

ENG23 2032 – OBJECT ORIENTED TECHNOLOGY

Week 01 : Introduction to OOP



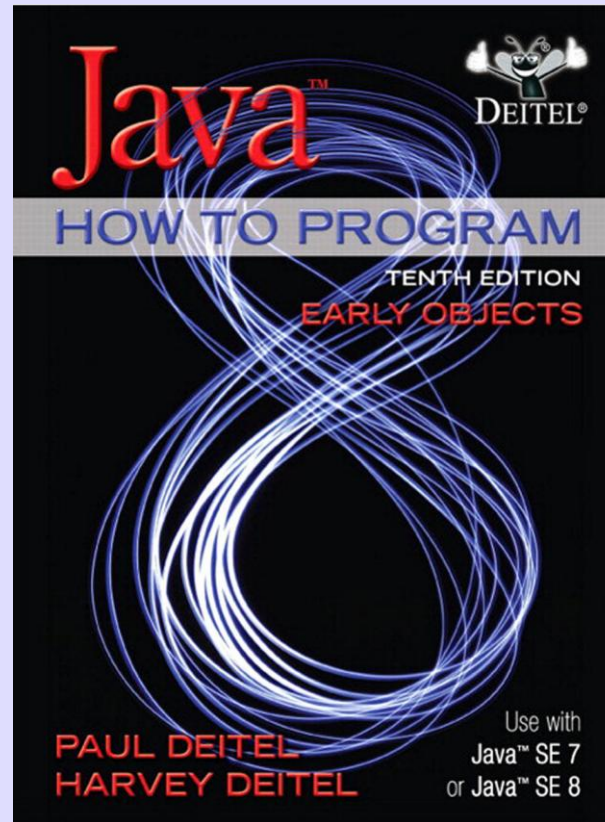
Ratiporn Chanklan

Assessment

10%	Attendance
30%	Weekly Lab Assignments:
10%	Lab Tests:
10%	Lab Exam:
20%	Midterm Exam:
20%	Final Exam:



Indicative Reading



Paul Deitel and Harvey Deitel, *Java: How to Program (Early Objects) (10th Edition)*

What is Java?

- Java is a popular programming language, created in 1995.
- It is owned by Oracle, and more than 3 billion devices run Java.
- It is used for:
 - Mobile applications (specially Android apps)
 - Desktop applications
 - Web applications
 - Web servers and application servers
 - Games
 - Database connection
 - And much, much more!

Integrated Development Environment (IDE)



VS CODE

Visual Studio Code is a source code editor.

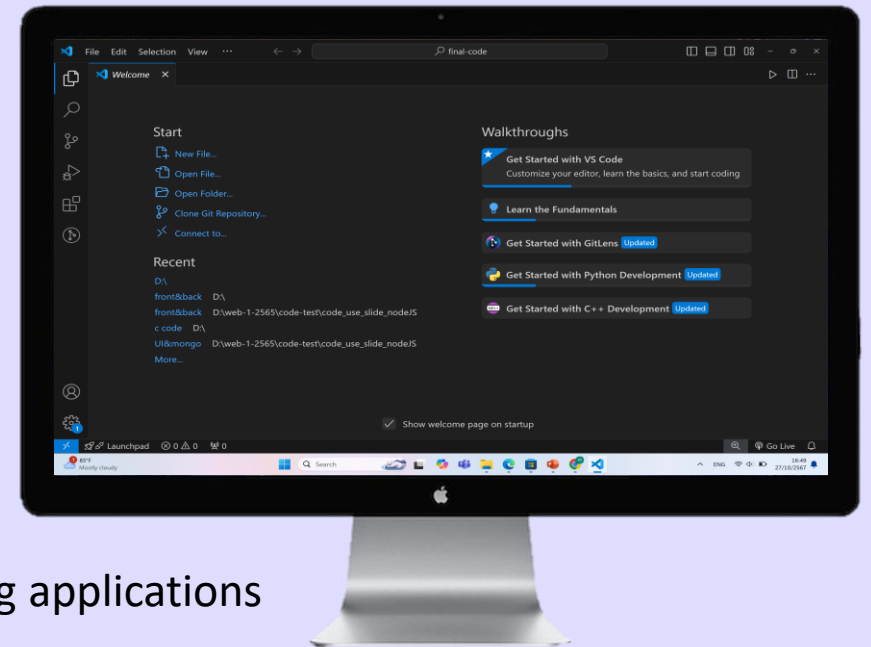
<https://code.visualstudio.com/>



JDK (Java Development Kit)

