

Python & Java {4} Teachers



Introduction to java

Level 1 - Python



Funded by the
Erasmus+ Programme
of the European Union



Python & Java 4 Teachers

Introduction To Java

Java is a general purpose, **object-oriented** programming language.

It was created in 1995 by James Gosling. It is owned by Oracle and more than **3 billion devices** run Java!

Java has a wide range of applications such as:

- Mobile applications
- Desktop applications
- Web applications
- Games
- Database connection



ORACLE



{4}

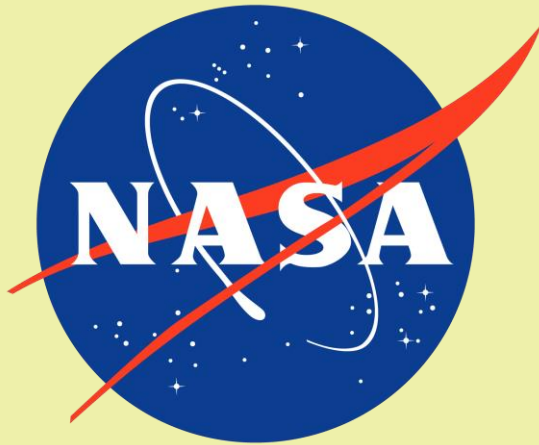
Companies that use Java

amazon

NETFLIX

NASA's open source
'WorldWind' software is
created using Java – click
[here](#) to try the demo!

LinkedIn



MINECRAFT

{4}

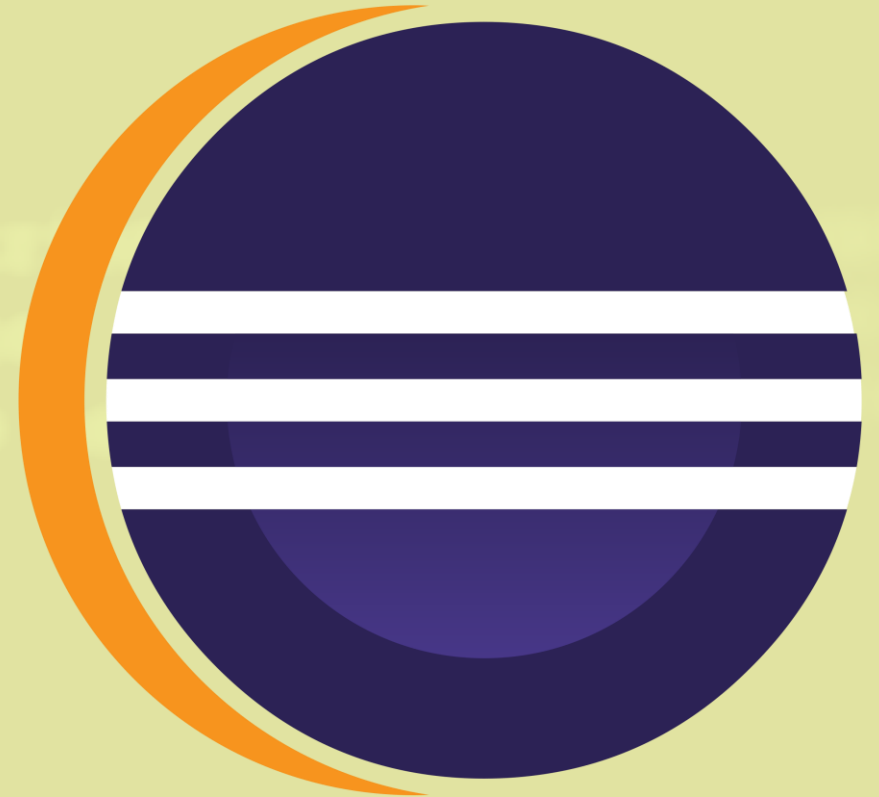
Installing Java

You can find the most recent version of Java [here](#) – download this before you get started.

To get started programming with Java, we need to install an IDE (or Integrated Development Environment).

IDEs are software used for building applications that combines useful developer tools in a single GUI (Graphical User Interface).

The IDE used in these resources will be [**ECLIPSE**](#)

