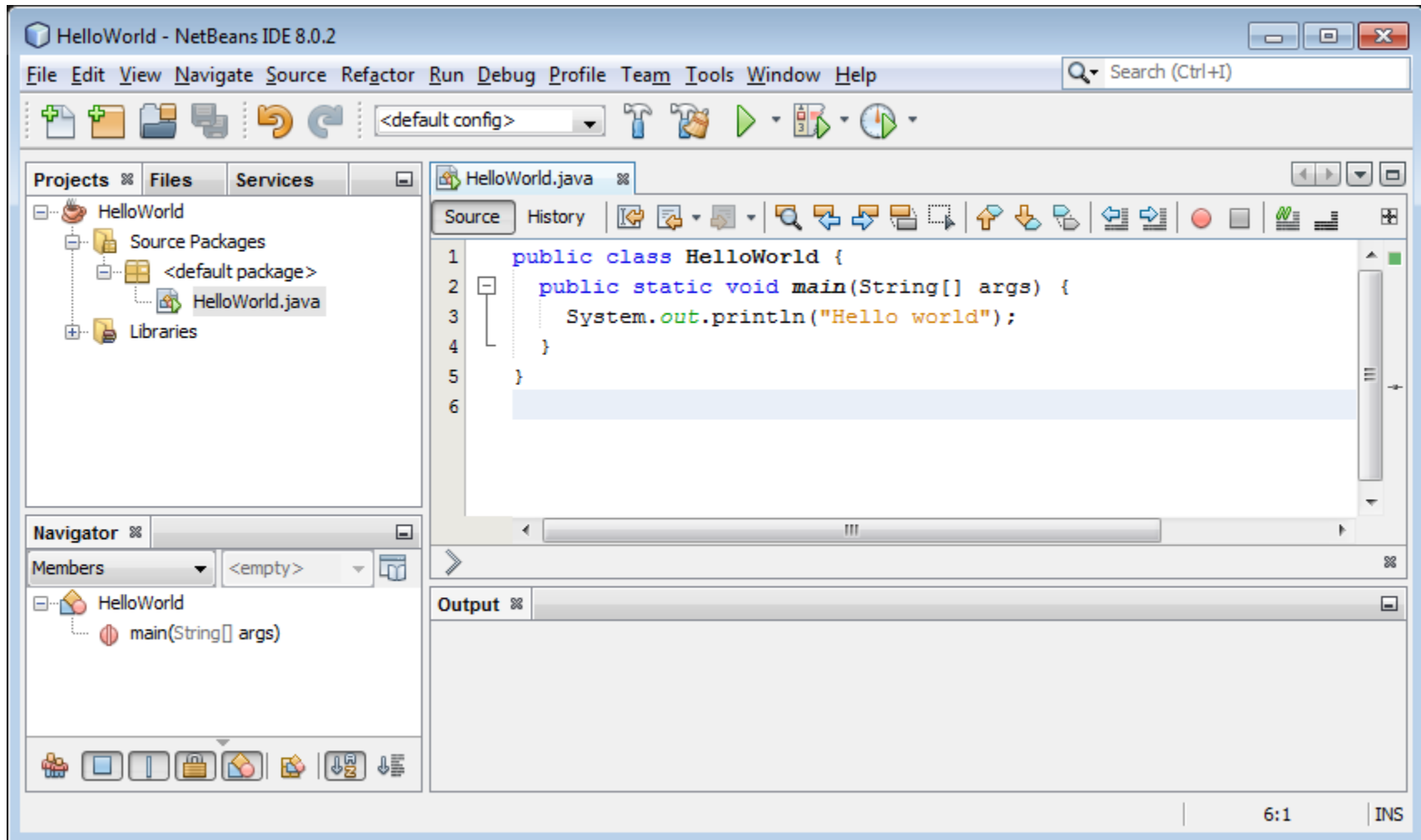
Running Existing Code – Step 2

- Step 2: When you create your project on NetBeans, use that name (e.g., **HelloWorld**) for your project and your main class.