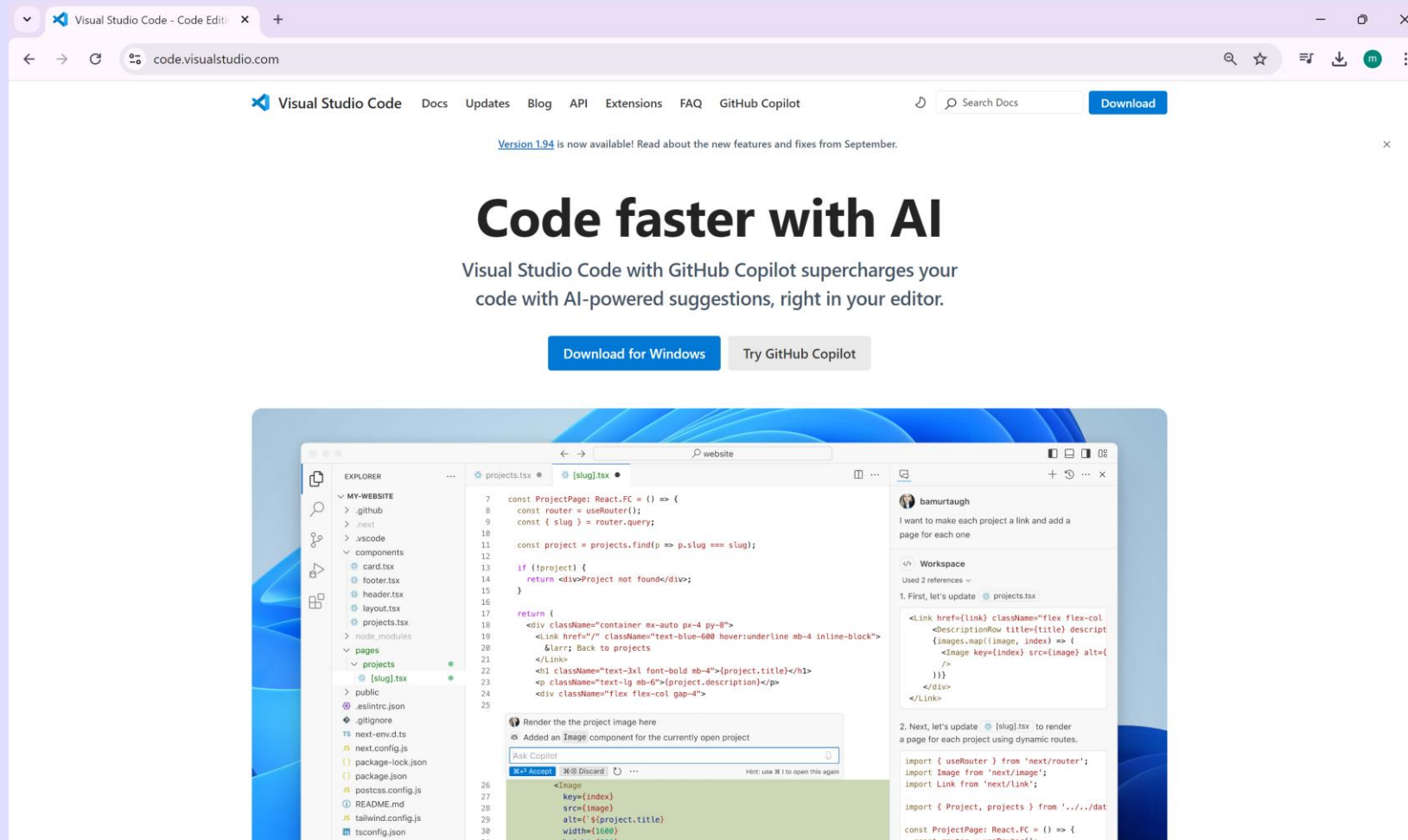
The JDK is a development environment for building applications using the Java programming language.

<https://www.oracle.com/java/technologies/downloads/>



Install Visual Studio Code.

- VSCode Link:- <https://code.visualstudio.com/>



Install the Coding Pack for Java (1)

- <https://code.visualstudio.com/docs/java/java-tutorial>

The screenshot shows the Visual Studio Code documentation website. The browser's address bar displays the URL `code.visualstudio.com/docs/java/java-tutorial`. The page features a navigation sidebar on the left with categories like Overview, SETUP, GET STARTED, USER GUIDE, SOURCE CONTROL, TERMINAL, GITHUB COPILOT, LANGUAGES, and various programming languages including JAVA. Under the JAVA section, 'Getting Started' is highlighted. The main content area is titled 'Getting Started with Java in VS Code' and includes an 'Edit' button. Below this, there's a section 'Setting up VS Code for Java development' and a 'Coding Pack for Java' section. In the 'Coding Pack for Java' section, two blue buttons are visible: 'Install the Coding Pack for Java - Windows' and 'Install the Coding Pack for Java - macOS'. The 'Windows' button is highlighted with a red dashed border. A note at the bottom states that the Coding Pack is only available for Windows and macOS. On the right side, there's a section 'IN THIS ARTICLE' with a list of topics and links to 'Subscribe', 'Ask questions', 'Follow @code', 'Request features', 'Report issues', and 'Watch videos'.

Visual Studio Code Docs Updates Blog API Extensions FAQ GitHub Copilot

Version 1.94 is now available! Read about the new features and fixes from September.

