

Introduction to Java: Eclipse Setup Guide

Step-by-step instructions for installing JDK and Eclipse, creating a project, and using the built-in terminal . Note - the versions may differ - as long as your JDK works!!

1. Install the Java Development Kit (JDK)

Eclipse runs on the Java platform, so you need a **Java Development Kit (JDK)** before installing the IDE. A JDK **includes** the runtime (JRE) plus tools and source code.

1. **Download the JDK** – Visit the [Eclipse Temurin downloads page](#) and select the latest long-term support installation, for example **JDK 21 (LTS)** for **Windows x64** (choose the `.msi` installer for easiest installation).
2. **Run the installer** – After the download completes, double-click the `.msi` file. Accept the license agreement and use the default installation path unless you have a specific reason to change it.
3. **Verify the installation** – Open a Command Prompt (press `Win + R`, type `cmd`, press **Enter**) and run `java -version`. You should see the current version information.

💡 *Oracle vs Temurin distribution: If you need a completely free, production-ready JDK with multi-year LTS updates, Temurin is generally the go-to choice.*

If you require Oracle's commercial support guarantees (SLA), or depend on their specific certification and branding, Oracle JDK with a support subscription may be preferable.

2. Install the Eclipse IDE

1. **Download the installer** – Go to the official [Eclipse Installer download page](#) and choose the **Windows 64-bit** installer.
2. **Run the installer** – When the installer opens, pick “**Eclipse IDE for Java Developers**” from the list of packages (this package includes everything you need for Java programming).



Figure 1. The Eclipse Installer lists available packages; choose “Eclipse IDE for Java Developers.”

3. **Choose an installation folder** – The default location is usually fine. Click **Install** to begin. The installer will download and install the IDE automatically.

4. Select your Eclipse workspace

On first launch, Eclipse will ask for a **workspace directory**. This is where Eclipse stores its own settings, metadata, and preferences.

Always choose a folder that **only** holds these Eclipse files!

Choose a location (e.g., C:\Users\YourName\eclipse-workspace) and click **Launch**

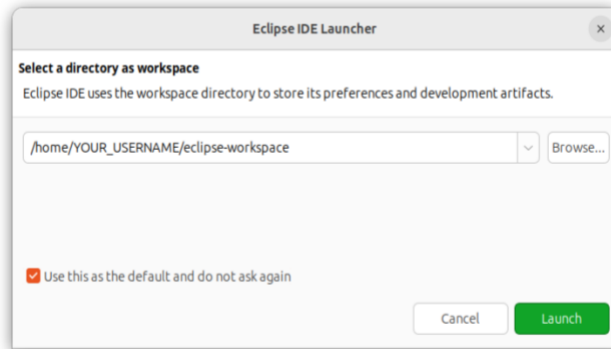



Figure 2. Eclipse prompts for a workspace folder on first launch.

 *Tip:* Check the “Use this as the default and do not ask again” box if you don’t want to choose the workspace every time.

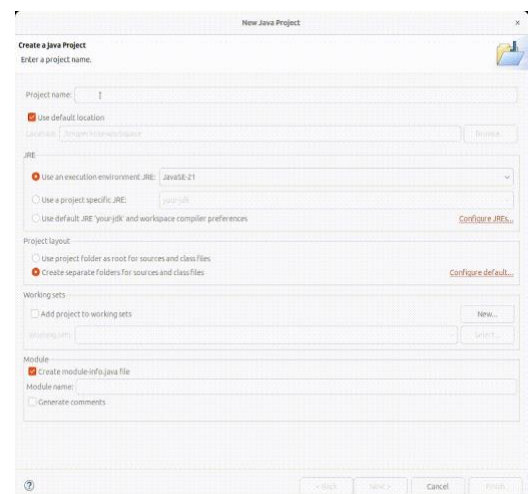
3. Create a Java Project, Package, and Class

Time to create your first project! 😊

Before you begin: Create a top-level **course folder** on your drive named, for example, CSSE220. Inside it, make a **subfolder** for this practice, **intro**.

Now you are ready:

1. In Eclipse, from the top menu, go to **File** → **New** → **Java Project**.
2. The New Java Project wizard opens. Enter a project name: HelloWorld
3. **Uncheck Use default location**
4. Click **Browse...** and navigate to your course subfolder
5. Uncheck *Create module-info.java* (modules aren’t needed for intro projects).
6. Click **Finish**.

