

How to install JAVA ?

Step-1: Download the package

Java SE 10.0.1 is the latest release for the Java SE Platform . Go to

<http://www.oracle.com/technetwork/java/javase/downloads/jdk10-downloads-4416644.html>

If you are downloading the JDK installer for 64-bit windows operating system then the file name is

jdk-10.0.1_windows-x64_bin.exe

Java SE Development Kit 10 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- [Java Developer Newsletter](#): From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- [Java Developer Day hands-on workshops \(free\) and other events](#)
- [Java Magazine](#)

[JDK 10.0.2 checksum](#)

Java SE Development Kit 10.0.2

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

☒ Accept License Agreement ☐ Decline License Agreement

Product / File Description	File Size	Download
Linux	306 MB	jdk-10.0.2_linux-x64_bin.rpm
Linux	338.43 MB	jdk-10.0.2_linux-x64_bin.tar.gz
macOS	395.46 MB	jdk-10.0.2_osx-x64_bin.dmg
Solaris SPARC	207.07 MB	jdk-10.0.2_solaris-sparcv9_bin.tar.gz
Windows	390.25 MB	jdk-10.0.2_windows-x64_bin.exe

Once you have obtained the .exe installation file, double-click it to begin the installation process. This process will lead you through the following series of windows:

- Setup → Click Next.
- Custom Setup → You do not need to make any changes to the default setting. Just verify the installation directory, which should be C:\Program Files\Java\jdk1.10.0 and Click Next.
- Progress → Wait for the next window to open.
- Destination Folder → You do not need to make any changes to the default setting. Just verify the installation directory, which should be C:\Program Files\Java\jre10\ → Click Next.
- Progress → Wait for process to end.
- Complete → Click Finish to complete.

How to install Eclipse ?

We will use the Eclipse Integrated Development Environment (IDE) to create, compile, and run Java programming assignments for this Lab.

Do not install Eclipse until you have installed Java.

Eclipse can be downloaded from the following website: <http://www.eclipse.org/downloads/eclipse-packages/>




You will be redirected to the mirror sites page. A mirror site should be pre-selected for you and appear as a link with a big download arrow:



Click to start downloading. Otherwise, you can use any of the mirror sites listed on the ' page.

Depending on your browser settings, you may be asked to confirm that where you want to save ' the file on your computer and/or choose where to save it. Make sure you know where the file is being downloaded.

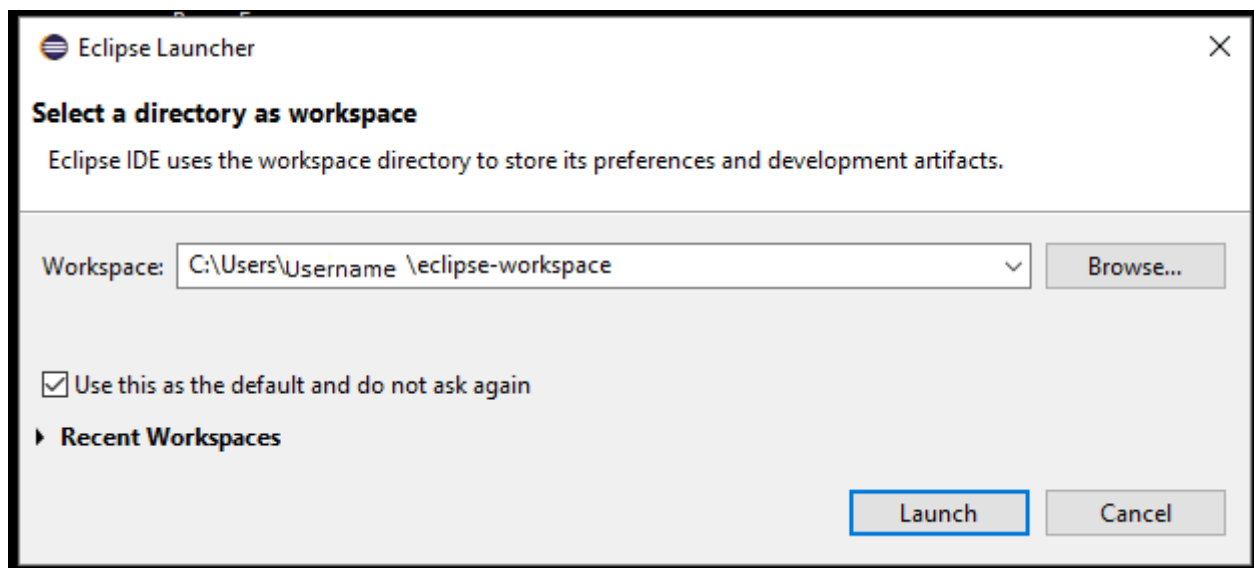
 eclipse-java-photon-R-win32-x86_64.zip

Once you have obtained the Eclipse .zip file, open it and choose Extract All to begin the extraction process. We recommend you move the extracted eclipse folder to C:\Program Files.