Getting Started with Java in VS Code

This tutorial shows you how to write and run Hello World program in Java with Visual Studio Code. It also covers a few advanced features, which you can explore by reading other documents in this section.

For an overview of the features available for Java in VS Code, see [Java Language Overview](#).

If you run into any issues when following this tutorial, you can contact us by entering an [issue](#).

Setting up VS Code for Java development

Coding Pack for Java

To help you set up quickly, you can install the **Coding Pack for Java**, which includes VS Code, the Java Development Kit (JDK), and essential Java extensions. The Coding Pack can be used as a clean installation, or to update or repair an existing development environment.

Install the Coding Pack for Java - Windows

Install the Coding Pack for Java - macOS

Note: The Coding Pack for Java is only available for Windows and macOS. For other operating systems, you will need to manually install a JDK, VS Code, and Java extensions.

IN THIS ARTICLE

- Setting up VS Code for Java development
- Installing and setting up a Java Development Kit (JDK)
- Creating a source code file
- Editing source code
- Running and debugging your program
- More features

[Subscribe](#)

[Ask questions](#)

[Follow @code](#)

[Request features](#)

[Report issues](#)

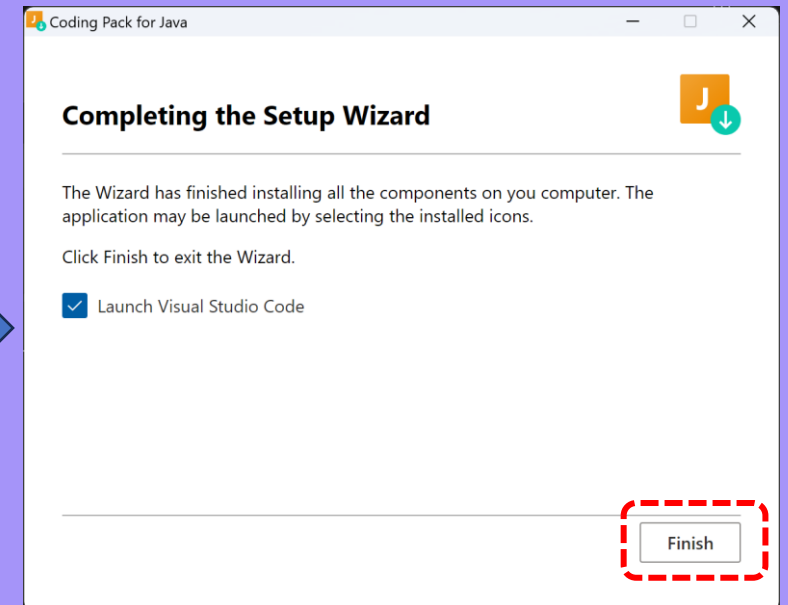
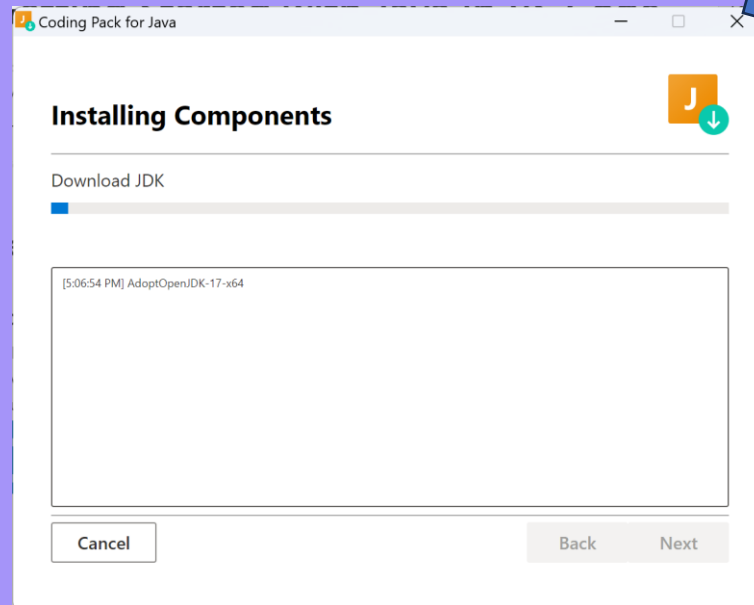
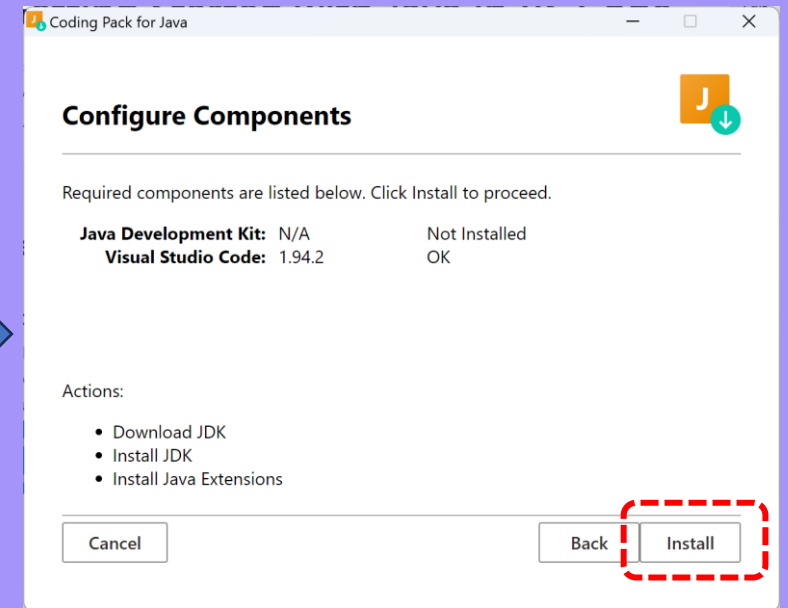
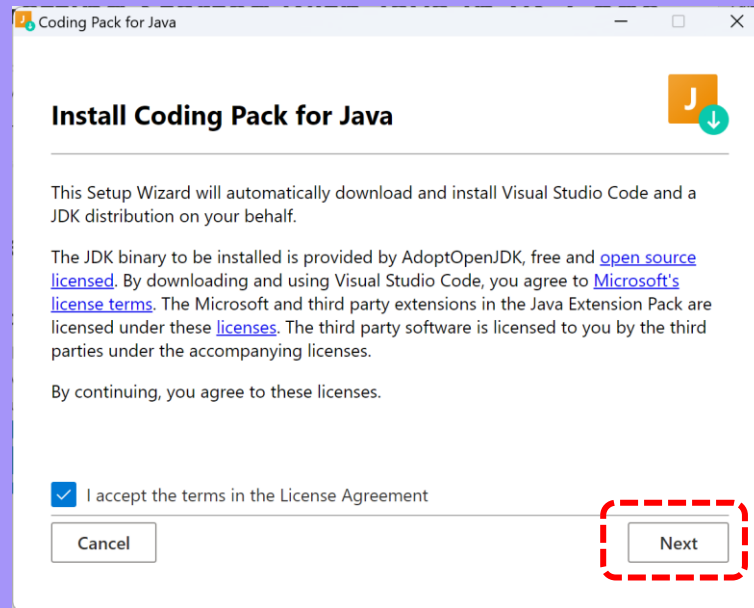
[Watch videos](#)