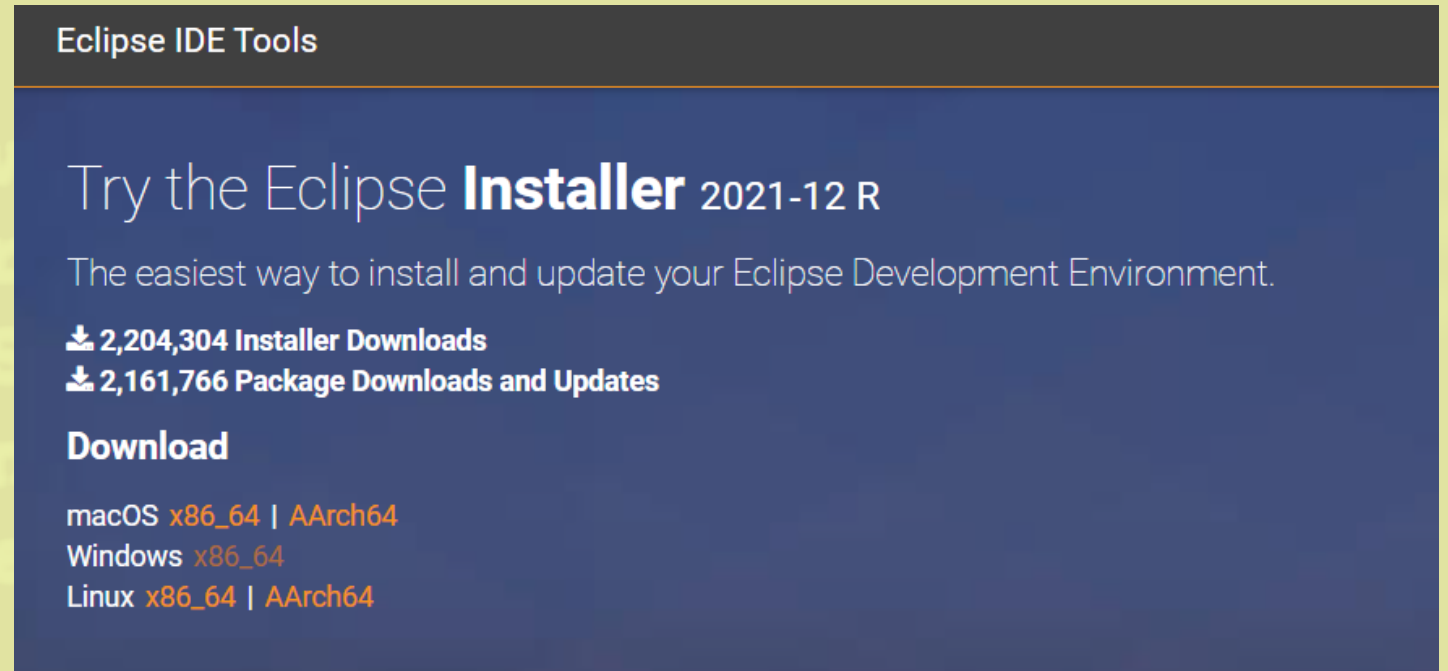


The Eclipse Installer

To install ECLIPSE, we will use the website's Eclipse Installer.

Click [here](#) for the Eclipse Installer.

When you arrive on the website, click on the appropriate bundle for your platform.



Eclipse IDE Tools

Try the Eclipse **Installer** 2021-12 R

The easiest way to install and update your Eclipse Development Environment.

📄 2,204,304 Installer Downloads
📄 2,161,766 Package Downloads and Updates

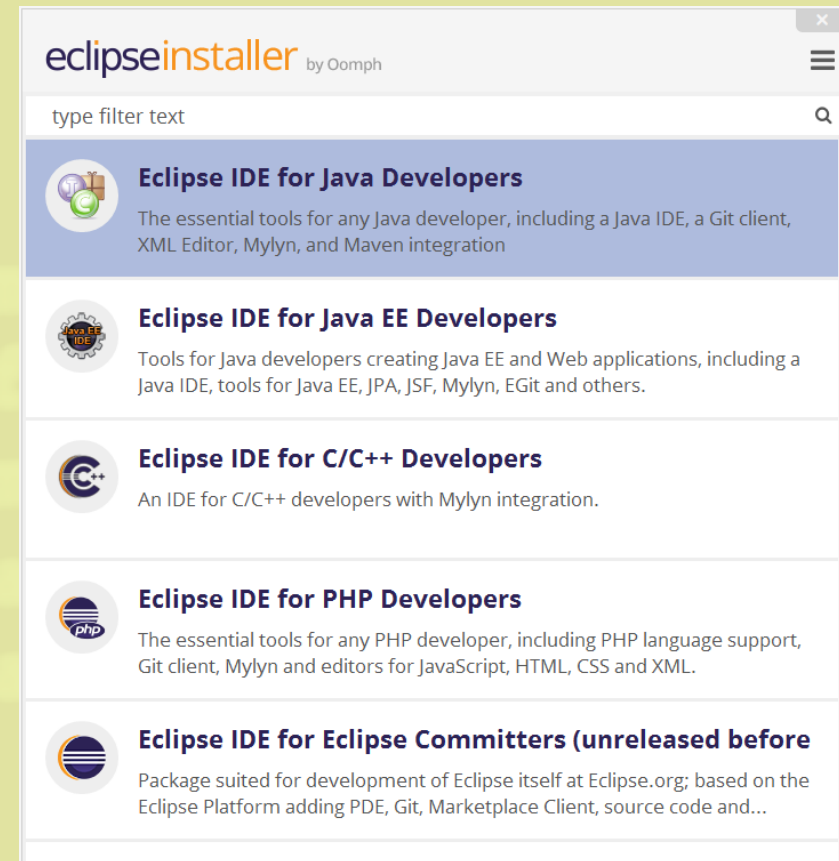
Download

macOS [x86_64](#) | [AArch64](#)
Windows [x86_64](#)
Linux [x86_64](#) | [AArch64](#)

The Eclipse Installer

The Eclipse Installer shows the packages available to Eclipse users.