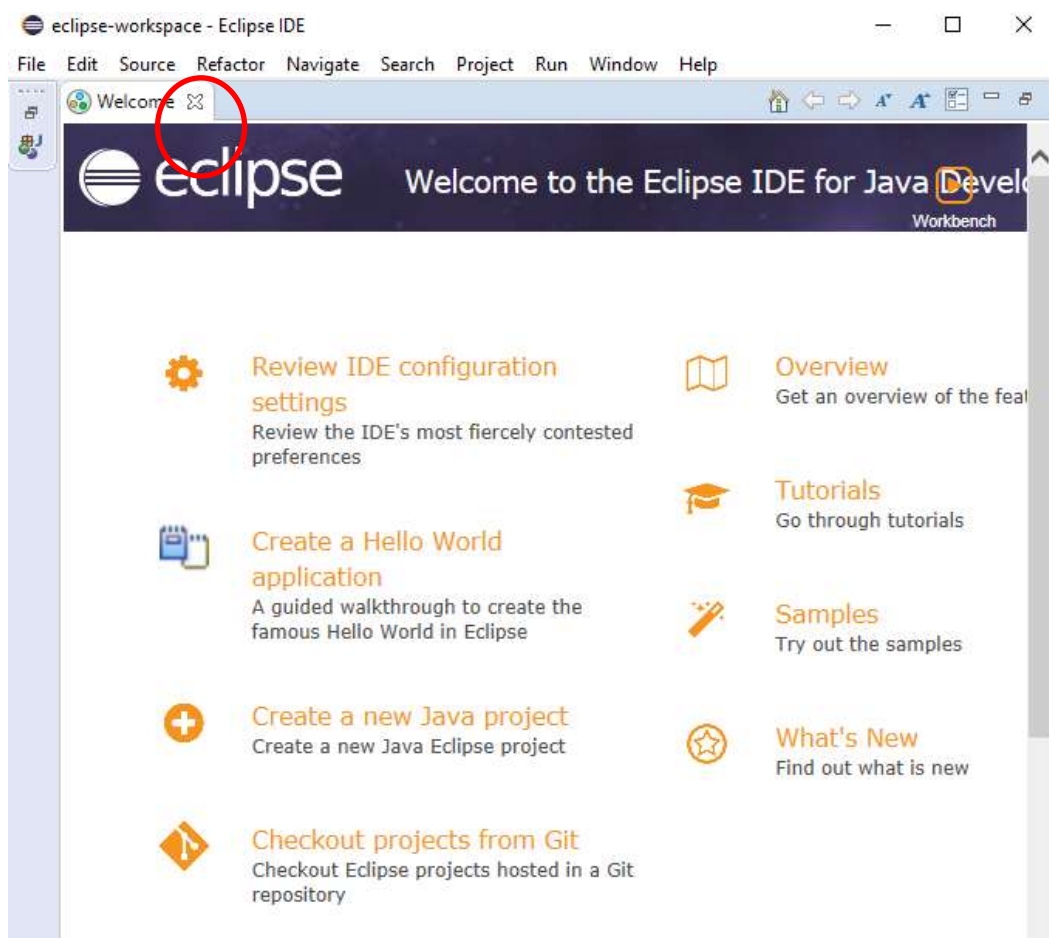
The extracted folder should then be C:\Program Files\eclipse.

Once you have extracted and moved the eclipse folder to its destination, you can create a shortcut to Eclipse. There are several ways to do this. In Windows, navigate to C:\Program Files\eclipse, right click on eclipse.exe, and select send to create shortcut.

