

# Download, Install, and Use Required Software: JDK and Eclipse

## Table of Contents

Download, Install, and Use Required Software: JDK and Eclipse.....	1
Download the Do-It-Yourself Java Games ZIP File.....	1
Create a Work Folder.....	1
Download the ZIP file.....	1
Download and Install Java.....	2
Which JDK should you use?.....	2
Download and Install Oracle JDK.....	3
Download and Install OpenJDK.....	3
Download and Install Eclipse.....	4
Set up Eclipse.....	5
Create a Java Program.....	7
Add a Project to the Build Path.....	12
Export a Program.....	13
Run Your Program Outside of Eclipse.....	14
Tips For Running the Exported Program:.....	15
Create Market-Ready Java Programs.....	15

---

## ***Download the Do-It-Yourself Java Games ZIP File***

---

Create your own Java work folder on your computer. It's recommended that you put your Java work folder in your Windows user folder, but you can put it anywhere on your computer. I called mine "java" and put it in my user folder, C:\Users\Annette. Therefore, my Java work folder is C:\Users\Annette\java.

### **Create a Work Folder**

To create a folder in Windows 11, first click Start > File Explorer. Then go to the folder in which you want to create the new folder, and click the New folder button. Enter a name for your new folder.

If you share your computer with someone else who also wants to do java programming, create a different Java work folder for each of you. Your Java work folders must either have different names or be in different folders.

1. Create a Java work folder on your computer now.

### **Download the ZIP file**

The lessons require a few additional data files. Those files are included in a free .zip file on the Godtland Software website:

1. Download the Do-It-Yourself Java Games ZIP file from

<http://www.godtlandsoftware.com/DIYJava/>

2. To unzip the ZIP file, right-click on it and choose Extract all.... Use the Browse button to extract the files into your Java work folder. I extracted mine to  
C:/Users/Annette/java

You've now installed everything you need for creating the programs in the Do-It-Yourself Java Games books.

---

## ***Download and Install Java***

---

You'll be writing your computer programs in the Java programming language. Your computer needs a Java Development Kit to generate programs from Java program code.

The Java Development Kit only needs to be installed one time on your computer, no matter how many people plan to do Java programming on it.

### **Which JDK should you use?**

I used to recommend getting Java from Oracle. However, Oracle recently changed its licensing of the Java Development Kit (JDK). Though you can still download the JDK from Oracle for free and use it for Java program development and testing purposes, you must pay Oracle a subscription fee if you want to deploy such a program for commercial use. But Oracle also provides the same Java code in OpenJDK. OpenJDK is free and comes with the GNU General Public License (GPL) so may be freely used to develop Java programs for commercial use. But OpenJDK comes with no Oracle support.

Whereas subscribers to Oracle JDK will have access to updated older releases of the JDK, users of OpenJDK may only access JDK builds for 6 months after they are released. According Dzone.com, OpenJDK may be sufficient for you:

The JKDs: Which One to Use?

If you only use the default Java features and do not rely heavily on security updates for your business, you should be fine using OpenJDK and manually migrating to the next release of OpenJDK if you so choose as it becomes available. Each version of OpenJDK will include the security updates of the previous version.

What additional support comes with the paid Java SE (Standard Edition) subscription from Oracle? Oracle lists these benefits and capabilities on their website:

- Oracle Java SE licensing and support for cloud deployments, server, and desktop
- Access to performance, stability, and security updates directly from Oracle
- Access to several Oracle Java SE versions past end of public updates (EoPU)
- Oracle Java SE 8 and 7 enterprise management, monitoring, and deployment features
- Around-the-clock My Oracle Support with support for 27 languages and fast-fix turnarounds
- Access to critical bug fixes before they get included in publicly available releases
- Simple monthly subscription pricing
- Access to Oracle Java SE 8 and 7 commercial features

- Access to updates for older releases of Oracle Java SE

If such support is important to you, you should consider paying for a Java SE subscription from Oracle.

## Download and Install Oracle JDK

If you want to download and install Oracle JDK, do the following.

1. Go to  
<http://www.oracle.com/technetwork/java/javase/downloads/>.
2. Scroll to the latest version of the Java SE Development Kit (JDK) and click the link for your computer operating system, either Linux, macOS, or Windows.
3. Click the download link for one of the installer files. I usually use the MSI file for Windows.
4. Double-click the downloaded file to install Java. You should be able to take all the installation defaults.

## Download and Install OpenJDK

If you want to download and install OpenJDK, do the following.

1. Go to  
<https://openjdk.java.net/>  
and click the JDK link in the download paragraph.
2. Click the build link for the JDK appropriate for your platform. JDK 19.0.2 for Windows x64 was used for this tutorial.
3. Save the file in  
`myFolder\Downloads\Java`
4. Scan the downloaded file for viruses.
5. Right-click on the downloaded file, select Extract All > Browse to extract the files to  
`C:\Program Files\Java`
6. Click Extract
  1. If you receive the warning that you need administrator permission to copy to this folder, click Continue.