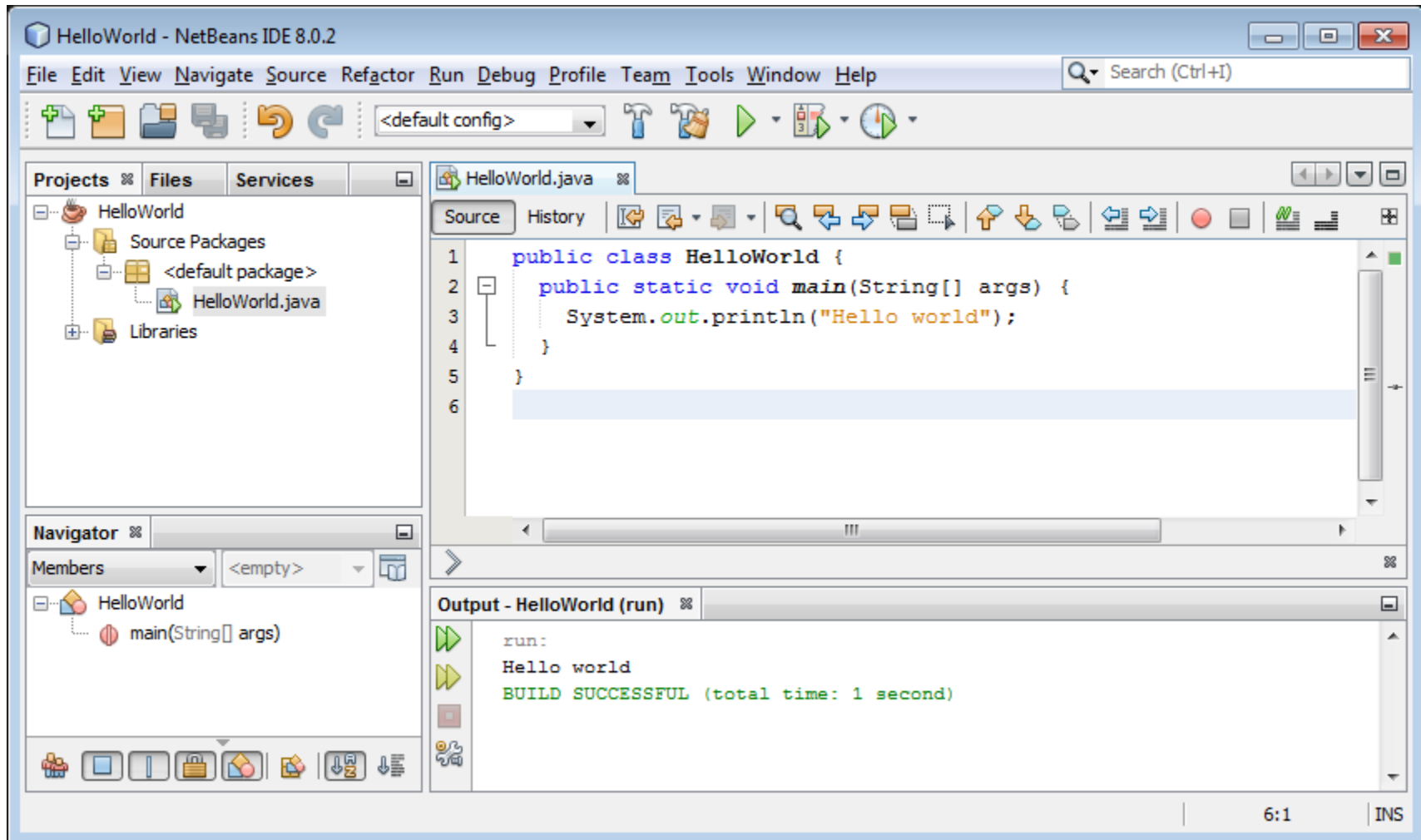
Running Existing Code – Step 3

- Step 3: Copy the code that you want to run to the .java file that NetBeans created.



Running Existing Code

- Now you can run the code by pressing F6 (or by selecting Run->Run Project from the top menu), and see the output.