Install the Coding Pack for Java (2)

- Once you finish downloading the coding pack, click on the file to install it.

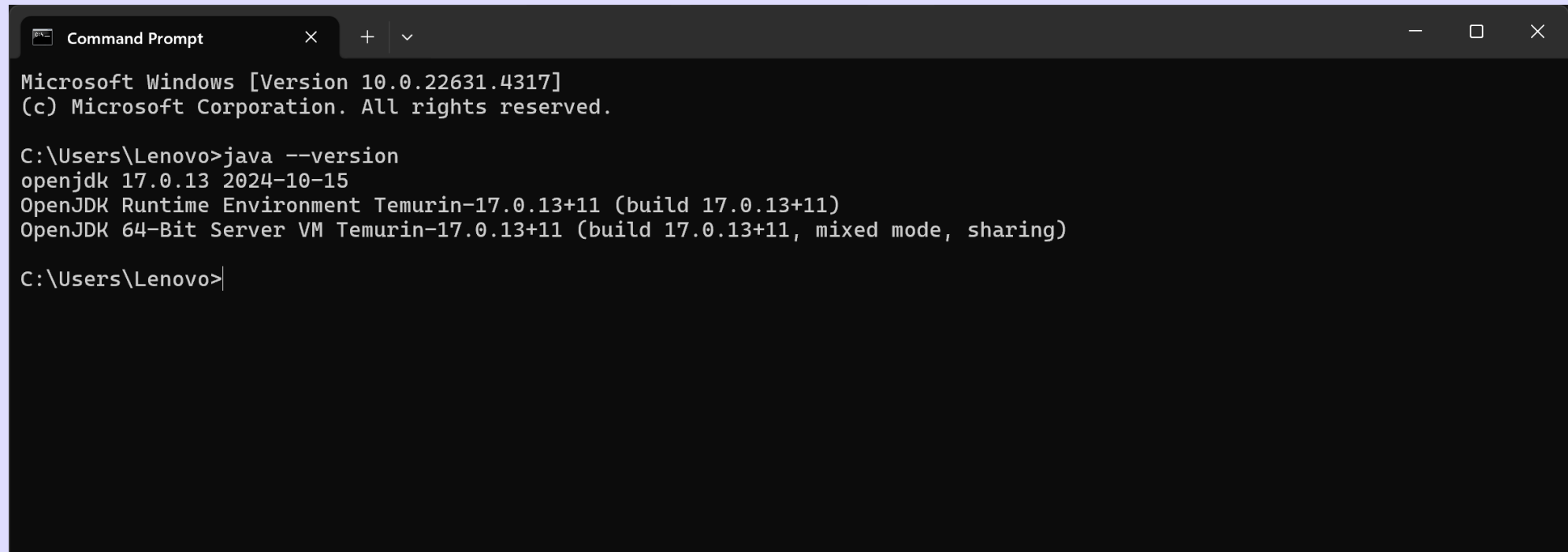
The screenshot shows a web browser window with the URL `code.visualstudio.com/docs/java/java-tutorial`. The page is titled "Getting Started with Java in VS Code" and includes a sidebar with navigation links like Overview, SETUP, GET STARTED, USER GUIDE, SOURCE CONTROL, TERMINAL, GITHUB COPILOT, LANGUAGES, NODEJS / JAVASCRIPT, TYPESCRIPT, PYTHON, and JAVA. The main content area is titled "Getting Started with Java in VS Code" and contains text about writing and running a Hello World program in Java. Below this, there is a section titled "Setting up VS Code for Java development" and a sub-section "Coding Pack for Java" which includes two buttons: "Install the Coding Pack for Java - Windows" and "Install the Coding Pack for Java - macOS". A note at the bottom states: "Note: The Coding Pack for Java is only available for Windows and macOS. For other operating systems, you will need to manually install a JDK, VS Code, and Java extensions." A red dashed box highlights a "Recent download history" overlay in the top right corner, showing a file named "JavaCodingPack-0.4.2.exe" with a size of 70.3 MB and a download time of 6 minutes ago. The overlay also includes a "Full download history" link.