Double-click on eclipse.exe or the Eclipse shortcut you created on your desktop. Each time you run Eclipse, you will be prompted to select a workspace, which is where the programs you write are stored. The default workspace is C:\Users\<username>\Eclipse\workspace



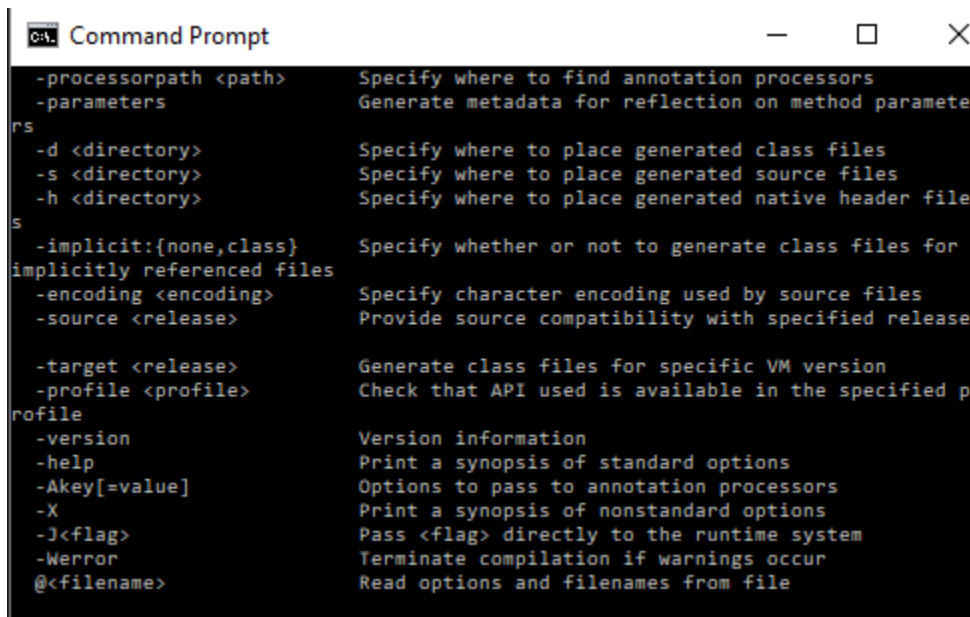
Click Use this as the default and do not ask again to avoid having to type in the workspace path every time you start.



Cross to close the welcome window to access eclipse IDE.

How can you check if JAVA has been installed properly or not.

Open the command prompt and type JAVAC then click enter, you should see messages as below scrolling up.



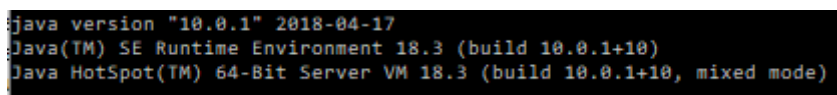
```
Command Prompt
-processorpath <path>      Specify where to find annotation processors
-parameters                Generate metadata for reflection on method parameters
-d <directory>            Specify where to place generated class files
-s <directory>            Specify where to place generated source files
-h <directory>            Specify where to place generated native header files
-implicit:{none,class}     Specify whether or not to generate class files for implicitly referenced files
-encoding <encoding>       Specify character encoding used by source files
-source <release>          Provide source compatibility with specified release
-target <release>          Generate class files for specific VM version
-profile <profile>          Check that API used is available in the specified profile
-version                   Version information
-help                     Print a synopsis of standard options
-Akey[=value]              Options to pass to annotation processors
-X                         Print a synopsis of nonstandard options
-J<flag>                   Pass <flag> directly to the runtime system
-Werror                    Terminate compilation if warnings occur
@<filename>               Read options and filenames from file
```

If "java' is not recognized as an internal or external command" message pops up you need to set the path manually.

Path Setting:

You need to configure your environment variables, JAVA_HOME and PATH.

1. Right click on My Computer
2. Select Properties
3. Select Advanced System Settings
4. Select Advanced tab
5. Select Environment Variables
6. Select Path under System Variables
7. Click on Edit button
8. In Variable value editor paste this at the start of the line
C:\Program Files\Java\jdk-10.0.1\bin;
9. Click Ok then Ok again
10. Restart command prompt otherwise it won't see the change to the path variable
11. Type java -version in command prompt.



```
java version "10.0.1" 2018-04-17
Java(TM) SE Runtime Environment 18.3 (build 10.0.1+10)
Java HotSpot(TM) 64-Bit Server VM 18.3 (build 10.0.1+10, mixed mode)
```

Program-1

1 a) **Installation of JAVA and eclipse: refer to**

<https://www.youtube.com/watch?v=NZ38tIhtQgw>

Executing JAVA program from command prompt:

Step-1

Type your code from a text editor like notepad or WordPad.