Running Existing Code – Another Way

- Suppose that you already have a Netbeans project open, with main class name xyz (and the corresponding Java file called xyz.java).
 - This name, xyz, is just an example, you can do the same thing regardless of the name of the main class.
- Suppose that YOU CAN DISCARD THE ENTIRE CONTENTS of file xyz.java. (if you need to keep those contents, do not use this approach).
- Step 1: Erase the entire contents of your xyz.java file (select all contents, and press backspace).
- Step 2: Copy to your xyz.java file the code that you want to run.
- Step 3: In the xyz.java file, look at the code that you want to run, and find the name that shows up right after "public class". REPLACE that name with xyz.

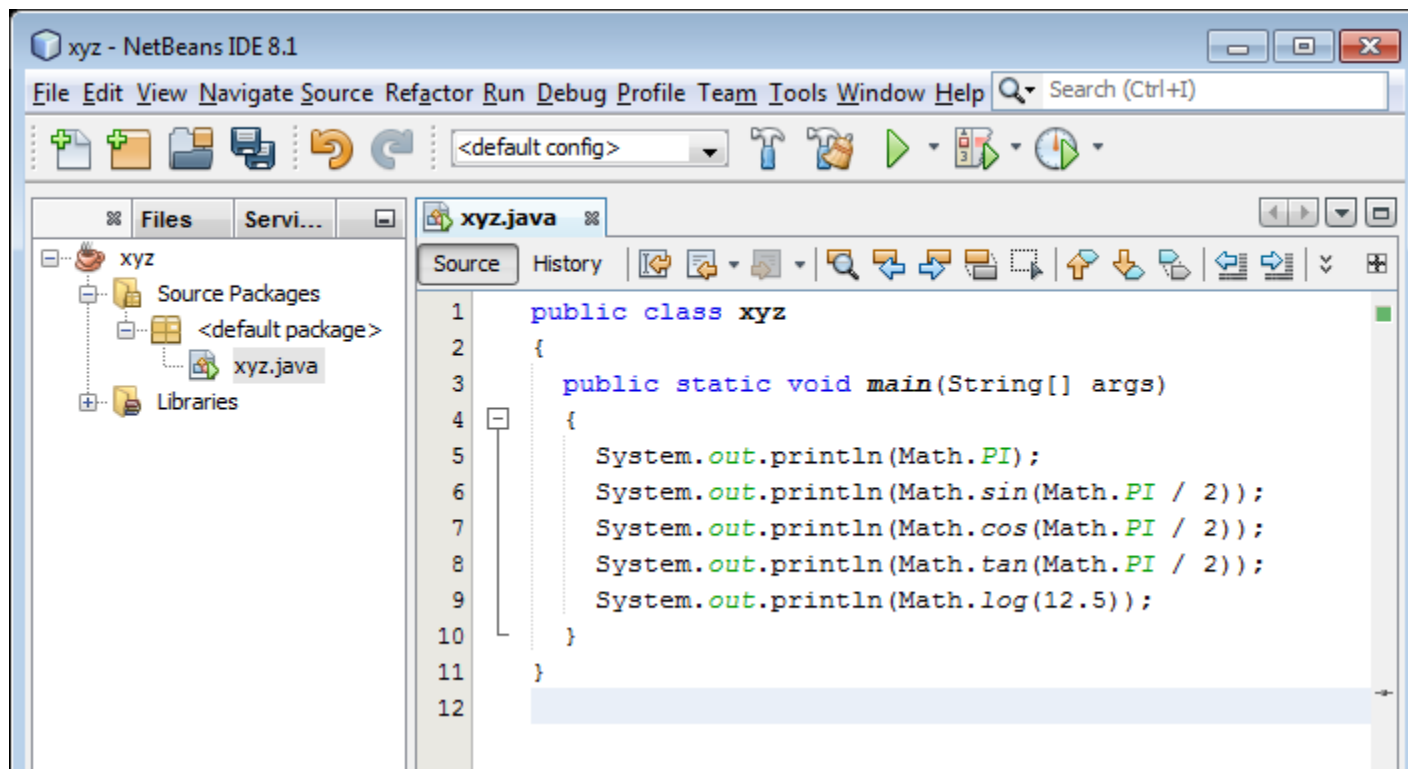
Running Existing Code – Another Way

- Let's see this second approach in more detail.
- As before, suppose that you want to run this Java code.

