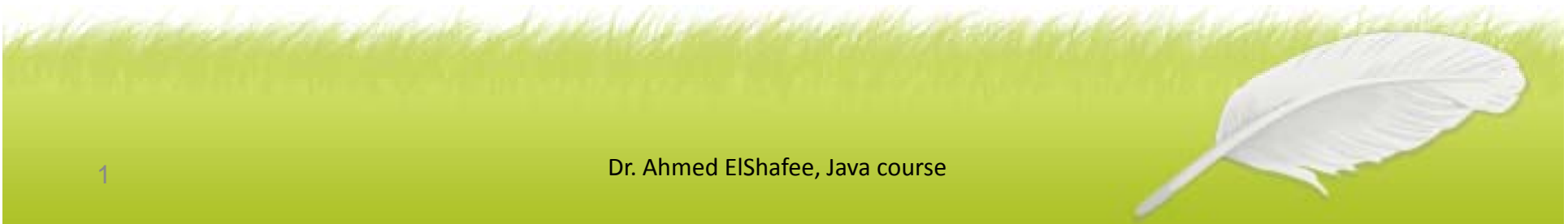




Lecture (01)

Getting started

Dr. Ahmed ElShafee



Agenda

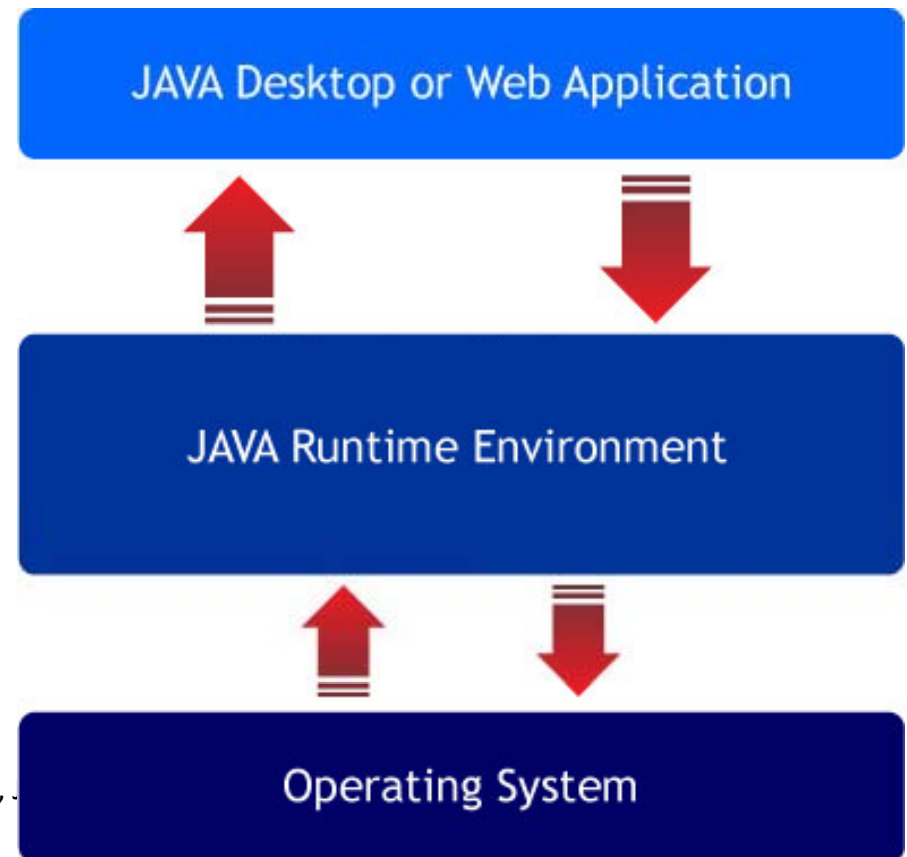
- Download and Installation
- Java – How things work

Download and Installation

Java components

1. The Java Virtual Machine or (Java Runtime Environment (JRE))

- Java is platform independent. This means that it will run on just about any operating system.
- That is done by Virtual Machine is a program that processes all your code correctly, run as an intermediate layer between your code and OS



- To download JRE use the following link
- <http://java.com/en/download/index.jsp>
- You can check to see if you already have the JRE on your computer by clicking the link "Do I have Java?".



The screenshot shows the Java website's download page. At the top, there is a red navigation bar with the Java logo on the left and a search box on the right. Below the navigation bar, the main content area is divided into two columns. The left column has a grey background and contains the text "All Java Downloads" followed by instructions on how to download Java for another computer or operating system, and a link to "All Java Downloads". The right column has a white background and features the heading "Free Java Download" in red. Below this heading, it says "Download Java for your desktop computer now!" and "Version 6 Update 26". A prominent red button with the text "Free Java Download" is centered in this column. Below the button, there are three links: "What is Java?", "Do I have Java?", and "Need Help?". At the bottom of the page, there is a section titled "Why download Java?" which explains that Java technology allows for a secure computing environment and provides examples of applications like online games, chat, and mortgage calculators. It also mentions that after downloading Java, users should visit java.com to check out "Java in Action".

- Or you can manually download and install it from
- <http://java.com/en/download/manual.jsp>

The screenshot shows the Java website's download page. At the top is a red navigation bar with the Java logo, a search box, and links for 'Java in Action', 'Downloads', and 'Help Center'. Below this is a sidebar titled 'Available Operating Systems' with links for Windows, Solaris, Linux, and Apple. The main content area is titled 'Java Downloads for All Operating Systems' and features a 'Recommended Version 6 Update 26' section. It includes instructions on how to select a file based on the operating system, links to 'Remove Older Versions' and 'What is Java?', and a disclaimer about the end user license agreement. A table lists two Windows download options: 'Windows 7, XP Online' (11 MB) and 'Windows 7, XP Offline' (16.0 MB), each with a download icon and a link to instructions. A 'Verify Now' link is also present. At the bottom, there is a link for 'Information about the 64-bit Java plug-in'.