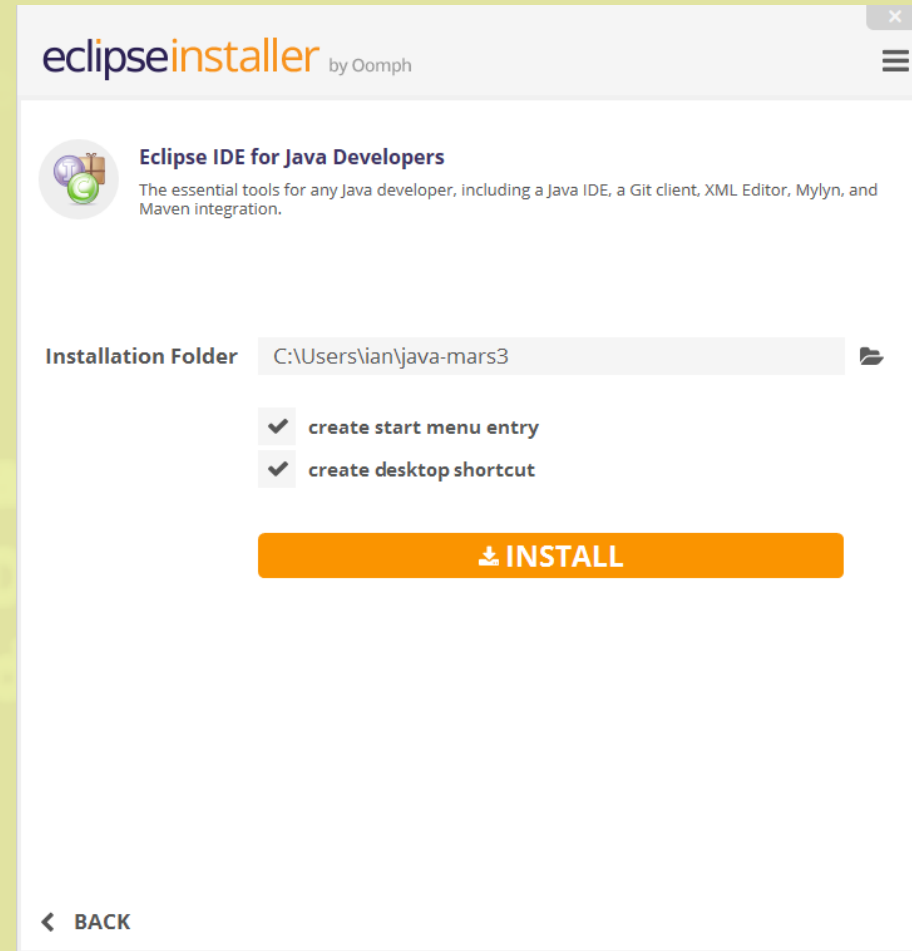
Make sure you select the **'ECLIPSE IDE for Java Developers'** option



The Eclipse Installer

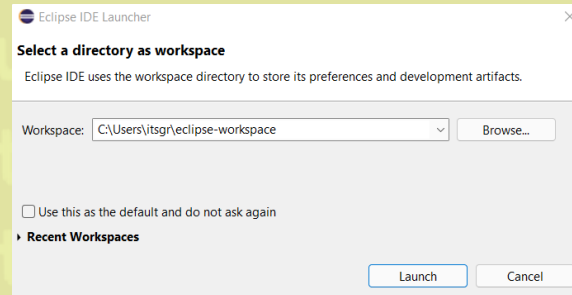
Specify the folder where you want Eclipse to be installed. The default folder will be in your User directory.

Select the 'Install' button to begin the installation.