Install the Coding Pack for Java (3)



Check your the Coding Pack for Java installation.

- Click on a search type in CMD
- Open a new Command Prompt and type: `java --version`



```
Command Prompt
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>java --version
openjdk 17.0.13 2024-10-15
OpenJDK Runtime Environment Temurin-17.0.13+11 (build 17.0.13+11)
OpenJDK 64-Bit Server VM Temurin-17.0.13+11 (build 17.0.13+11, mixed mode, sharing)

C:\Users\Lenovo>
```

Installing extensions (1)

- <https://code.visualstudio.com/docs/java/java-tutorial>

The screenshot shows a web browser displaying the Visual Studio Code documentation page for installing Java extensions. The page title is "Installing extensions". The left sidebar shows the navigation menu with "Getting Started" selected under the "JAVA" section. The main content area explains that users need to manually install a JDK, VS Code, and Java extensions. It lists several extensions: Language Support for Java™ by Red Hat, Debugger for Java, Test Runner for Java, Maven for Java, Project Manager for Java, and Visual Studio IntelliCode. A blue button labeled "Install the Extension Pack for Java" is highlighted with a red dashed box. Below the button, it states that the Extension Pack for Java provides a Quick Start guide and tips for code editing and debugging. At the bottom, there is a section titled "Tips for Beginners" with tabs for Quick Start, Code Editing, Debugging, and FAQ. The Quick Start tab is active, showing a 1-minute tutorial on creating a quick-start Java program in VS Code and setting up the workspace.

you will need to manually install a JDK, VS Code, and Java extensions.

Installing extensions

If you are an existing VS Code user, you can also add Java support by installing the [Extension Pack for Java](#), which includes these extensions:

- [Language Support for Java™ by Red Hat](#)
- [Debugger for Java](#)
- [Test Runner for Java](#)
- [Maven for Java](#)
- [Project Manager for Java](#)
- [Visual Studio IntelliCode](#)

Install the Extension Pack for Java

The [Extension Pack for Java](#) provides a Quick Start guide and tips for code editing and debugging. It also has a FAQ that answers some frequently asked questions. Use the command **Java: Tips for Beginners** from the Command Palette (**Ctrl+Shift+P**) to launch the guide.

Tips for Beginners

[Quick Start](#) [Code Editing](#) [Debugging](#) [FAQ](#)

In this 1-minute tutorial, we'll show you how to create a quick-start Java program in VS Code.

Setup the Workspace

VS Code Java works directly with **folders** that have source code. To setup the workspace, simply open a folder using **File > Open Folder...**

IN THIS ARTICLE

- [Setting up VS Code for Java development](#)
- Installing and setting up a Java Development Kit (JDK)
- Creating a source code file
- Editing source code
- Running and debugging your program
- More features

[Subscribe](#)

[Ask questions](#)

[Follow @code](#)

[Request features](#)

[Report issues](#)

[Watch videos](#)

Installing extensions (2)

Getting Started with Java in Visual Studio Code

code.visualstudio.com/docs/java/java-tutorial

Visual Studio Code

Docs Updates

you will need

Installing extensions

If you are an existing user, you can install extensions which includes the following:

- [Language Support for Java](#)
- [Debugger for Java](#)
- [Test Runner for Java](#)
- [Maven for Java](#)
- [Project Manager for Java](#)
- [Visual Studio IntelliCode](#)

[Install the Extension Pack for Java](#)

The [Extension Pack for Java](#) provides a Quick Start guide and tips for code editing and debugging. It also has a FAQ that answers some frequently asked questions. Use the command **Java: Tips for Beginners** from the Command Palette (**Ctrl+Shift+P**) to launch the guide.