Available Operating Systems

- » [Windows](#)
- » [Solaris](#)
- » [Linux](#)
- » [Apple](#)



Java Downloads for All Operating Systems

Recommended Version 6 Update 26

Select the file according to your operating system from the list below to get the latest Java for your computer.

> [Remove Older Versions](#) > [What is Java?](#)

By downloading Java you acknowledge that you have read and accepted the terms of the [end user license agreement](#)

Windows	Which should I choose?	
 Windows 7, XP Online filesize: ~ 11 MB	Instructions	Verify Now After installing Java, restart your browser and verify Java has been installed correctly.
 Windows 7, XP Offline filesize: 16.0 MB	Instructions	

[Information about the 64-bit Java plug-in](#)

2. The Java Software Development Kit

- To write code and test it out, you need something called a Software Development kit.
- <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- The one we're going to be using is called Java SE. (The SE stands for Standard Edition.).

- Java SE
- Java EE
- Java ME
- Java SE Support
- Java SE Advanced & Suite
- Java Embedded
- JavaFX
- Java DB
- Web Tier
- Java Card
- Java TV
- Community
- Java Magazine

- Overview
- Downloads**
- Documentation
- Community
- Technologies
- Training

Java SE Downloads

[Latest Release](#)

[Next Release \(Early Access\)](#)

[Embedded Use](#)

[Real-Time](#)

[Previous Releases](#)



Java Platform (JDK)



JDK + NetBeans Bundle



JDK + Java EE Bundle

Here are the Java SE downloads in detail:

Dr. Ahmed Elshater, ACC Spring 2011, Wireless Network

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [JavaFX](#)
- [Java Card](#)
- [NetBeans IDE](#)

Java Resources

- [New to Java?](#)
- [APIs](#)
- [Code Samples & Apps](#)
- [Developer Training](#)
- [Documentation](#)
- [Java BluePrints](#)
- [Java.com](#)
- [Java.net](#)

3. IDE (Integrated Development Environment)

- We're going to write all our code using a free piece of software called NetBeans.
- This is one of the most popular IDEs (Interface Development Environment) in the world for writing Java programmes.
- <http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html>
- Choose you OS, download then install

- Java SE
- Java EE
- Java ME
- Java SE Support
- Java SE Advanced & Suite
- Java Embedded
- JavaFX
- Java DB
- Web Tier
- Java Card
- Java TV
- Community
- Java Magazine

- Overview
- Downloads**
- Documentation
- Community
- Technologies
- Training

JDK 6 Update 26 with NetBeans 7.0

This distribution of the JDK includes the [NetBeans IDE](#), which is a powerful integrated development environment for developing applications on the Java platform. [Learn more](#)

You must accept the [JDK 6u26 and Netbeans 7.0 Cobundle License Agreement](#) to download this software.

- Accept License Agreement
 Decline License Agreement

Java SE and NetBeans Cobundle (JDK 6u26 and NB 7.0)

Product / File Description	File Size	Download
Linux	165.40 MB	jdk-6u26-nb-7_0-linux-ml.sh
Solaris SPARC	174.80 MB	jdk-6u26-nb-7_0-solaris-sparc-ml.sh
Solaris x86	163.96 MB	jdk-6u26-nb-7_0-solaris-x86-ml.sh
Windows	157.32 MB	jdk-6u26-nb-7_0-windows-ml.exe

- [License](#)

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [JavaFX](#)
- [Java Card](#)
- [NetBeans IDE](#)