Next, you must set some system variables for Java.

1. Go to Start > All apps > Windows System (or Windows Tools) > Control Panel, click View by: and select Large icons.
2. Click System > About.
3. Scroll down and click the Advanced system settings link.
4. Click Advanced > Environment Variables....

5. Under System variables, if JAVA\_HOME exists, select it and click Edit. If JAVA\_HOME does not exist, click New.
  1. Set JAVA\_HOME to the directory of the newly installed JDK and click OK. I entered  
C:\Program Files\Java\jdk-19.0.2
6. Under System variables, select PATH (or Path) and click Edit > New. Add "%JAVA\_HOME%\bin" to the beginning of the list and click OK.
7. Under User variables, if Path exists, select Path and click Edit. Add "%JAVA\_HOME%\bin" to the beginning of the list and click OK. If Path doesn't exist as a user variable, Windows will use the PATH system variable.
7. Click OK to close the Environment Variables window.
8. Click OK to close the System Properties window.
9. Close the Control Panel window.

---

## Download and Install Eclipse

---

Eclipse is a free Java development environment. Eclipse simplifies typing program code, identifies programming errors, and even suggests ways to fix those errors.

It's recommended that you install Eclipse in your Windows user folder, but you can put it anywhere on your computer. I put mine in my user folder, C:\Users\Annette.

1. Download the latest version of Eclipse IDE from  
<http://www.eclipse.org/downloads/>

to any location on your computer.

2. Double click the downloaded file to start the Eclipse installer.
3. Click Eclipse IDE for Java Developers.
4. Click Install. This defaults to installing Eclipse to C:\Users\Annette\Eclipse\java-2022-12. If more than one person will be using this computer, other users should install Eclipse to a different folder.
5. Close the installer window.

Create a shortcut on your desktop for Eclipse. These instructions are for Windows 7 and 10:

1. Open the eclipse folder where Eclipse was installed. My eclipse folder was at  
C:\Users\Annette\Eclipse\java-2022-12\eclipse
2. Right-click on eclipse.exe and choose copy.
3. Right-click on your desktop and choose Paste shortcut. You might have to click Show more options if you don't see the Paste shortcut option.
4. If you aren't the only one who will use Eclipse for programming on your computer, right-click on the shortcut you just created and rename it for your use. For example, I renamed mine "Eclipse for Annette."

## Set up Eclipse

Now that everything is installed, set up Eclipse.

1. Double-click your Eclipse shortcut. The Workspace Launcher window should appear:

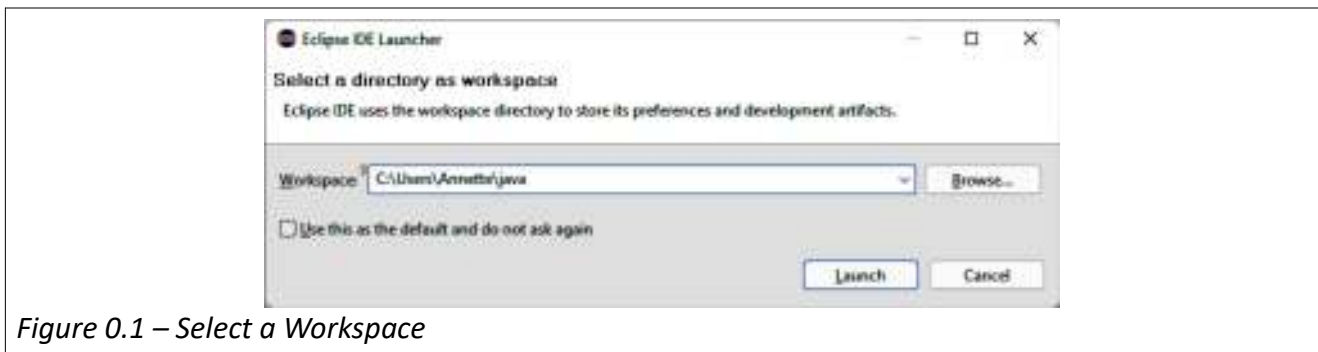


Figure 0.1 – Select a Workspace

2. Click Browse... to find your Java work folder and then click Launch. The Welcome to the Eclipse IDE for Java Developers window should appear.