In this example, we'll use Notepad, a simple editor included with the Windows platforms. You can easily adapt these instructions if you use a different text editor.

To find Notepad on Windows:

1. Click on **Start** in the taskbar at the bottom of the desktop
2. Click on **Programs** or **All Programs**.
3. Click on the **Accessories** folder to expand it.
4. Click on the **Notepad** icon to open the program.

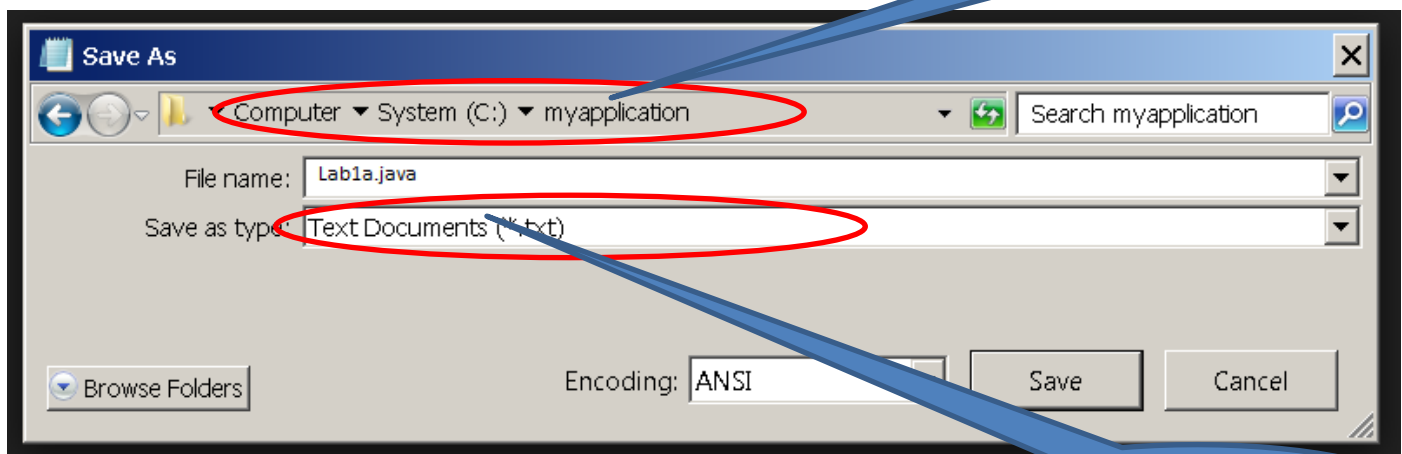


Type the below code:

```
// Your First Program

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

Let's save your file as in C:\user\username\desktop\mywork\Lab1a.java



Step-2 Compile the program

You will use the `javac` command to convert your Java program into a form more amenable for execution on a computer.

- From the Command Prompt, navigate to the directory containing your `.java` files, say `C:\user\username\desktop\mywork` , by typing the `cd` command below.
- `C:\Users\username>cd C:\user\username\desktop\mywork`
- `C:\user\username\desktop\mywork \>`
- Assuming the file, say `Lab1a.java`, is in the current working directory, type the `javac` command in boldface below to compile it.
- `C:\user\username\desktop\mywork >javac Lab1a.java`
- `C:\user\username\desktop\mywork >`

If everything went well, you should see no error messages.

You will use the `java` command to execute your program.

Step-3 Executing the program

- From the Command Prompt, type the `java` command below.
- `C:\user\username\desktop\mywork >Lab1a`
Hello, World

If all goes well, you should see the output of the program - Hello, World.

Troubleshooting:

If your program gets stuck in an infinite loop, type `Ctrl-C` to break out.

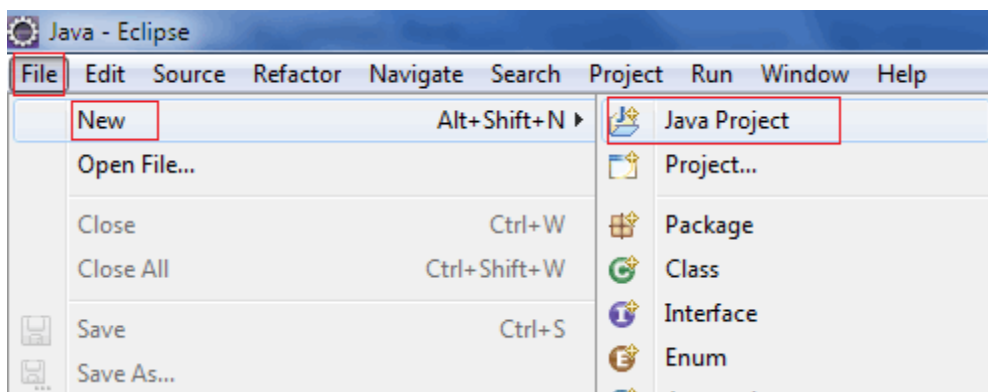