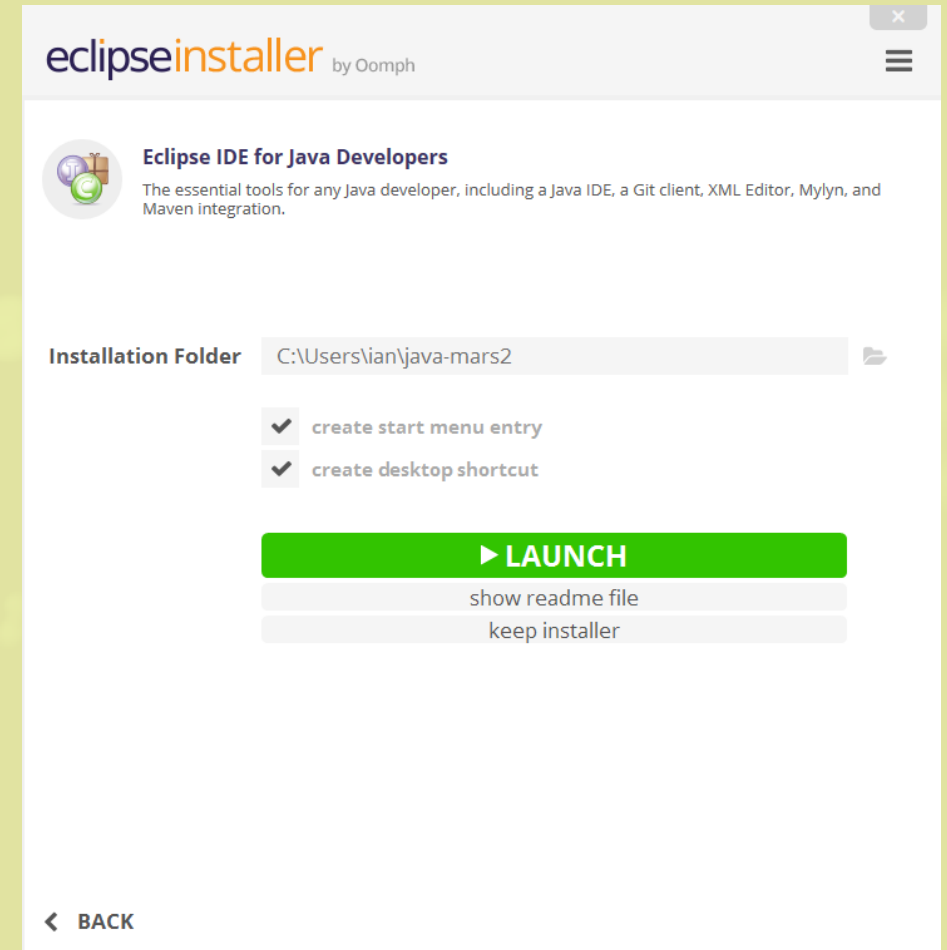


The Eclipse Installer

Once the installation process is complete you can now launch Eclipse!



Launch Eclipse in the default workspace.



{4}

For a more in depth tutorial of how to get started with Eclipse, click [here](#)...

Python & Java 4 Teachers

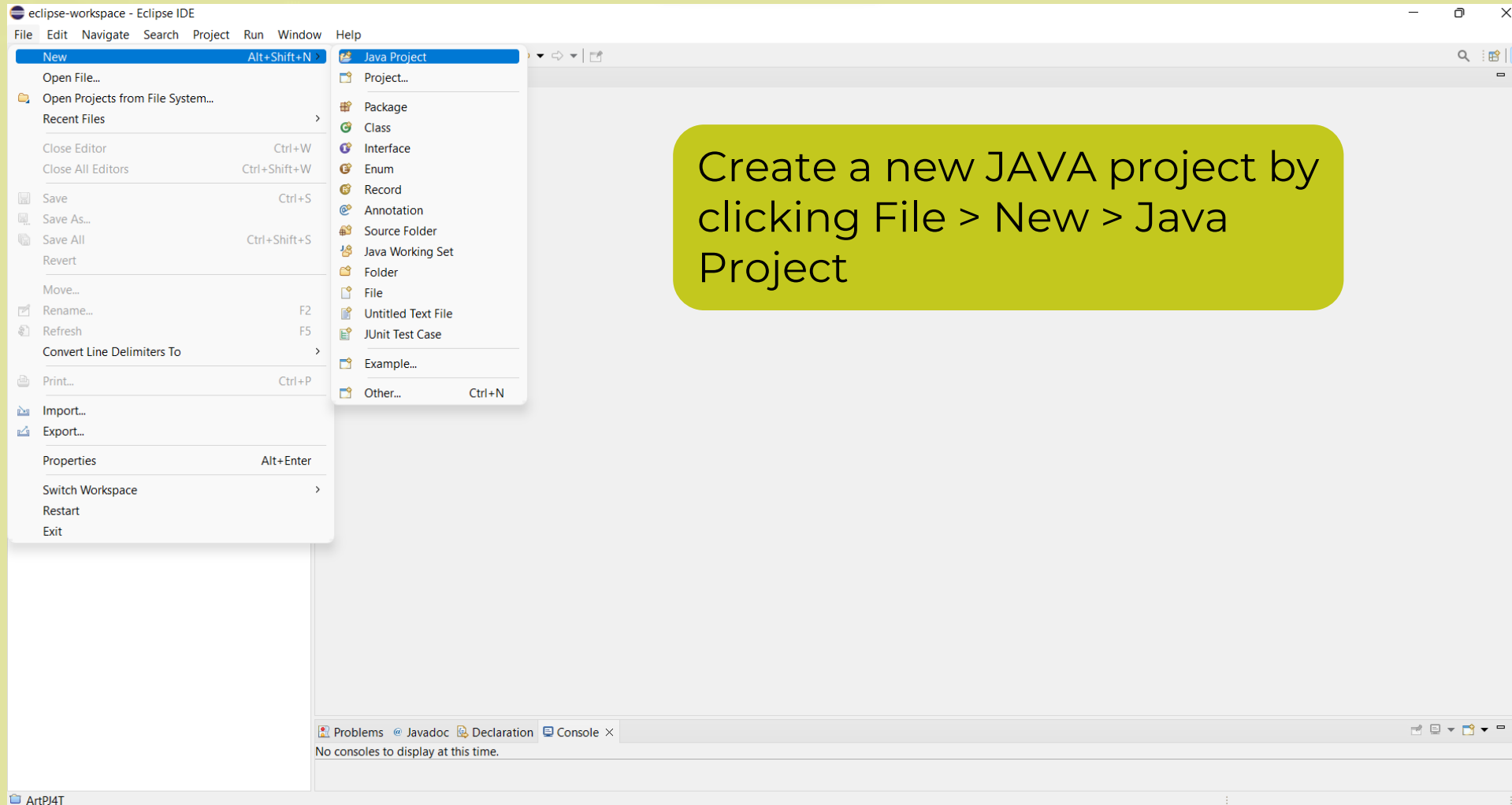
Creating our first Java program...