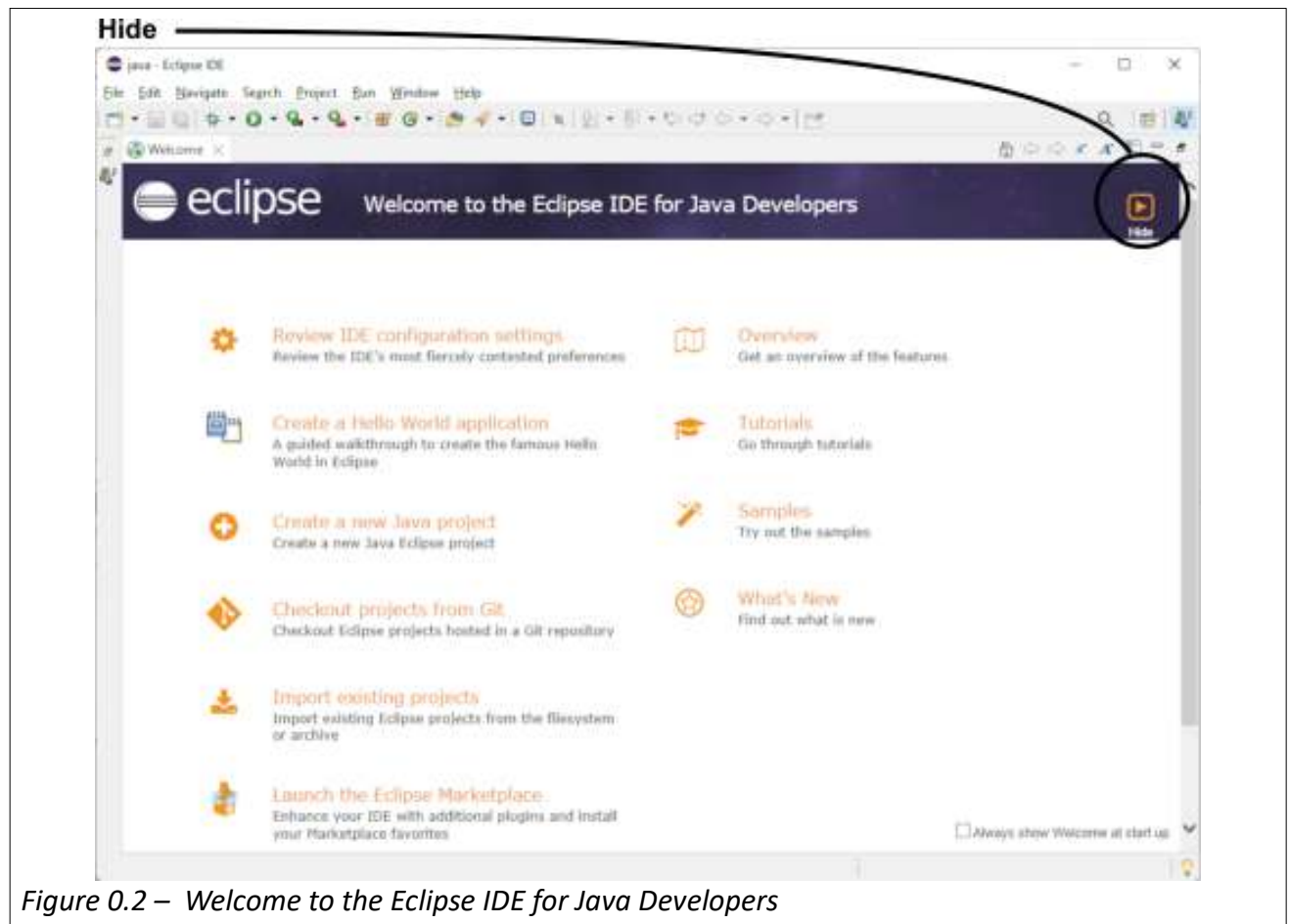


Figure 0.2 – Welcome to the Eclipse IDE for Java Developers

3. Click Hide to see the main Eclipse window.

Use the Java perspective:

1. If the Perspective indicator isn't showing the Java Perspective icon, click Window > Perspective > Open Perspective > Java.