Java Resources

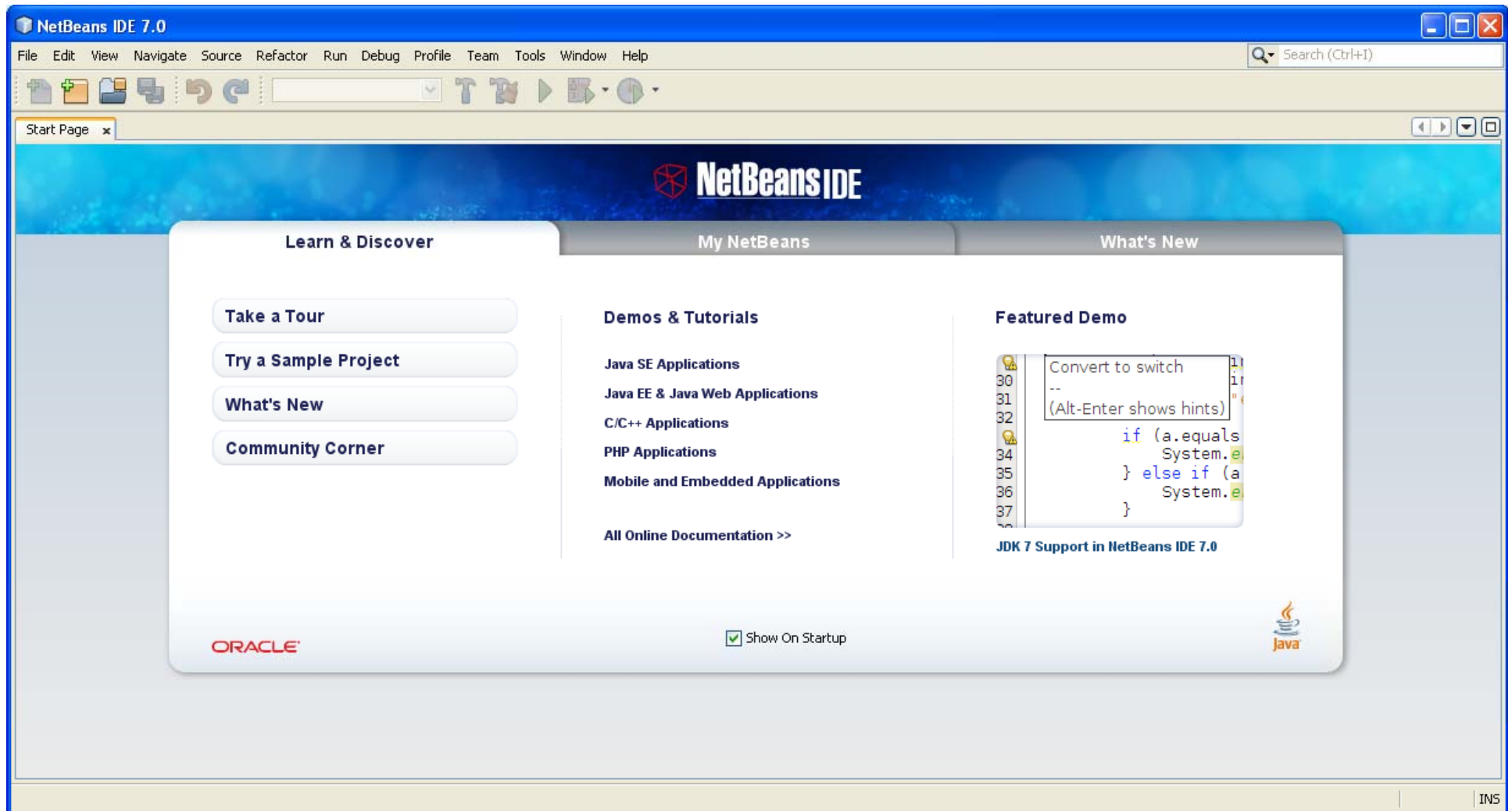
- [New to Java?](#)
- [APIs](#)
- [Code Samples & Apps](#)
- [Developer Training](#)
- [Documentation](#)
- [Java BluePrints](#)
- [Java.com](#)
- [Java.net](#)

Java – How things work

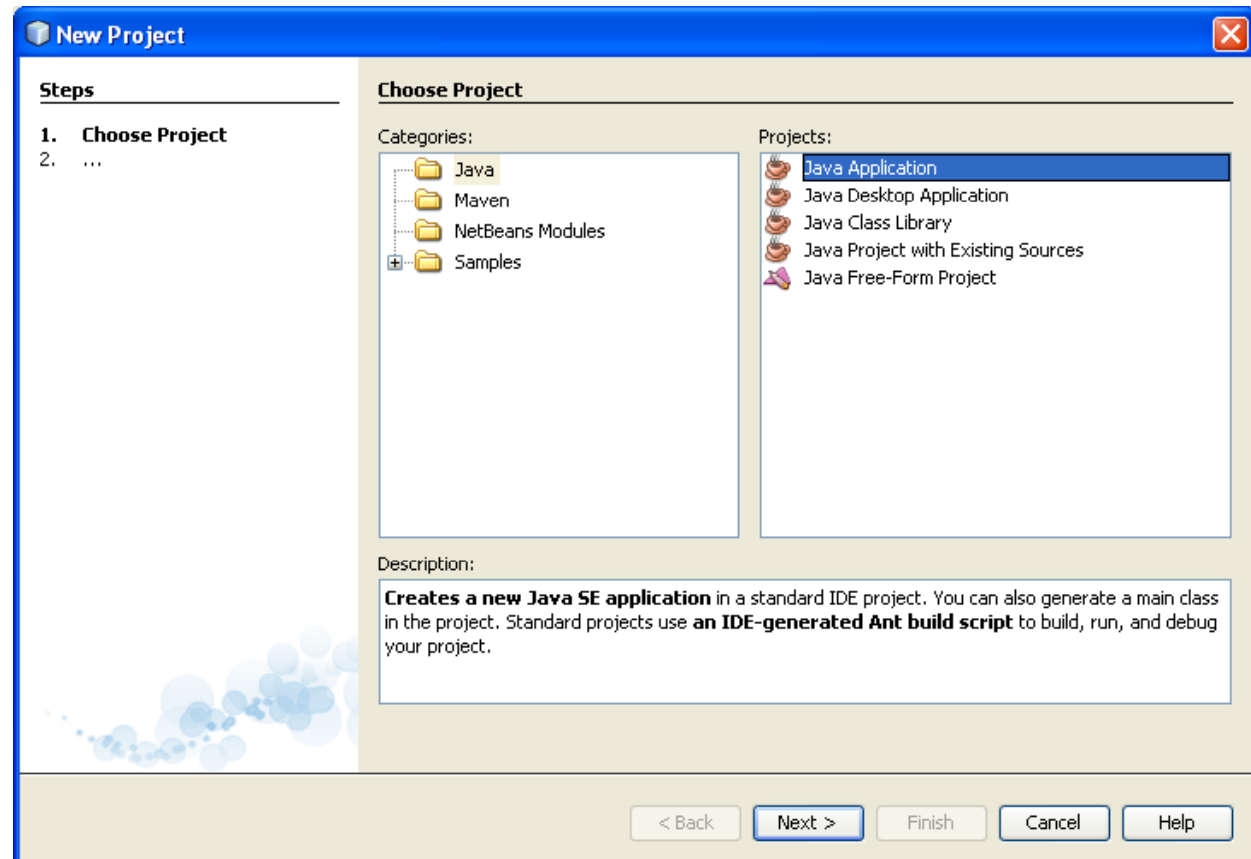
- Create source code with the extension .java (text Editor)
- Use Javac to create (compile) a file ending in .class (>Javac.exe)
- Run the compiled class (>java file.class)