Steps to create a Java Program:

- Create a Java Project and module
- Create a package
- Create a class
- Write a program



Creating a Java Project



{4}

Creating a Java Project and Module

The image shows two overlapping dialog boxes from the Eclipse IDE. The background dialog is 'New Java Project', which has the following fields and options: 'Project name' set to 'HelloWorld', 'Use default location' checked (location: 'C:\Users\itsgr\eclipse-workspace\HelloWorld'), 'Use an execution environment JRE' selected with 'JavaSE-17' chosen, and 'Create separate folders for sources and class files' selected under 'Project layout'. The foreground dialog is 'New module-info.java', with 'Module name' set to 'helloworld' and 'Generate comments' unchecked. Both dialogs have 'Create' and 'Don't Create' buttons at the bottom.

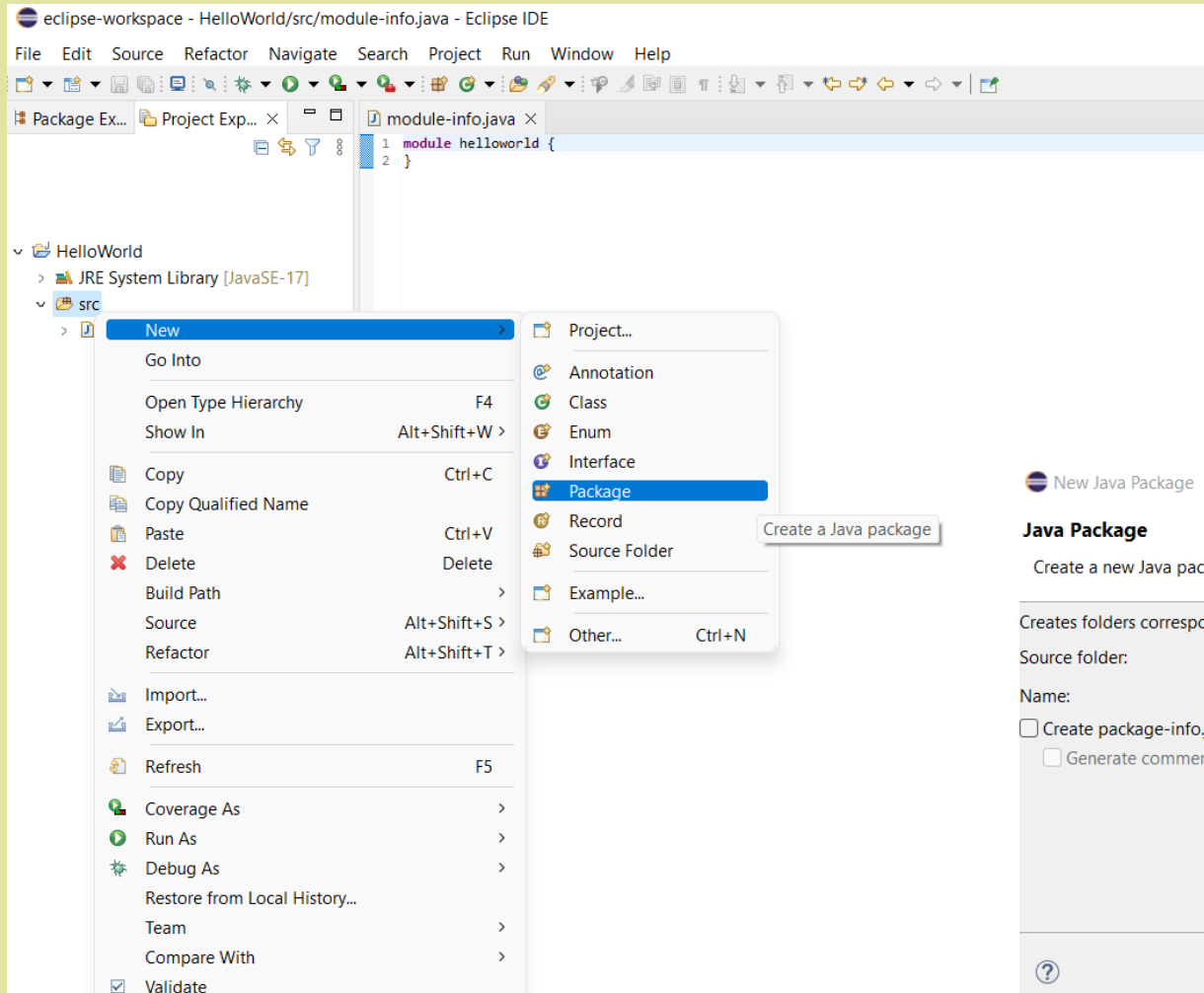
Give your project a name!