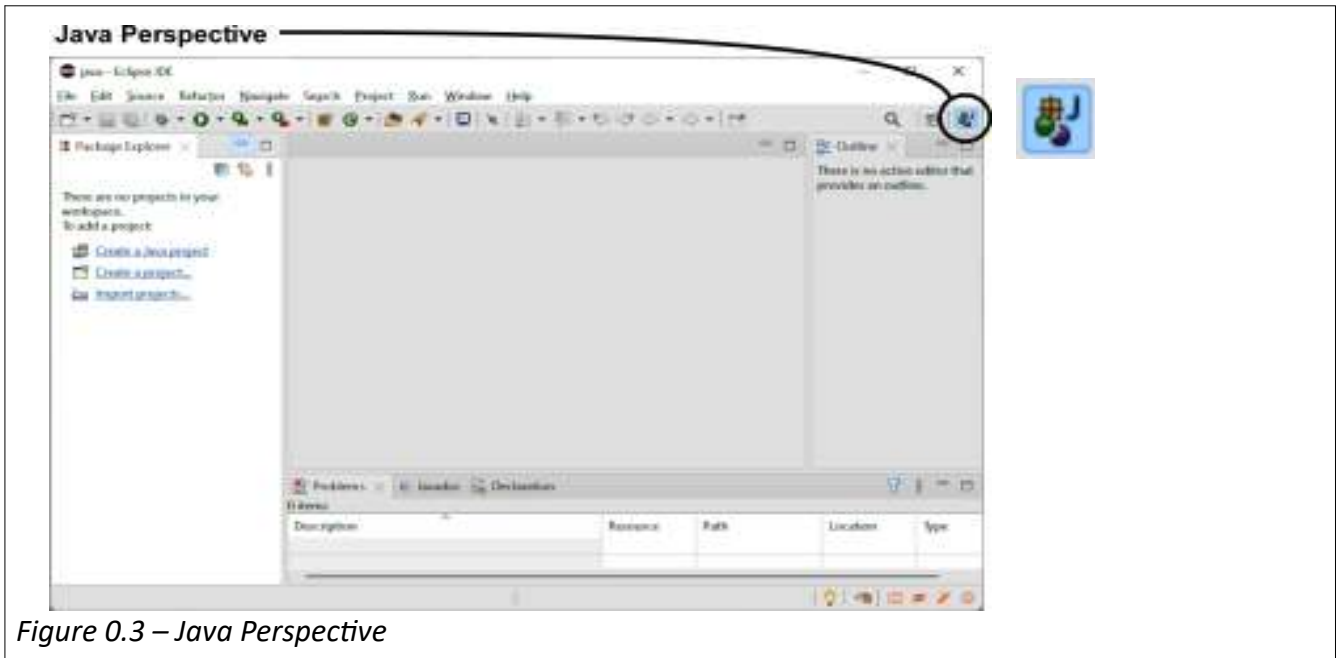


Figure 0.3 – Java Perspective

Make more room for writing code.

1. The Outline tab should be the only tab in the right pane. If there are any other tabs in the right pane, close them. We won't need them for the projects in this book.
2. Move the Outline tab to the bottom of the Package Explorer pane:

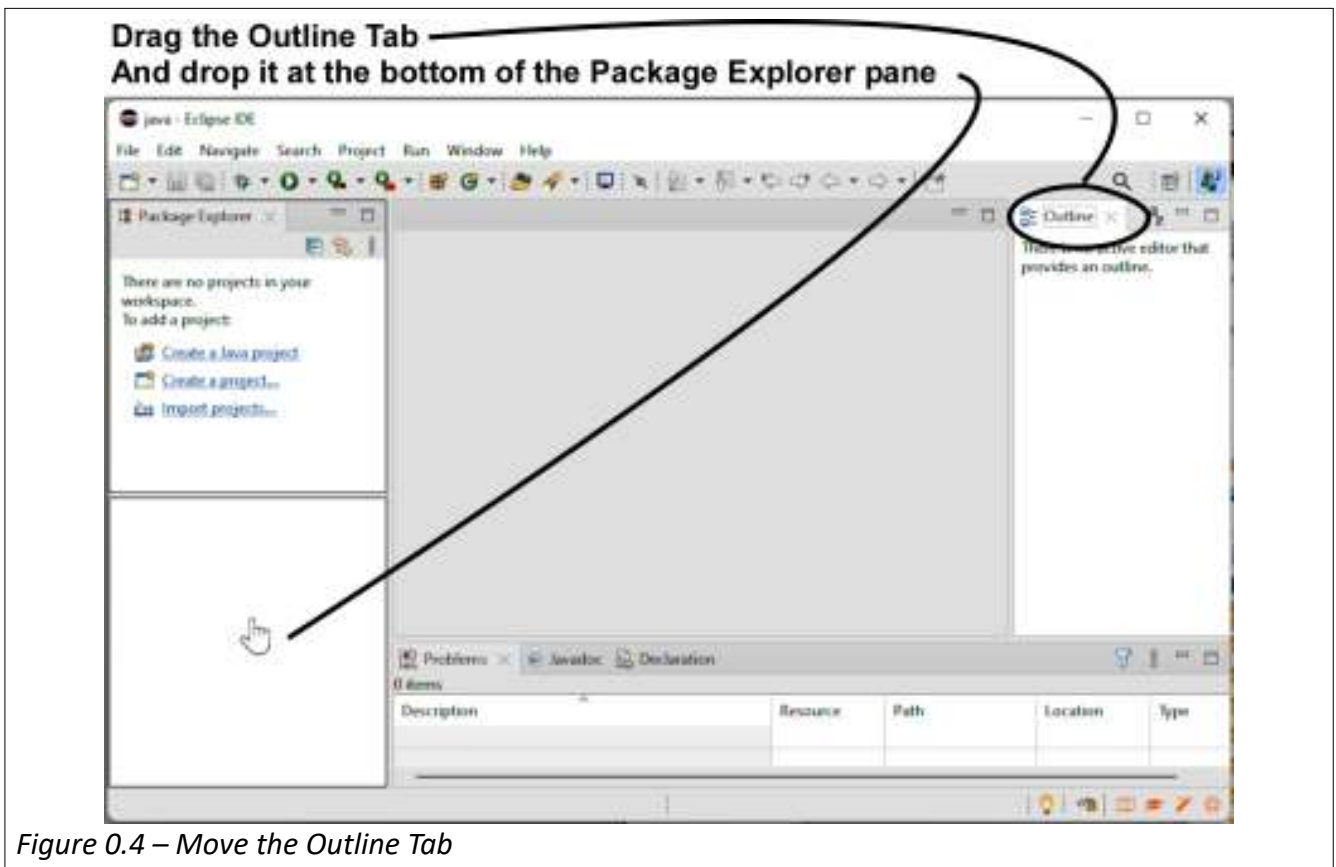


Figure 0.4 – Move the Outline Tab

You now have everything you need to create your first program!

## **Create a Java Program**

Programs are first organized by Java projects, then by packages within the Java projects. You'll create a Java project for each program in this book.

Packages hold program files that are usually used together. Because the programs in this book will be small, most of the Java projects you create will have only one package.

Java programs are made from one or more classes. Classes contain the actual program code: the instructions that, when run in sequence, perform a desired task.

Every class must name some other class as its superclass. For example, programs intended to run in a window must name some type of window class as its superclass.

This little tutorial will walk you through creating a program called Hello World. You'll need to create a project, a package, and a class.

Create a Java project called Hello World:

1. If Eclipse is no longer open:
  1. Double-click the Eclipse shortcut you created on your desktop.
  2. Click OK to use your Java work folder as your workspace.
2. Click File > New > Java Project.

**Name the project**      **Do not create module-info.java file**  
**Use default JRE (18 or higher)**      **Create Separate Folders**