If you are entering input from the keyboard, you can signify to your program that there is no more data by typing `Ctrl-Z` for EOF (end of file). On some DOS systems the first line of output sent to the screen after you enter EOF will be rendered invisible by DOS. This is not a problem with your code, but rather a problem with DOS. To help you debug your program, we recommend including an extra `System.out.println();` statement before what you really want to print out. If anyone knows of a better fix, please let us know!

Executing “HelloWorld” JAVA program from eclipse IDE

1. Create a New Package Lab1

Because we're using Eclipse, we need to create a Java project before we can begin coding. A Java project is just a bunch of Java files and other files needed for your program to work. Every time you want to work on something new, I recommend creating a new project for it, as it's a good habit to have. So, if you haven't already, go start up Eclipse, and hit ok when the Workspace Launcher box appears.

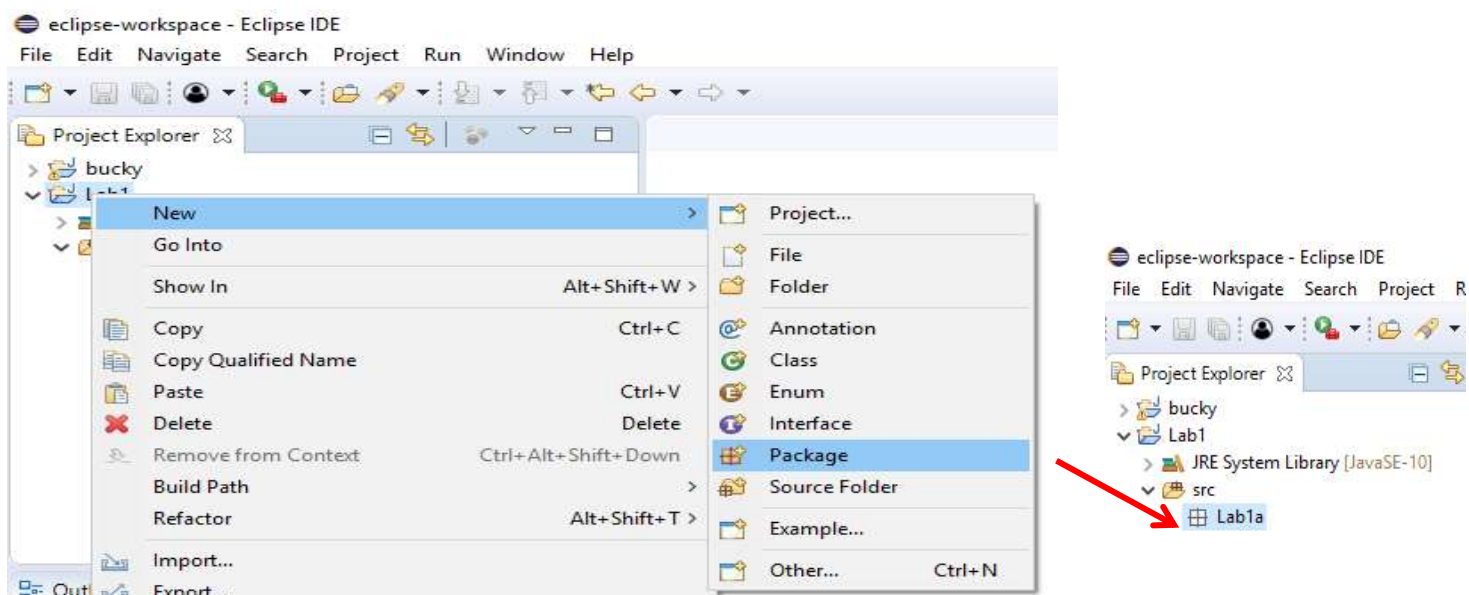
Right click on the left side panel, go to new, and select Java Project. (If you don't see that immediately, just hit Project... and you'll see more project types. Java Project should be at the very top, but if not, go to the folder that says Java, and you should see Java Project in there.)



Great! Now we have a bunch of options to set for our new Java Hello World project. All of these options can quickly become overwhelming, so once again we won't go into any details. Name your new project as Lab1, leave everything else how it was, and then hit finish.

The form will close and you should see your new project for your Java Lab1 on the side panel.

2. Create a New Package Lab1a



3. Create a New Class helloWorld

If you expand your new project by clicking the plus sign next to the folder, you'll see a folder called `src` and also a JRE System Library. Again, don't worry about these things. Right click on the `src` folder, go to new, and select Class. All Java files are the same as class files, so that's usually the only option you'll ever select.

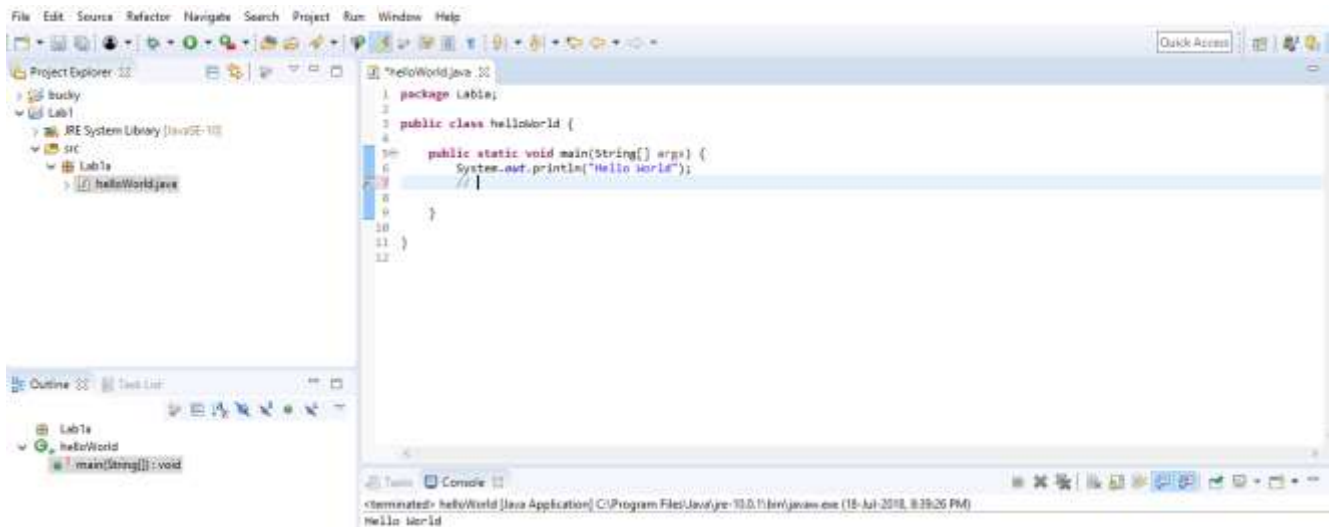