It's good luck to have your first project named as 'HelloWorld'!

You can also create a module-info.java – make sure the name you write uses lowercase letters!

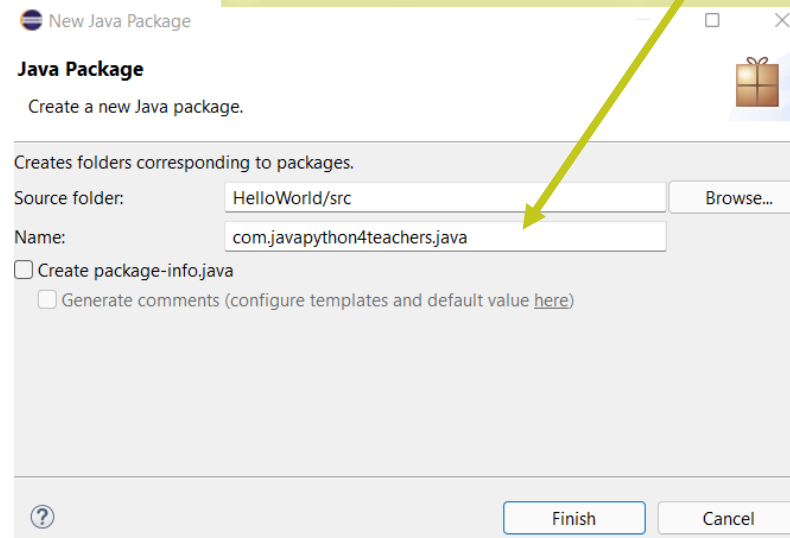
{4}

Creating a Package



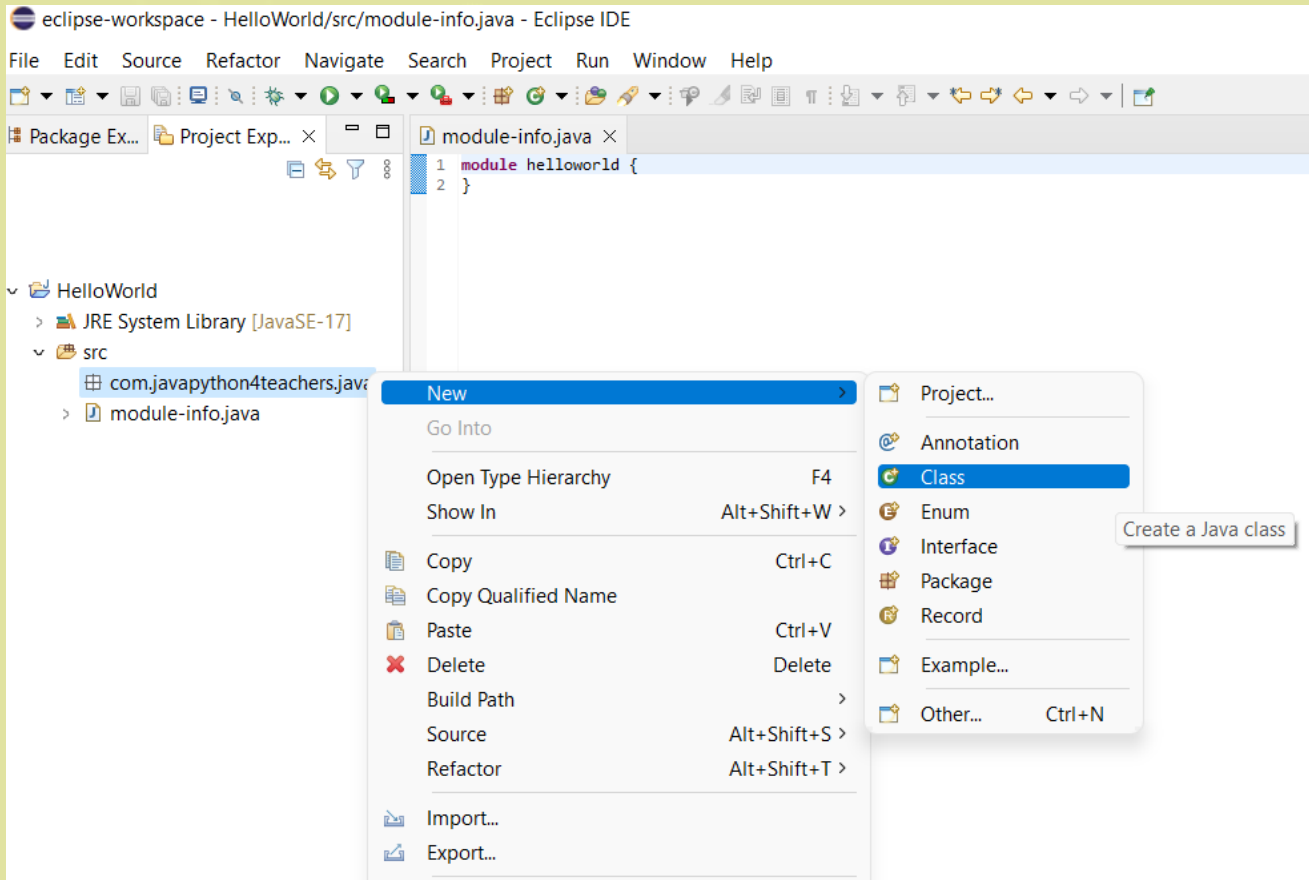
Right click on your 'src' folder and create a new Package.