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

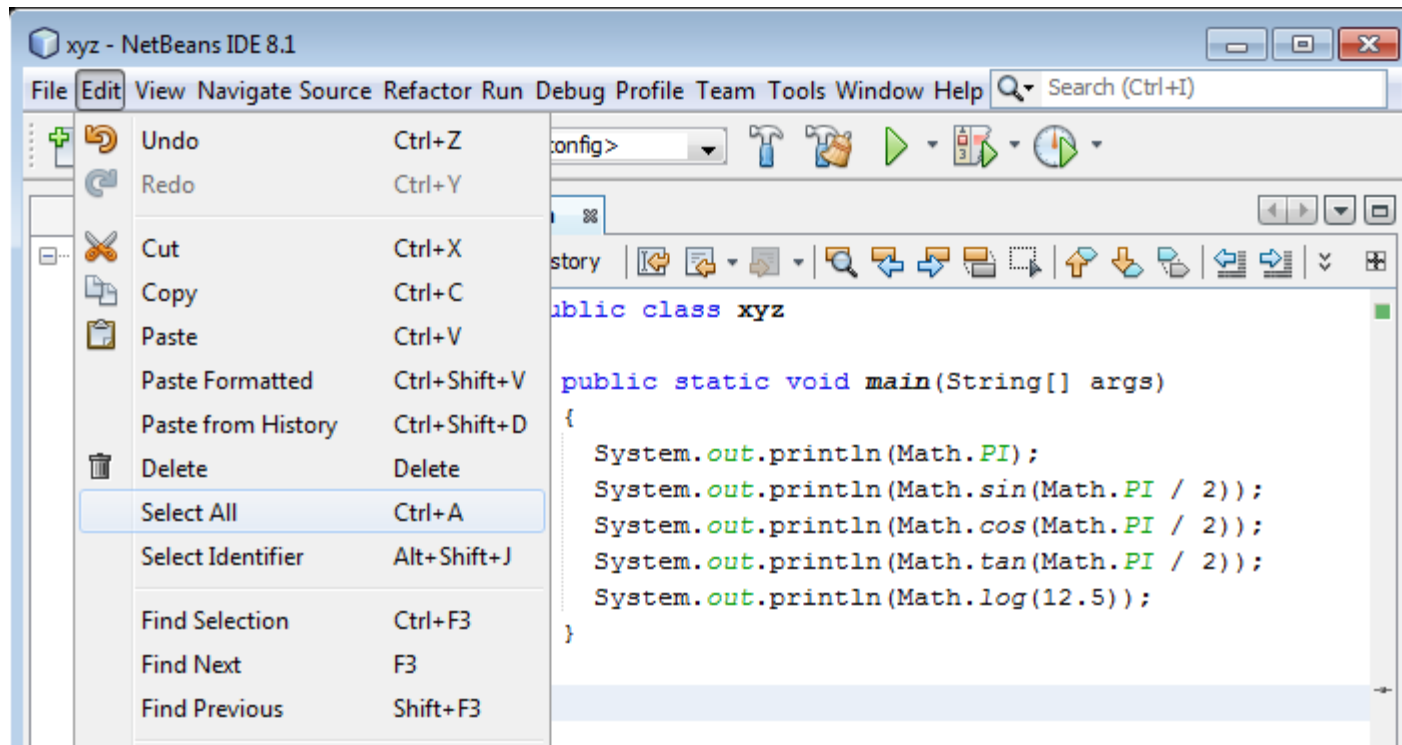
Running Existing Code – Another Way

- This approach is only useful if you already have a Netbeans project open, and you don't care about its contents.
- Here we have some project called xyz, and we don't mind erasing its contents.