- NetBeans handles all the creating and compiling for you. Behind the scenes, though, it takes your sources code and creates the java file.
- It will launch Javac and compile the class file. NetBeans can then run your programme inside its own software.
- This saves you the hassle of opening up a terminal window and typing long strings of commands,

NetBeans



- To start a new project, click on **File > New Project**. You'll see the following dialogue box appear:



- We're going to be creating a Java Application, so select **Java** under **Categories**, and then **Java Application** under **Projects**.
- **Click the Next button at the bottom to go to step two:**

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

Set as Main Project

< Back Next > **Finish** Cancel Help

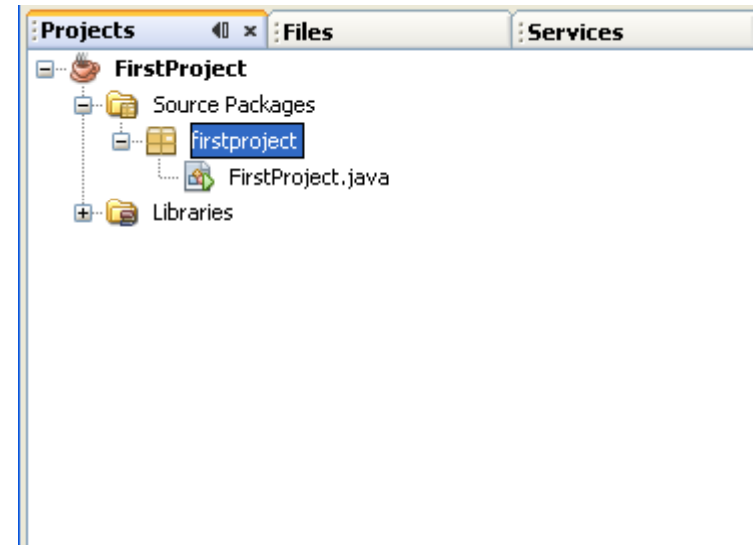
-
- In the Project Name area at the top, type a Name for your Project.
 - Notice how the text at the bottom changes to match your project name (in the text box to the right of **Create Main Class**): **firstproject.FirstProject**



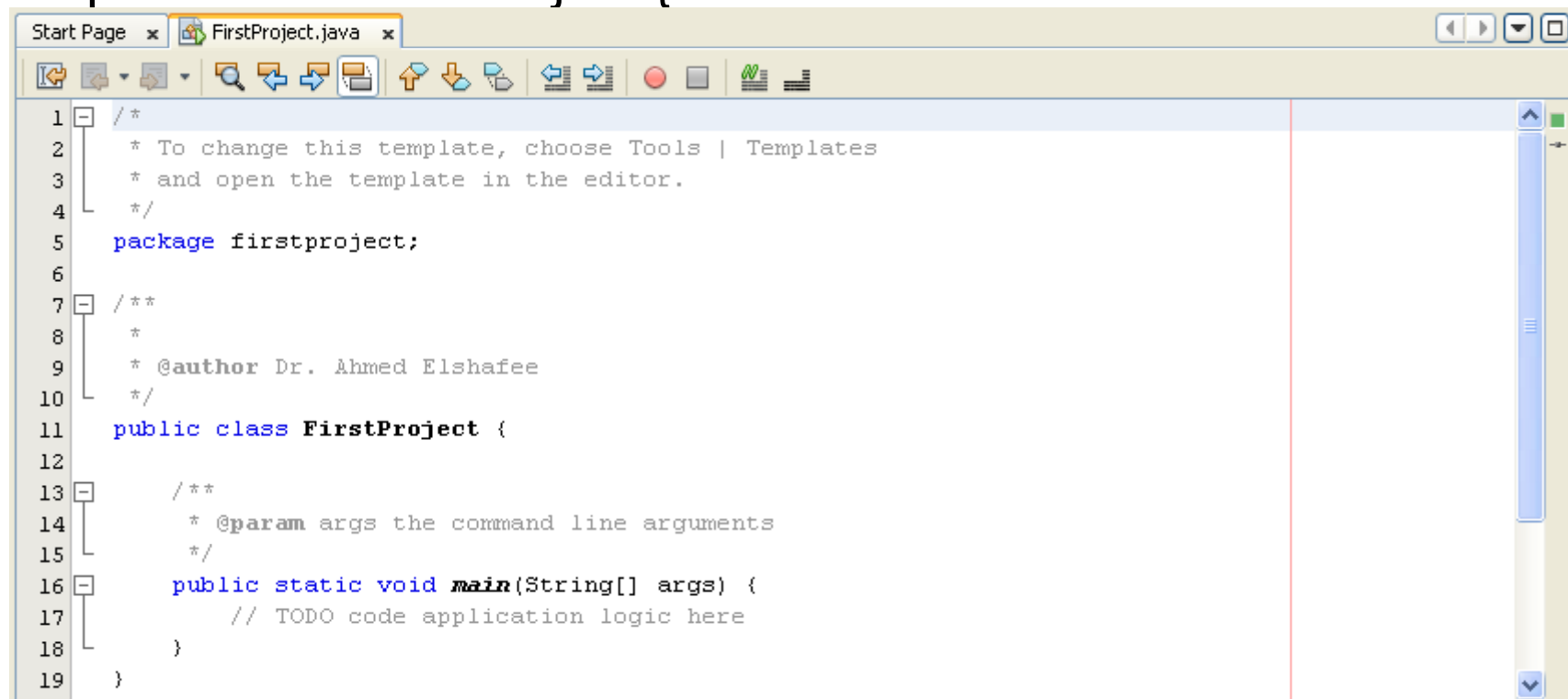
- Now, the Class created will be called FirstProject, with a capital “F”, capital “P”.
- The package is also called firstproject, but with a lowercase “f” and lowercase “j”.