Name your package something like this.



{4}

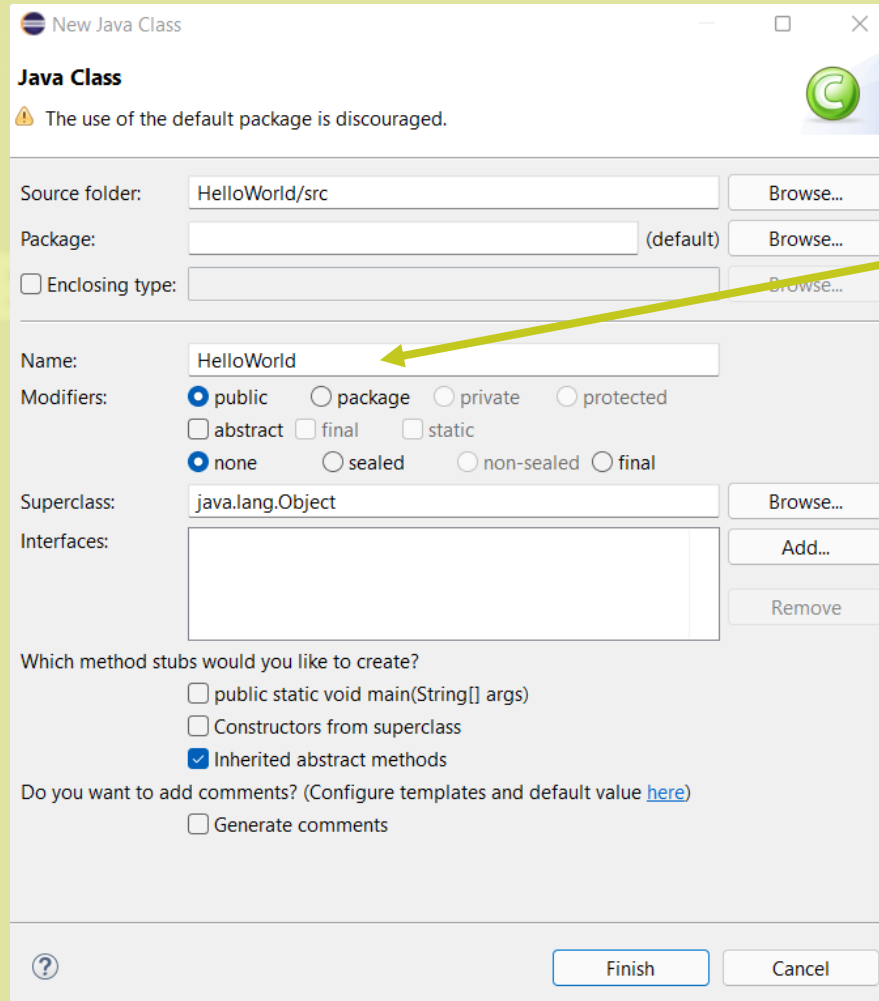
Creating a Class



Finally, right click on the package you just created and select a new 'Class'.

This class is where we will write our first Java program!

Creating a Class



New Java Class

Java Class

⚠ The use of the default package is discouraged.

Source folder: HelloWorld/src Browse...

Package: (default) Browse...

☐ Enclosing type: Browse...

Name: HelloWorld

Modifiers: ☒ public ☐ package ☐ private ☐ protected
☐ abstract ☐ final ☐ static
☒ none ☐ sealed ☐ non-sealed ☐ final

Superclass: java.lang.Object Browse...

Interfaces: Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

? Finish Cancel

Name your class 'HelloWorld'

Classes

A class is a way of grouping data and instructions that belong together.