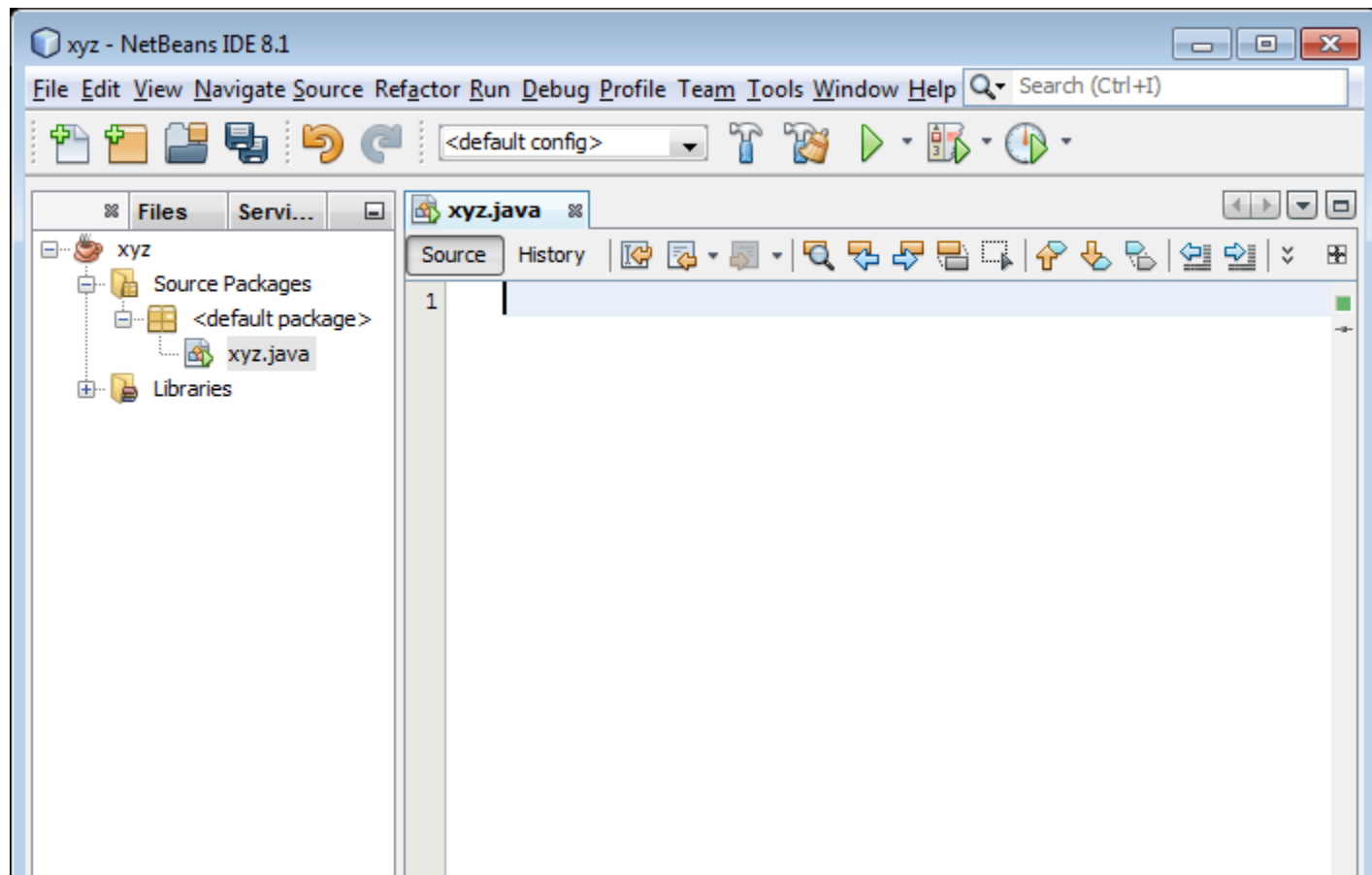
Running Existing Code – Another Way

- To erase all contents of xyz.java, click on Edit->Select All, and then press backspace.