-
- The default location to save your projects appears in the Project Location text box. You can change this, if you prefer. NetBeans will also create a folder with your project name, in the same location.
 - Click the **Finish button and NetBeans will go** to work creating all the necessary files for you.
 - When NetBeans returns you to the IDE, have a look at the Projects area in the top left of the screen (if you can't see this, click **Window > Projects from the menu** bar at the top of the software):

- Now expand Source Packages to see your project name again.
- Expand this and you'll see the Java file that is your source code.
- This same source code should be displayed to the right, in the large text area. It will be called **FirstProject.java**



- The coding window that appears should look like this (we've changed the author's name):
- One thing to note here is that the class is called FirstProject:
- `public class FirstProject {`



```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5  package firstproject;
6
7  /**
8   *
9   * @author Dr. Ahmed Elshafee
10  */
11  public class FirstProject {
12
13     /**
14      * @param args the command line arguments
15      */
16     public static void main(String[] args) {
17         // TODO code application logic here
18     }
19 }
```

-
- This is the same name as the java source file in the project window: **FirstProject.java**.
 - **When you run your programs, the compiler demands that the source file and the class name match.**
 - So if your **.java file is called firstProject** but the class is called FirstProject then you'll get an error on compile.
 - And all because the first one is lowercase "f" and the second one uppercase.

-
- Note that although we've also called the package **firsproject**, **this is not** necessary.
 - We could have called the package **firstpackage**.
 - **So the name of** the package doesn't have to be the same as the java source file, or the class in the source file: it's just the name of the java source file and the name of the class that must match.

Comments

- In the image above, you'll notice that some text is grayed out, with lots of slashes and asterisks. These are comments.
- You can have a single line comment by typing two slashes, followed by your comment:

//This is a single line comment

- If you want to have more than one line, you can either do this:

//This is a comment spreading

//over two lines or more

-
- Or you can do this:

*/**

***This is a comment spreading
over two lines or more***

**/*

- In the comment above, note how it starts with */**. To end the comment, we have **/* instead.

-
- There's also something called a Javadoc comment. You can see two of these in the coding image on the previous page. A Javadoc comment starts with a single forward slash and two asterisks (`/**`) and ends with an asterisk and one slash (`*/`).
 - Each line of the comment starts with one asterisk:

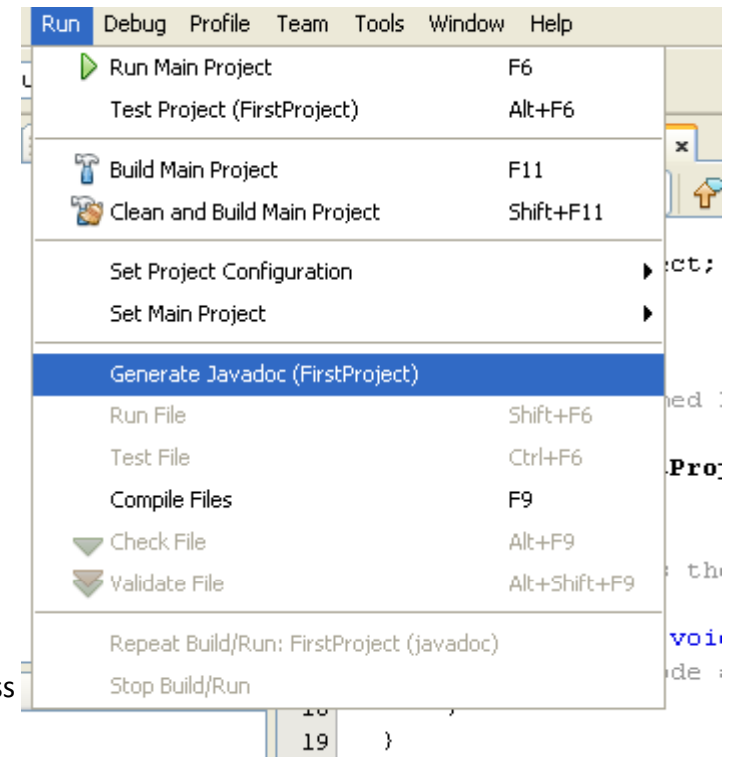
`/**`

****This is a Javadoc comment***

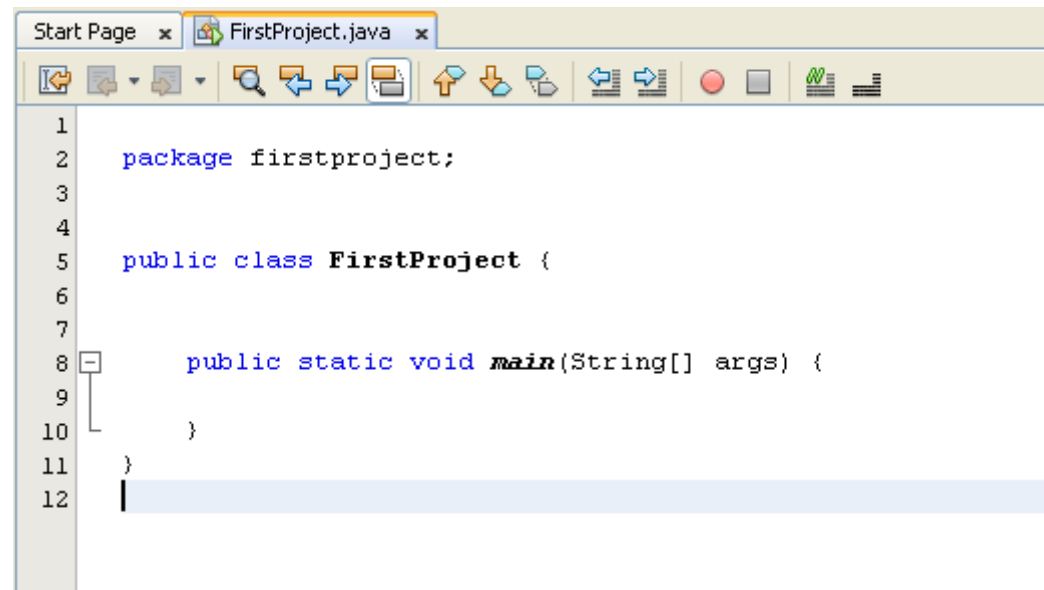
`*/`

- Javadoc comments are used to document code.
- The documented code can then be turned into an HTML page that will be helpful to others.