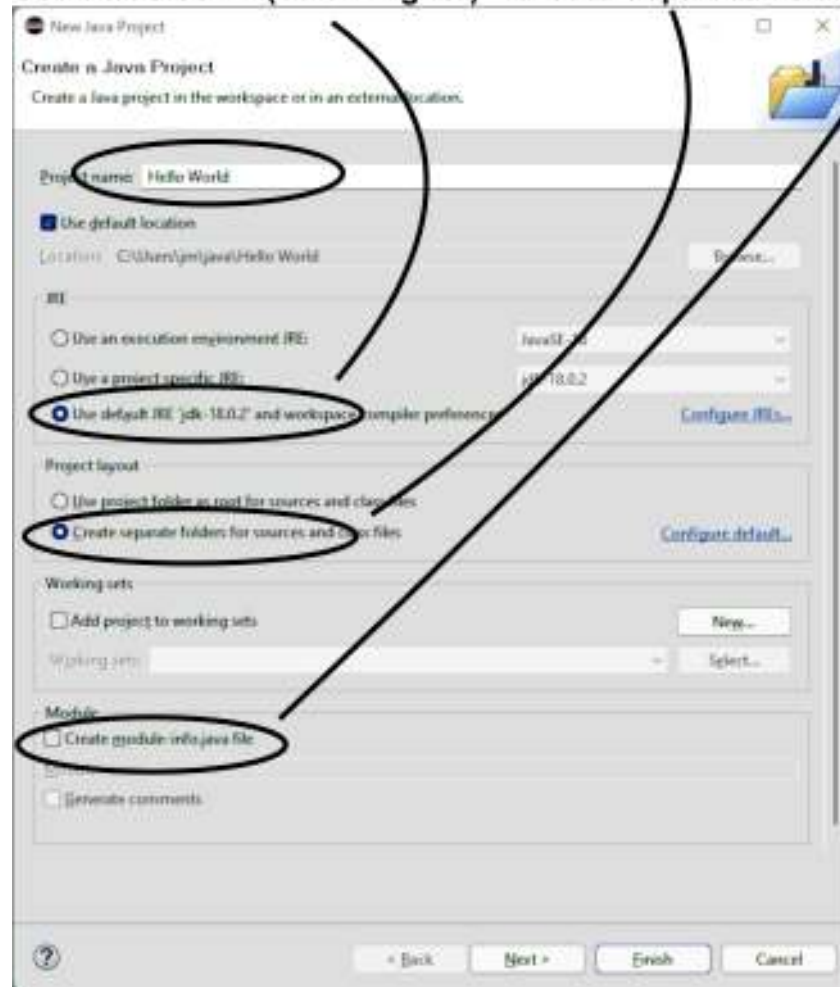


Figure 1.0 – Create a Java Project

3. Use the following settings for the Java project.
  1. Name the Java project “Hello World”
  2. Select Use Default JRE. If you get an error that the current workspace uses a JRE version with higher compliance level, it is because your default JRE is more recent than was available when Eclipse was created. In that case, select the option to Use an execution environment JRE and select the level that matches the compliance level in the error message.
  3. Select Create Separate folders for source and class files.
  4. Do not select Create module-info.java file.
4. Click Next.
5. If your project requires use of an existing JAR file, add the JAR file to the classpath now.
  1. Click Libraries, select Classpath, then click Add External JARs.





Figure 1.1 – Add External Jar to the Classpath

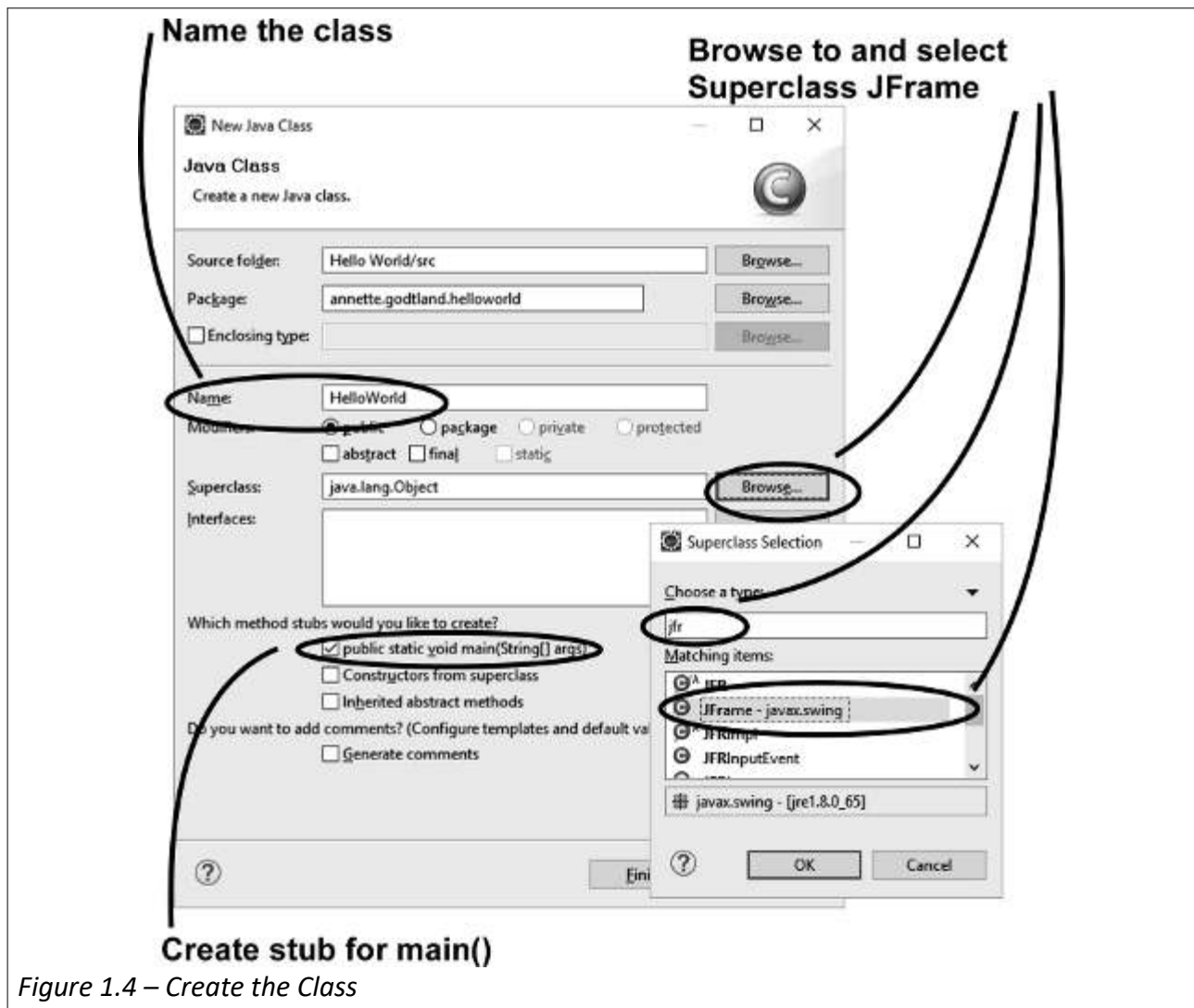
2. Browse to and select the JAR file. The above screenshot adds an external JAR file named DIYJava.jar from my Java work folder.
3. Click Open.
6. Click Finish.