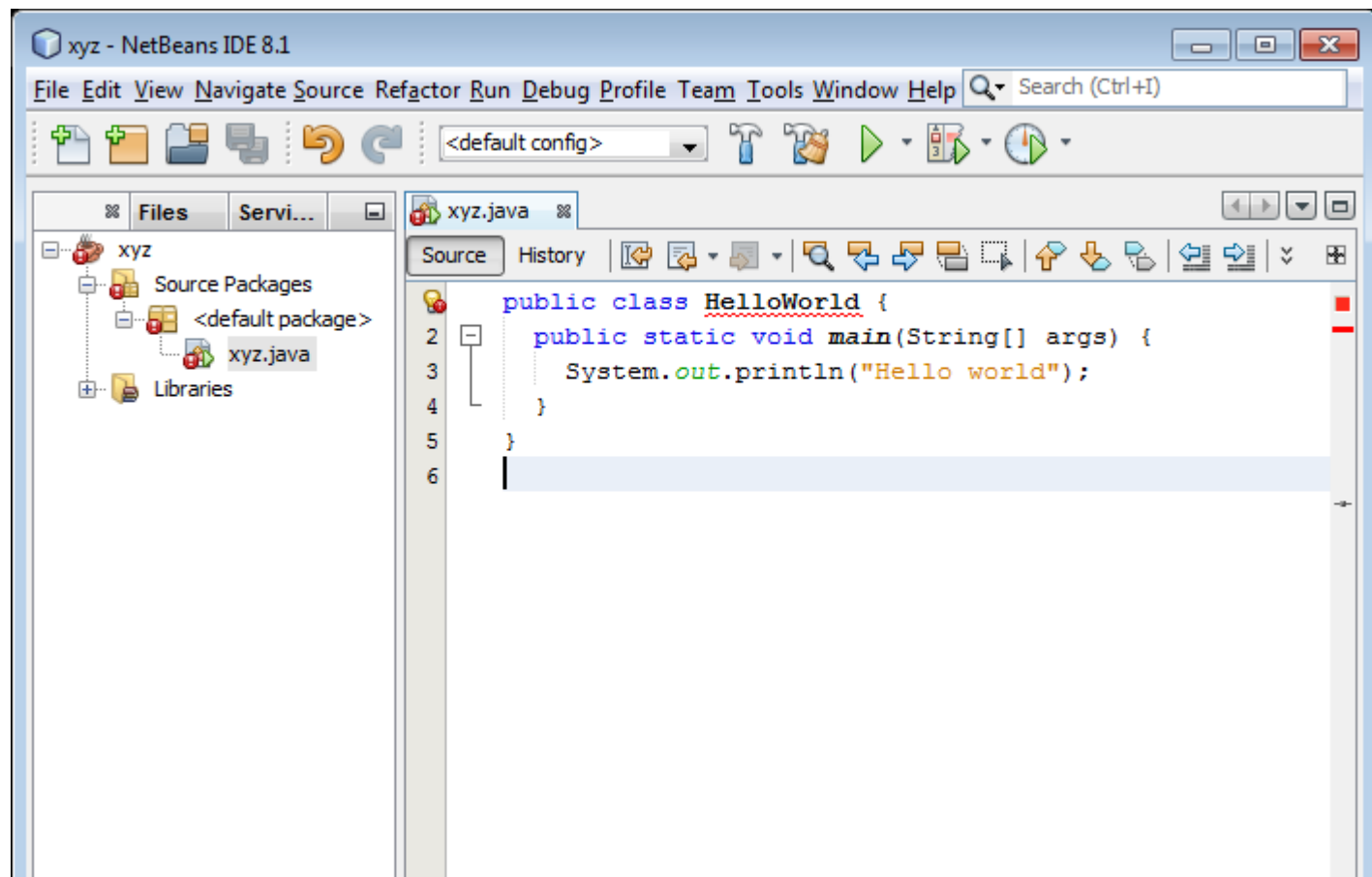
Running Existing Code – Another Way

- Now your xyz.java file is empty.