-
- You can see what these look like by clicking **Run from the menu at the top of NetBeans.**
 - **From the Run menu, select Generate Javadoc.**
 - **There's not much to see, however, as you haven't written any code yet!**



-
- At this stage of your programming career, you can delete the comments that NetBeans generates for you.
 - Here's our code again with the comments deleted: **Structure**



```
1
2 package firstproject;
3
4
5 public class FirstProject {
6
7
8     public static void main(String[] args) {
9
10    }
11 }
12
```


Structure of the program

- Now it looks a lot cleaner! You can see we have the package name first.
- Notice how the line ends with a semicolon. If you miss the semicolon out, the program won't compile:

package firstproject;

The class name comes next:

***public class FirstProject {
}***

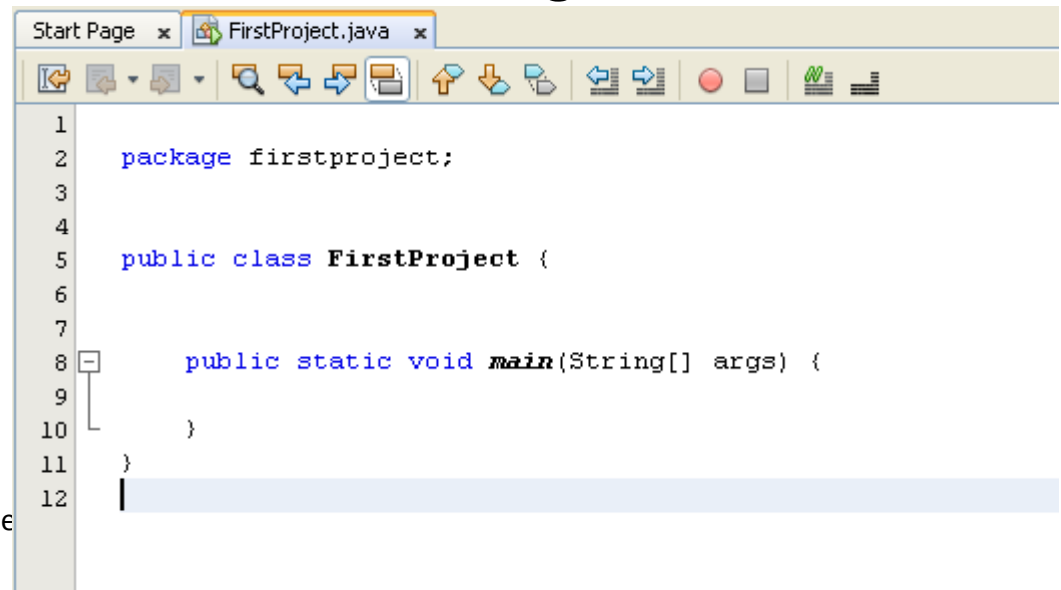
- You can think of a class as a code segment. But you have to tell Java where code segments start and end. You do this with curly brackets.

-
- The start of a code segment is done with a left curly bracket { **and is ended with a right curly bracket }**.
 - Anything inside of the left and right curly brackets belong to that code segment.
 - What's inside of the left and right curly brackets for the class is another code segment. This one:

```
public static void main( String[ ] args ) {  
    }  
}
```

- The word “main” is the important part here. Whenever a Java program starts, it looks for a method called **main**.

-
- It then executes any code within the curly brackets for main.
 - You'll get error messages if you don't have a main method in your Java programs.
 - But as its name suggest, it is the main entry point for your programs.
 - The blue parts before the word "main" can be ignored for now.
 - The parts between the round brackets of main are something called command line arguments (ignore for now).



```
Start Page x FirstProject.java x
package firstproject;

public class FirstProject {

    public static void main(String[] args) {

    }

}
```

Dr. Ahme

-
- The important point to remember is that we have a class called FirstProject.
 - This class contains a method called **main**. **The two have their own sets of curly brackets.**
 - But the **main chunk of code belongs to the class FirstProject.**

Writing a code

- So let's add one line of code just so that we can see how it works.
- We'll output some text to a console window. Add the following line to your **main** method:

```
public static void main(String[ ] args) {  
    System.out.println("My First Project");  
}
```

- When you type the full stop after “System”, NetBeans will try to help you by displaying a list of available options:

- Double click **out** to add it to your code, then type another full stop.
- Again, the list of options appears:
- Select **println()**.
- What this does is to print a line of text to the output screen.
- But you need to place your text between the round brackets of **println**.

```
public static void main(String[] args) {
```

```
System.|
)
err      PrintStream
in       InputStream
out      PrintStream
arraycopy(Object src, int srcPos, int length) void
clearProperty(String key) String
console() Console
currentTimeMillis() long
exit(int status) void
gc() void
getProperties() Properties
getProperty(String key) String
getProperty(String key, String def) String
getSecurityManager() SecurityManager
getenv() Map<String, String>
getenv(String name) String
identityHashCode(Object x) int
inheritedChannel() Channel
```

```
public static void main(String[] args) {
```

```
System.out.|
)
println() void
println(Object x) void
println(String x) void
println(boolean x) void
println(char x) void
println(char[] x) void
println(double x) void
println(float x) void
println(int x) void
println(long x) void
toString() String
wait() void
wait(long timeout) void
wait(long timeout, int nanos) void
write(byte[] b) void
write(int b) void
write(byte[] buf, int off, int len) void
```

-
- **Your text** needs to go between a pair of double quotes:

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

- Once you have your double quotes in place, type your text:
- Notice that the line ends in a semicolon.