Tips for Beginners

[Quick Start](#) [Code Editing](#) [Debugging](#) [FAQ](#)

In this 1-minute tutorial, we'll show you how to create a quick-start Java program in VS Code.

Setup the Workspace

VS Code Java works directly with **folders** that have source code. To setup the workspace, simply open a folder using **File > Open Folder...**

Create a Class

[Link for Java](#)

IN THIS ARTICLE

- [Setting up VS Code for Java development](#)
- Installing and setting up a Java Development Kit (JDK)
- Creating a source code file
- Editing source code
- Running and debugging your program
- More features

[Subscribe](#)

[Ask questions](#)

[Follow @code](#)

[Request features](#)

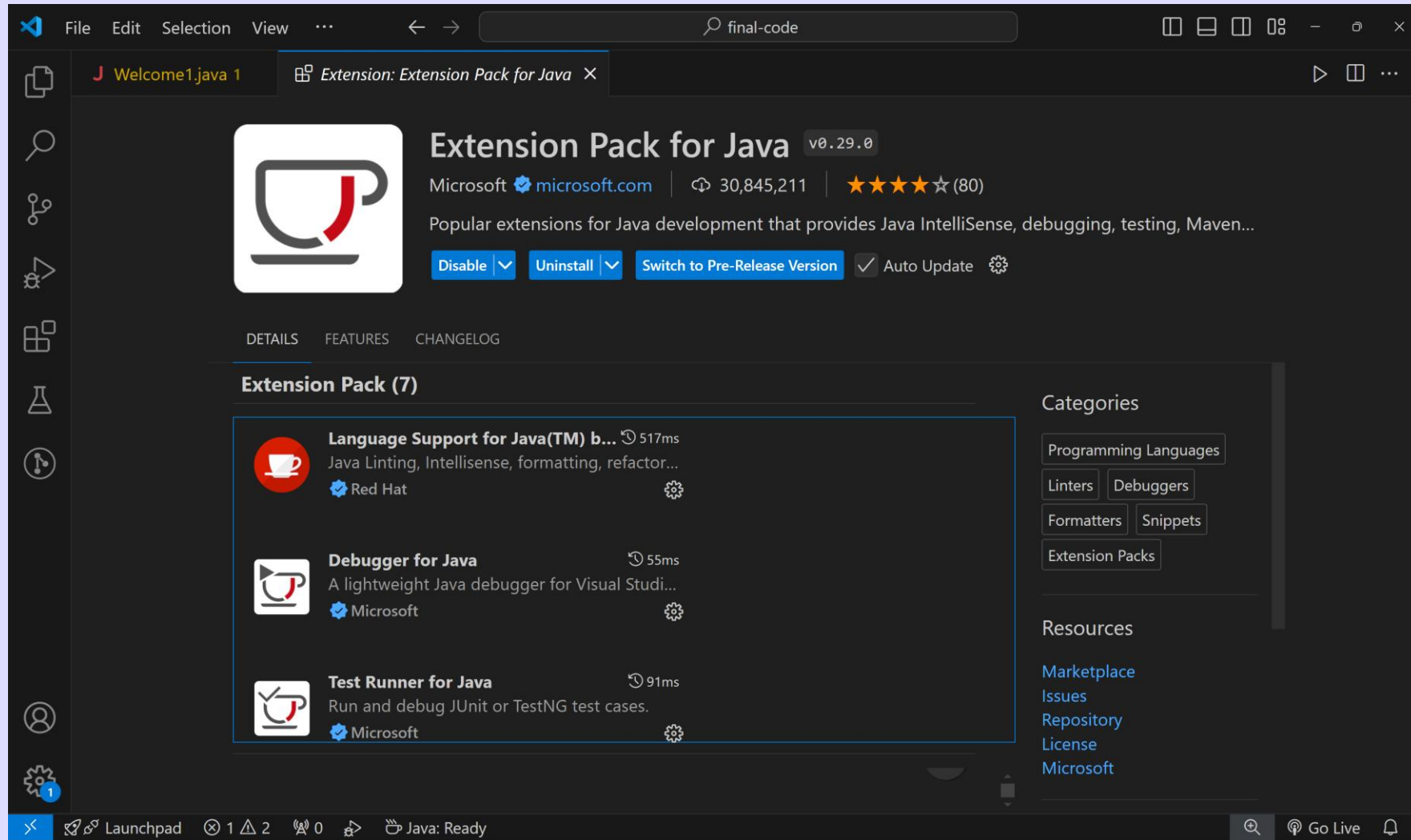
[Report issues](#)

[Watch videos](#)

[Download](#)