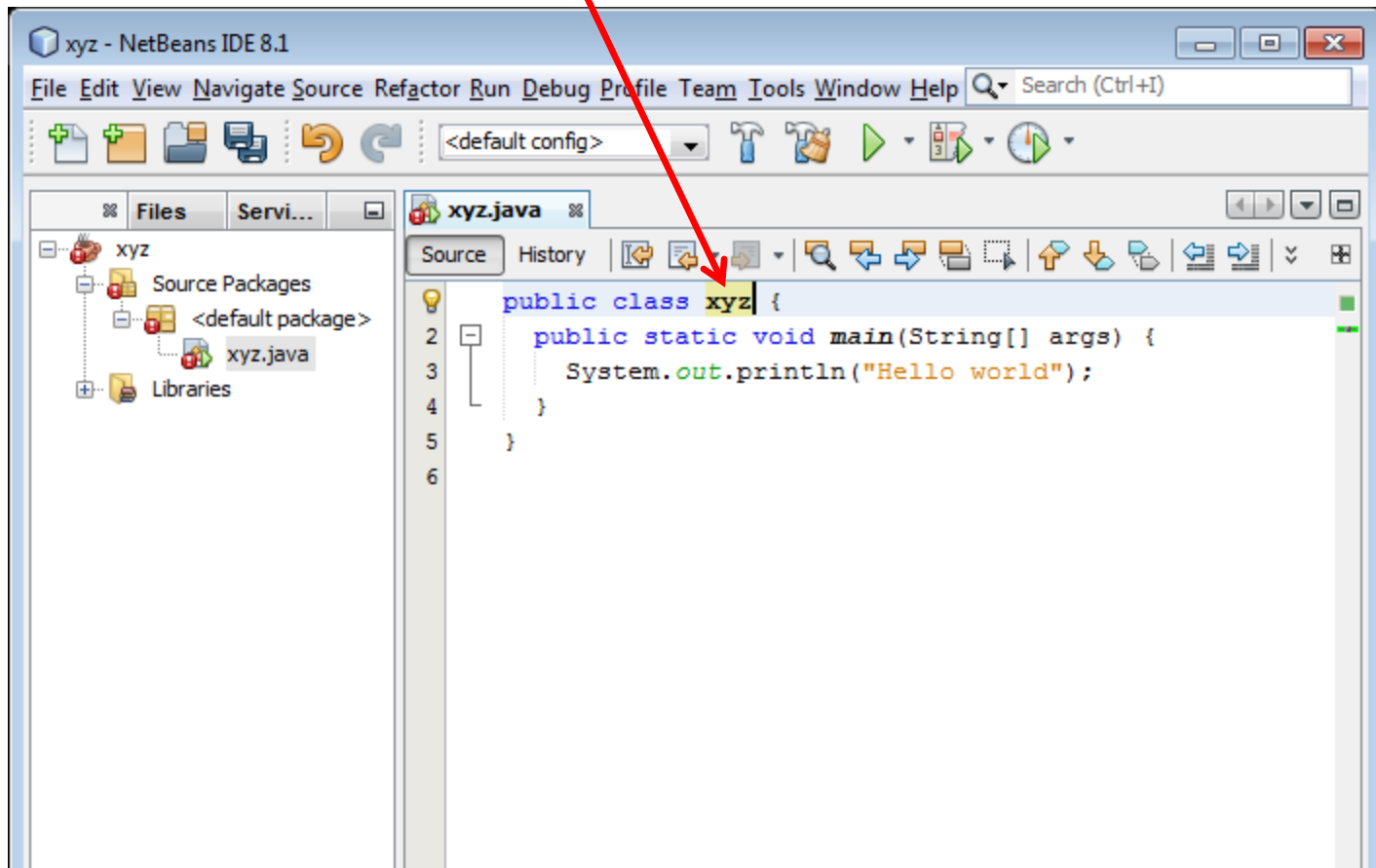
Running Existing Code – Another Way

- Next, copy and paste to xyz.java the code that you want to run.



Running Existing Code – Another Way

- Next, change the main class name to match the name of your Netbeans project (in this example, set the main class name to xyz).



Running Existing Code – Another Way

- Now you can run the code by pressing F6 (or by selecting Run->Run Project from the top menu), and see the output.