Now you have yet another form to fill. That's ok; just give this new file a name. We're going to call it `helloWorld`, so type in `helloWorld` (no spaces!) where it asks for the name. Java files should always start with a capital letter as the first letter and should have no spaces, so remember that. Also, check the box that says `public static void main(String[] args)`, because you'll need that. Then hit finish.



4. Writing the Java Hello World Program

Now you'll see your newly created file, and you'll see that some of the work has been done for you. Let's take a quick look at what it is you're seeing.

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

public class helloWorld is the beginning of your class file. Notice it has the same name as the file itself. The name, including the capitalized letters, must match the file name in Java or it won't work. Eclipse does this work for you of course. helloWorld is not the same as HelloWorld, because they are not exactly the same. The second HelloWorld has a capitalized H, the first one does not.

public static void main(String[] args) is called the main method. I'm not going to discuss what methods are right now, but without this statement, you cannot run your program. So remember that!

You'll notice two open brackets like this one in front of two of the lines of code:

```
{
```

and then further down you see two closing brackets that looks like:

```
}
```

Anything you write inside the inner two brackets belong to the main method, and anything inside of the outer brackets belongs to the class. All methods and classes have opening and closing brackets, but we'll touch on that some other time.

5. Console Output

Your code will print 'Hello World' in the eclipse console.

Write code to enter values from keyboard.

Write codes to learn about arithmetic, logic, conditional, math operators and expressions.

Chapter 3,4 & 5 of E.Balagurusamy