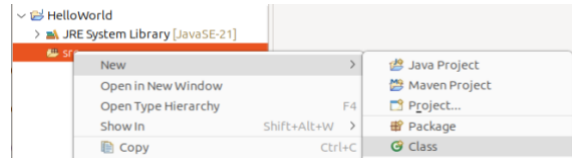


Create a package

In the *Project Explorer* on the left, right-click the `src` folder and select **New** → **Package**. Enter a package name, such as `edu.yourname.intro`, and click **Finish**. Packages organize classes.

Create a class

Right-click the package you just created and choose **New** → **Class**. Name the class (e.g., HelloWorld). Check the box to generate a `public static void main(String[] args)` method and click **Finish**.



When you click **Finish**, Eclipse will create your class and open it in the editor. Type a simple `System.out.println("Hello, world!");` inside the main method, save the file, and click the **Run** button (the green play icon) to see your program output in the console.

```
1 package introduction;
2
3 public class HelloWorld {
4     public static void main(String[] args) {
5         System.out.println("Hello, World");
6     }
7 }
```

Lost or stuck? 😞 No worries!

1. In Eclipse's top menu, click **Help** 📖
 2. Choose **Window** → **Tutorials** 🛠️
 3. Select the **Hello World** tutorial and work through the steps 🧑🏫 🧑🏫
- This guided walkthrough will get you back on track in no time! 🚀

4. Enable the Terminal in Eclipse (Optional)

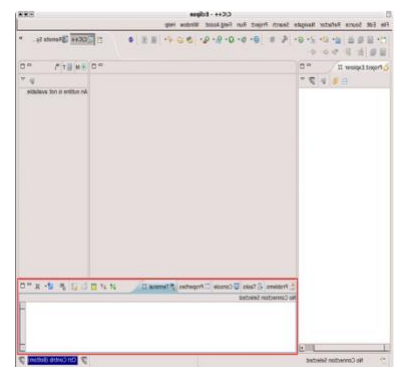
Eclipse includes a handy **Terminal view** (though it may not be enabled by default). This view lets you run command-line tools without leaving the IDE.

1. Show the Terminal view 🖥️

- Go to **Window** → **Show View**.
- If you see **Terminal**, select it.
- Otherwise, click **Other...**, expand **Terminal**, pick **Terminal**, and click **OK**.

2. Install TM Terminal (if needed) 🐘

- If no Terminal view appears, open **Help** → **Eclipse Marketplace**, search for **TM Terminal**, click **Install**, then restart Eclipse.
- After restarting, repeat **Step 1** to open the Terminal.



5. Using Terminal Commands (cd, pwd, ls)

Using the Terminal on Windows 🌀

1. Show the Terminal view

- Go to **Window** → **Show View**.
- If you see **Terminal**, select it.
- Otherwise, click **Other...**, expand **Terminal**, pick **Terminal**, and click **OK**.

2. Install TM Terminal (if needed)

- If no Terminal view appears, open **Help** → **Eclipse Marketplace**, search for **TM Terminal**, click **Install**, then restart Eclipse.
- After restarting, repeat **Step 1** to open the Terminal.

Using the Terminal on Windows


Once the view is open, you're likely dropped into PowerShell or the Windows Command Prompt:

macOS/Linux command	Windows equivalent
Change directory <code>cd <folder></code>	<code>cd <folder></code> (same)
View current directory <code>pwd</code>	<code>pwd</code> in PowerShell or <code>cd</code> in Command Prompt
List files <code>ls</code>	<code>ls</code> in PowerShell or <code>dir</code> in Command Prompt

Tip:

- PowerShell supports many Unix-style commands (`pwd`, `ls`), but in plain Command Prompt use `dir` to list files and `cd` without arguments to show your current path.
- You can switch the default shell in the Terminal view's dropdown if you prefer one over the other.

6. You're Ready to Code!

Congratulations  – you've set up your Java development environment! You installed the JDK and Eclipse, created your first project, package and class, enabled the Terminal, and learned some basic command-line navigation. Have fun and keep learning! 