The Package Explorer pane now shows the package you created in your Hello World project.

Next, create the main class for the Hello World program using the JFrame class as its superclass:

1. Right-click on your package in the Package Explorer pane and choose New / Class.
2. Enter HelloWorld for the class name, as shown in the image below. (Hint: there's no space between "Hello" and "World".)
3. Click Browse for Superclass.
  1. Enter "jfr" for the type, select JFrame, and click OK.
4. For Which method stubs(stub) would you like to create?, select only public static void main (String[] args).



5. Click Finish to create the class.

Eclipse will create code for a HelloWorld class that looks like the following listing.

*Listing 0-0, from HelloWorld.java*

```
package anette.godtland.helloworld;
```

```
import javax.swing.JFrame;

public class HelloWorld extends JFrame {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

    }

}
```

---

Save and run the program.

1. Press Ctrl-S to save the program.
2. Click the Run button, as shown in this image, to run the program:



Figure 1.5 – Run the Program

A tiny window should appear with nothing in it, like this:



Figure 1.6 – The Program Window

The lessons in the Do-It-Yourself Java Games books will teach you how to put content into a window.

### Add a Project to the Build Path

The build path holds the location of the projects and libraries needed for creating the program in the Java project. When a program needs to use code that is in a different project or library, the other project or library must be added to the build path of the program's project.

For example, program code that will be shared between multiple programs in these books will be in a separate Java project. The shared project needs to be added to the build path of the project of each program that will use the shared code.

To add additional projects of shared code to a new project as the new project is being created:

1. Click the Next button.

2. Click the Projects tab.
3. Select Classpath.
4. Click Add....
5. Select the project to add, then click OK.
6. Click Finish to finish creating the new project.

To add additional projects of shared code to an existing project:

1. Right-click on the existing project and select Build Path / Configure Build Path...
2. Click the Projects tab.
3. Click Classpath.
4. Click Add....
5. Select the project to add, then click OK.
6. Click OK to close the Java Build Path window.

## **Export a Program**

You must export a finished program if you want to run it without Eclipse or if you want to run it on a different computer.

If you export your program, it will be made into a JAR file (a Java ARchive file) on your computer. JAR files have names that end with “.jar”. To run the program, double-click on the JAR file name.

To export a program:

1. Right-click on the Java project name in the Package Explorer pane and choose Export....
2. Select Runnable JAR File, as shown in the image below, and click Next.
3. Select the launch configuration for your project. Eclipse creates a launch configuration for every new program you run in Eclipse, and adds the launch configuration to this list. The image below shows selecting the launch configuration for the ASCII Art project which was a program created in the first Do-It-Yourself Java Games book.
4. Enter a name for the JAR file as the Export destination. Naming the JAR file after the Java project will help you remember what program the JAR file contains. The image below shows the JAR file being named ASCIIArt.jar.
5. Select the option to Package required libraries into generated JAR, as shown in this image:

**Export as Runnable JAR file**  
**Select ASCII Art launch configuration**  
**Save as ASCII Art.jar**  
**Package the required libraries**