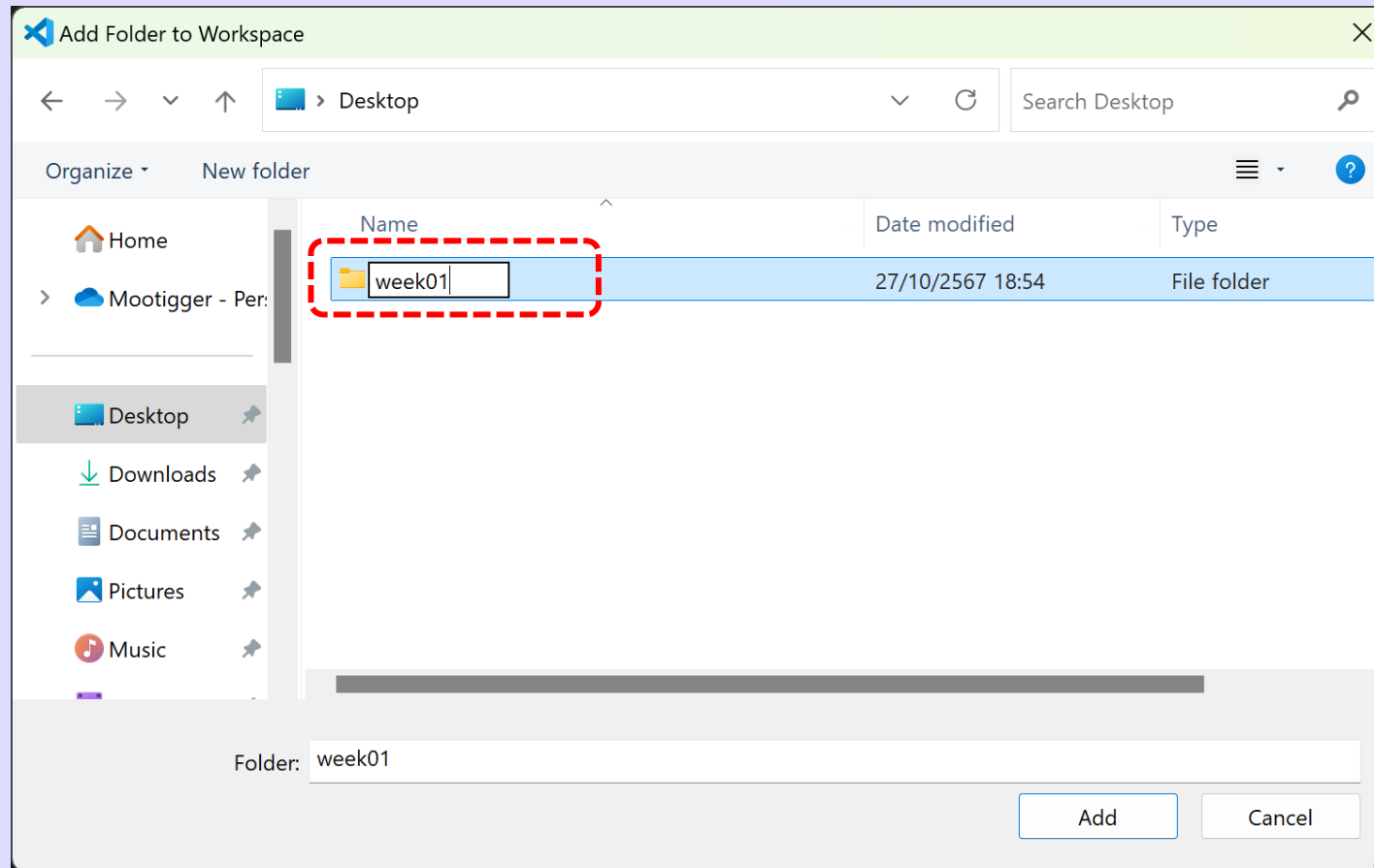
[Open Visual Studio Code](#) [Cancel](#)

Installing extensions (3)