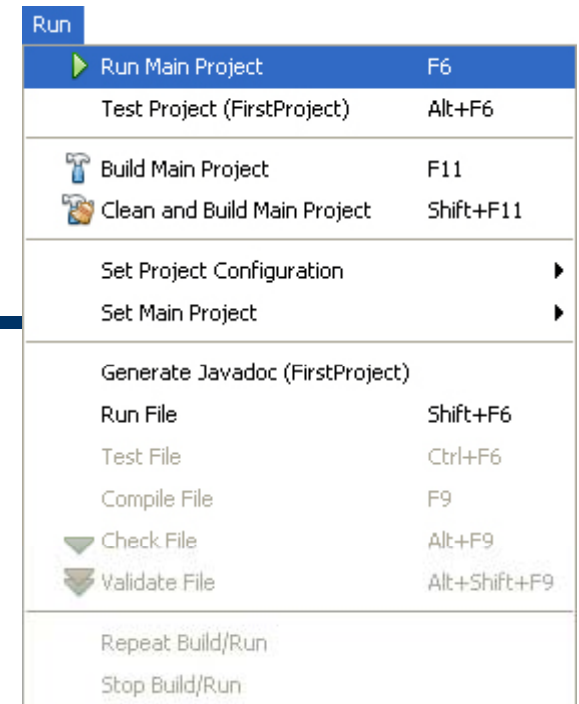
```
public static void main(String[] args) {  
    System.out.println("My First Project");  
}
```

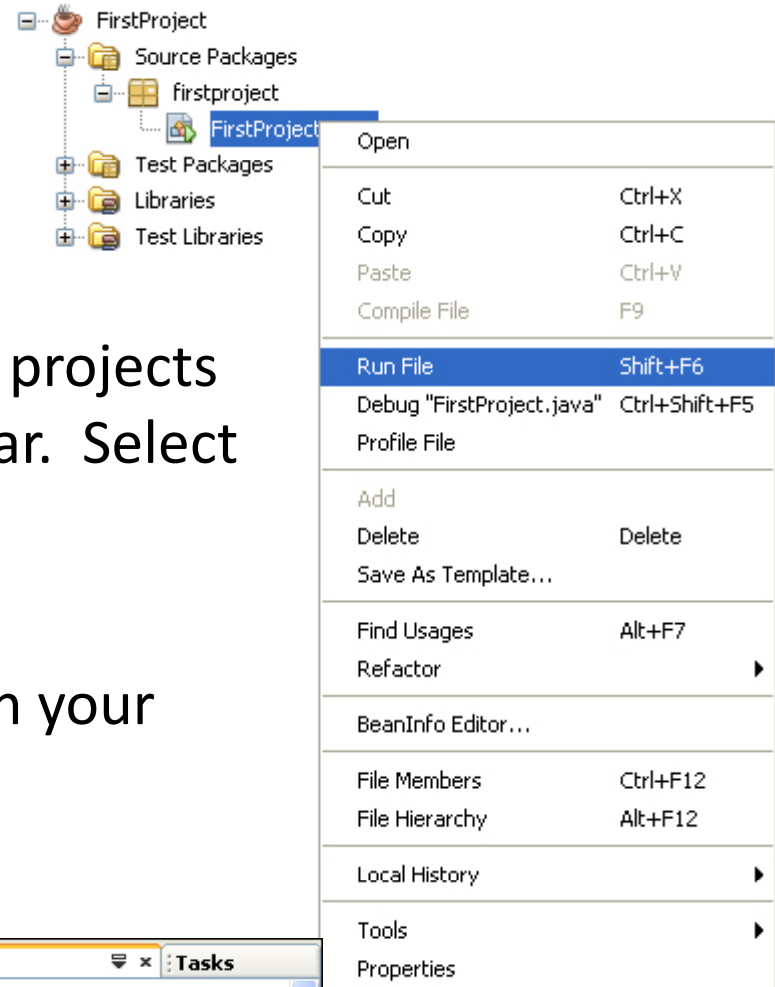
- Each complete line of code in Java needs a semicolon at the end.
- Miss it out and the program won't compile.

-
- OK, we can now go ahead and test this programme out. First, though, save your work.
 - You can click **File > Save, or File > Save All. Or click the Save icon on the NetBeans toolbar.**

Running a code

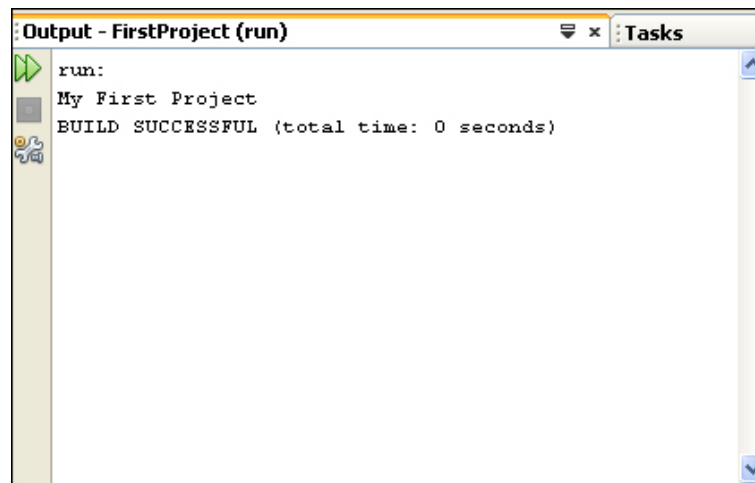
- When you run a program in NetBeans, it will run in the Output window at the bottom of your screen, just underneath your code.
- There are various ways to run your program in NetBeans.
 1. The easiest way is to press F6 on your Keyboard.
 2. using the menus as the top of NetBeans. Locate the Run menu, then select Run Main Program:
 3. You can also click the green arrow on the NetBeans toolbar:





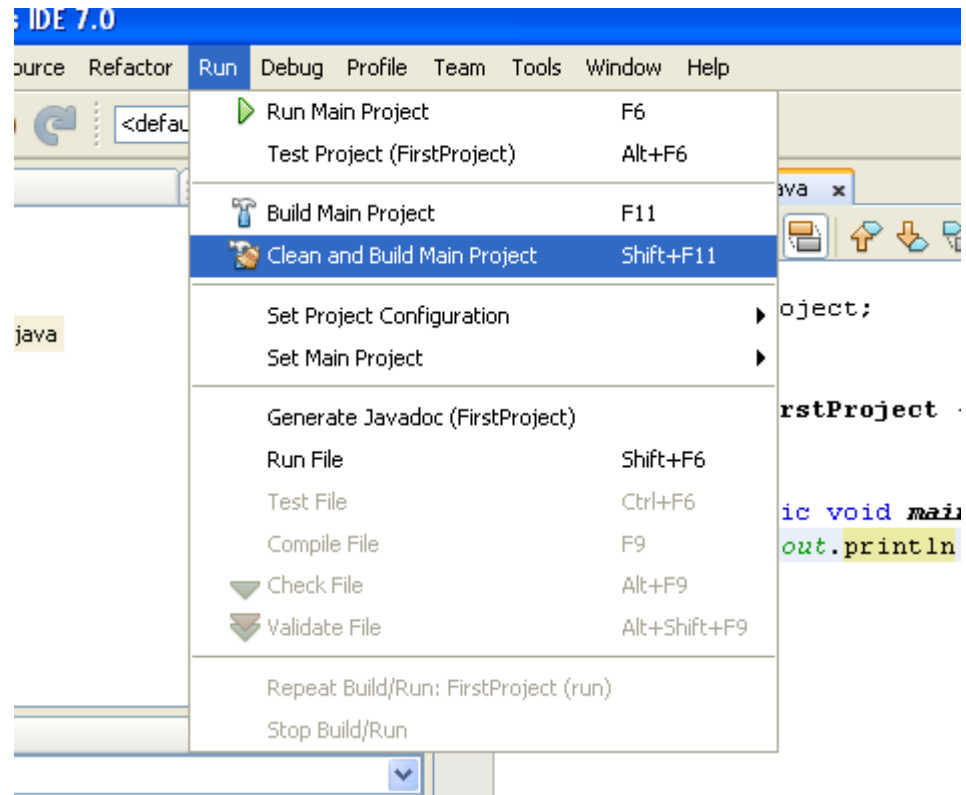
4. right click your java source file in the projects window and you'll see a menu appear. Select **Run File**.

- Using one of the above methods, run your program. You should see something happening in the Output window:

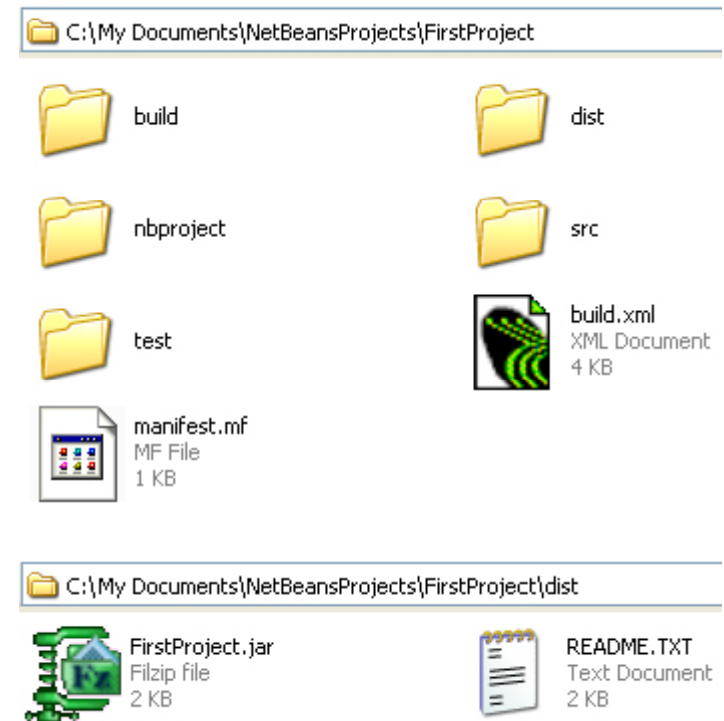


Sharing your programs with others

- You can send your programs to other people so that they can run them.
- To do that, you need to create a JAR file (Java Archive). NetBeans can do all this for you.
- From the Run menu at the top, select **Clean and Build Main Project**.



- When you do, NetBeans saves your work and then creates all the necessary files.
- It will create a folder called **dist** and place all the files in there.
- **Have a look in the** place where your NetBeans projects are and you'll see the **dist** folder:
- Double click the **dist** folder to see what's inside of it:



-
- You should see a JAR file and README text file. The text file contains instructions on how to run the program from a terminal/console window.

>java -jar FirstProject.jar

- Now that you know how to run your java source files, let's do some programming.



Thanks,
See you next Lecture, isA