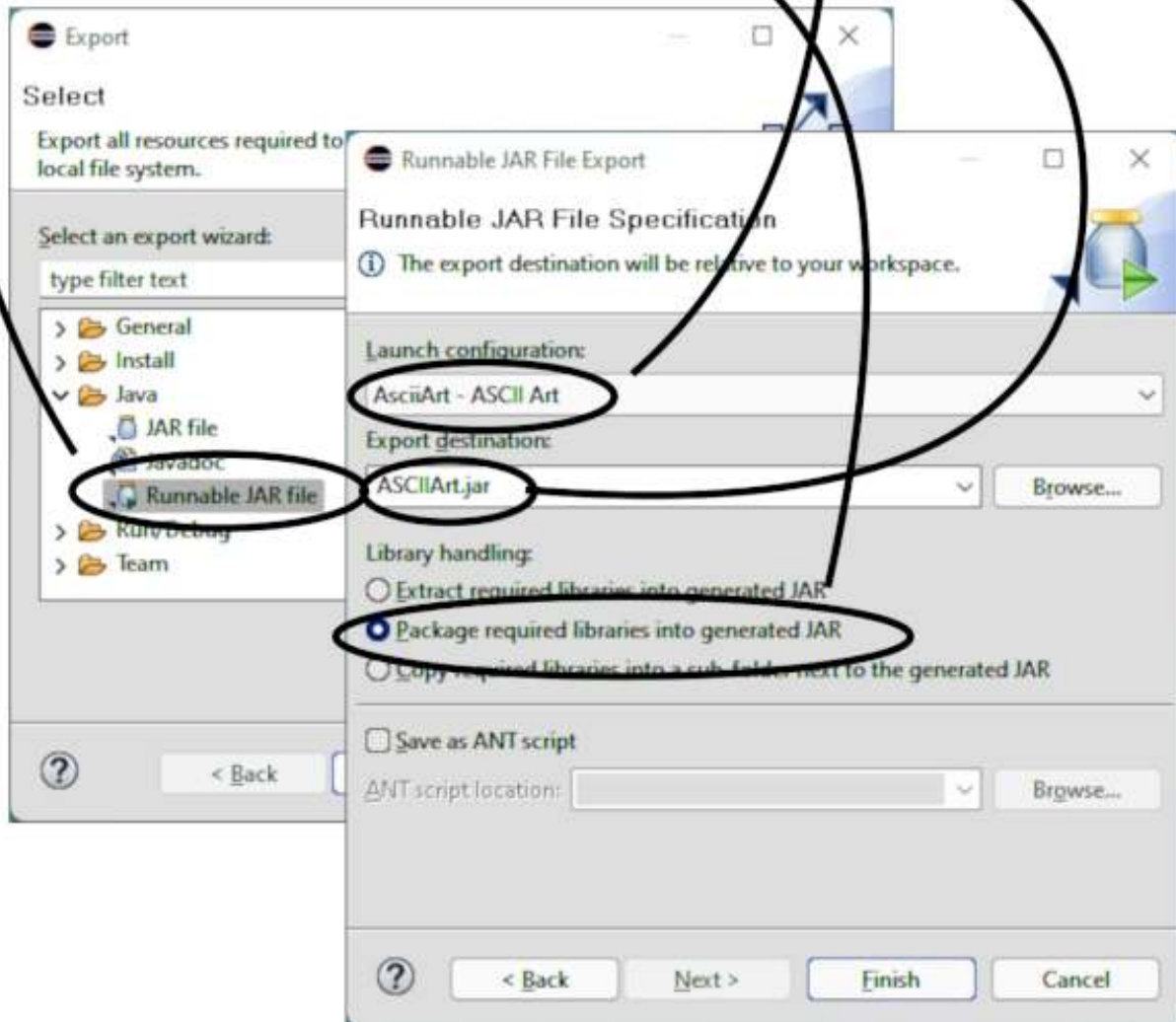


Figure 1.8 – Export a Project

6. Click Finish to complete the export.

---

## ***Run Your Program Outside of Eclipse***

---

To run your program without Eclipse:

1. Find your newly created JAR file in your Java work folder.
2. Double-click the JAR file name.

You may copy the JAR file to any other folder or any other computer. However, remember to copy any files in the Java work folder that are used by the program, to the same folder as the JAR file.

## **Tips For Running the Exported Program:**

- Export the program as a JAR file if you want to run the program outside of Eclipse or on a different computer.
- You can save different versions of your program in different JAR files. Use all the same export options but name a different Export destination for each version of the program.
- Your exported JAR file for the programs created in these books can be run on any computer running a Windows, OS X, or Linux operating system, if a Java Runtime Environment, version 1.7 or later, is installed.
  - The Java Runtime Environment was included in the Java Development Kit you installed. Therefore, the Java Runtime Environment also was already installed on your computer.
- If there were any files in the Java work folder that were used by the program, always keep a copy of those files in the same folder as the exported JAR file.
- The JAR file will contain all the .class files that are in the bin folder of the Java work folder. If you want any other files that are in the bin folder inserted into the JAR file, copy them to the src folder before you export the project.
- If double-clicking the JAR file did not run the program, you should be able to run the JAR file from a command prompt.
  - Click Start > Windows System (or Windows Tools) > Command Prompt
  - Use the change directory command, `cd`, to change the directory to the location of the JAR file. For example, mine was at `C:\Users\Annette\java`. So I would enter the following command.  

```
cd C:\Users\Annette\java
```
  - Enter the Java command, `java`, with a `jar` parameter, `-jar`, to run the JAR file.  

```
java -jar ASCIIArt.jar
```
- If running the JAR file from the command prompt works but double-double clicking the JAR file did not work, the file type association may be configured incorrectly on your computer. Refer to this Windows Help Online article for steps that may correct the file type association problem.

<https://www.winhelponline.com/blog/repair-jar-file-association-windows/#regfile>

---

## ***Create Market-Ready Java Programs***

---

To simplify running your program on other computers, refer to the, *Create Market-Ready Java Programs*, by Annette Godtland, available on Amazon at

<https://www.amazon.com/dp/B0BZFPJTKK>

*Create Market-Ready Java Programs* demonstrates how to automate bundling your program, all its

resources, and a minimal Java runtime environment into a single, downloadable, signed file that will easily and properly install your program into the Windows environment. You'll learn to create a trial version, add some piracy protection, sell your program securely online, and much more.