A class may contain variables and methods.

A variable is a named location in memory in which we can store data.

A method groups together a set of operations or instructions to complete specific tasks

```
public class HelloWorld {  
  
}
```

Writing our first Java Program

```
HelloWorld.java ×  
1 package com.javapython4teachers.java;  
2  
3 public class HelloWorld {  
4  
5  
6  
7 }  
8  
9
```

Your first Java program should look something like this!

All of the code we write will be inside our 'HelloWorld' class

The main() method

A Java program is a sequence of Java instructions that are executed in a certain order – a Java program has a start and an end.

A Java program needs to start somewhere – the entry point of Java programs is the 'main()' method.

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

A method is a set of instructions that can be executed by running a single line of code (or calling it). By calling a method, you execute all instructions inside of it.

Writing our first program - add main()

Add a main() method into your program.

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

Writing our first program - print out "Hello World"

Print out 'Hello World' to the console using a System command.

```
package com.javapython4teachers.java;

public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("Hello World!");

    }

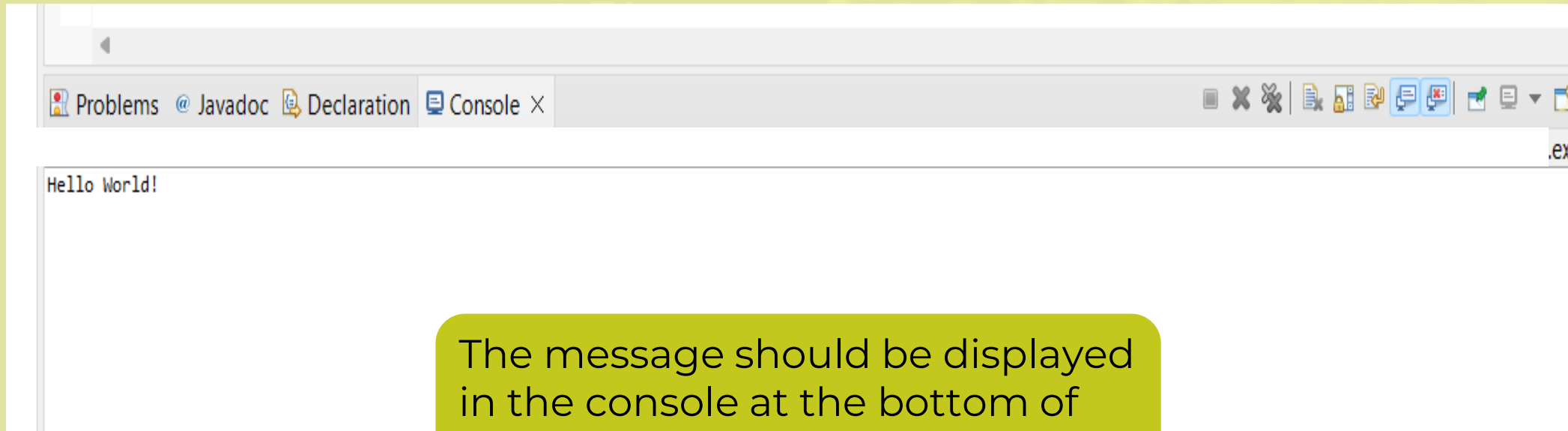
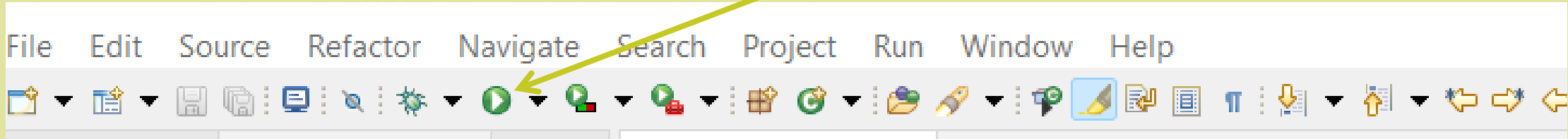
}
```

If you've programmed in Python before, this is the equivalent of a print() statement

Note that every line ends in a semicolon!

Running the program...

Click the green play button to compile and run the program



The message should be displayed in the console at the bottom of the screen

Challenge

Can you expand the code to print out the following to the console?

```
J           A           V           V           A
J           A A        V           V           A A
J           A   A      V   V      A   A
J           A     A      V           A     A
```

Solution

```
package com.javapython4teachers.java;

public class HelloWorld {

    public static void main(String[] args) {

        System.out.println("  J      A      V      V      A");
        System.out.println("  J      A A    V      V      A A");
        System.out.println("  J      A  A    V  V    A  A");
        System.out.println("  J      A    A    V      A  A");

    }

}
```

Conclusion

By the end of this resource, you should have a working HelloWorld class that prints out a message to the user.

Learning Objectives:

- Download Eclipse and Java
- Learn how to make a new java project, package and class
- Learn how to write a basic program in Java using `System.out.println()`