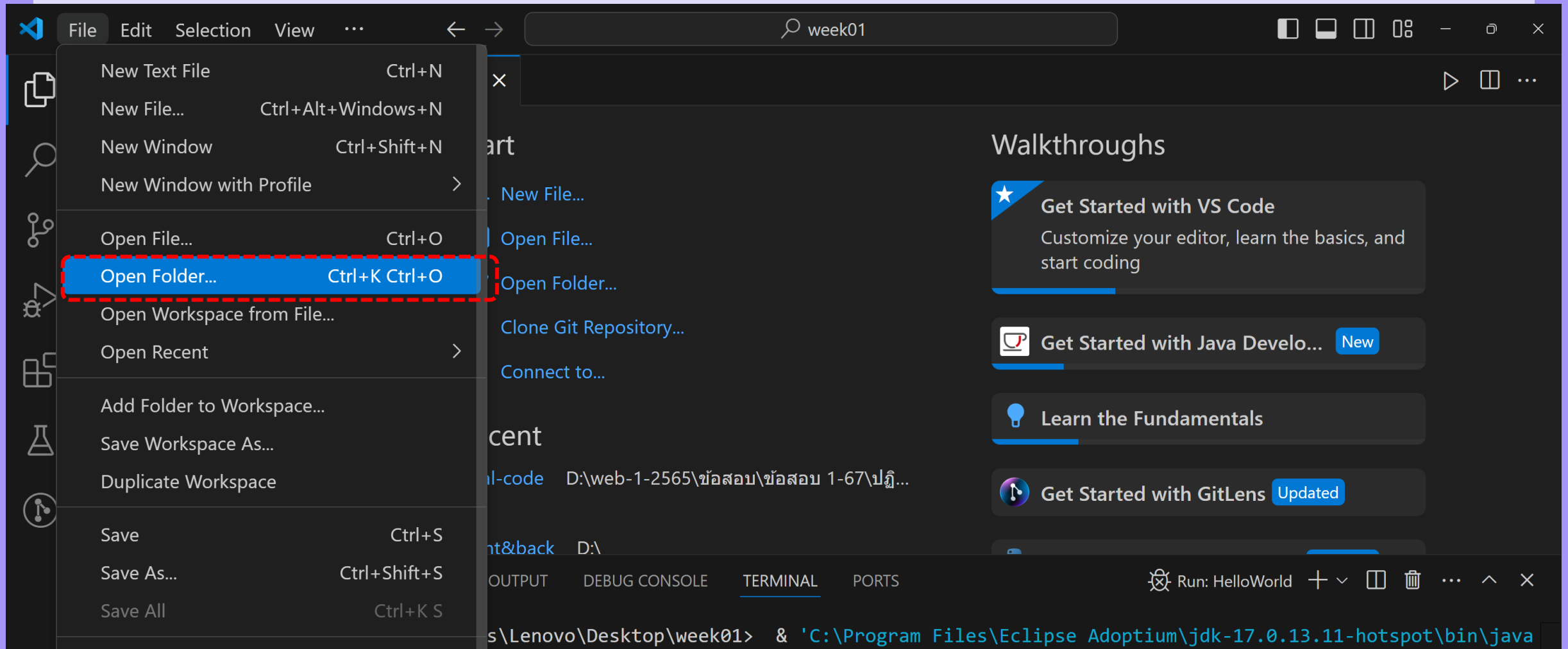
Creating a source code file (1)

Create a folder to save .java files.



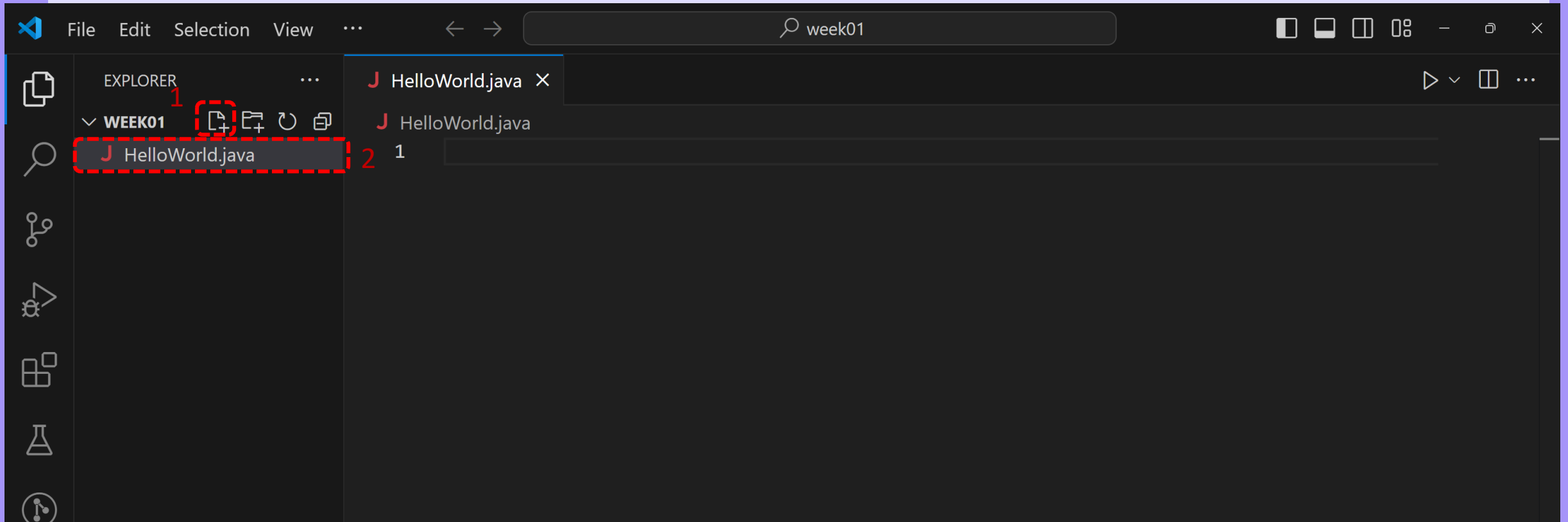
Creating a source code file (1)

- File->Open Folder...->select the folder you created before (week01)



Creating a source code file (3)

- create a new file and save it with the name HelloWorld.java.



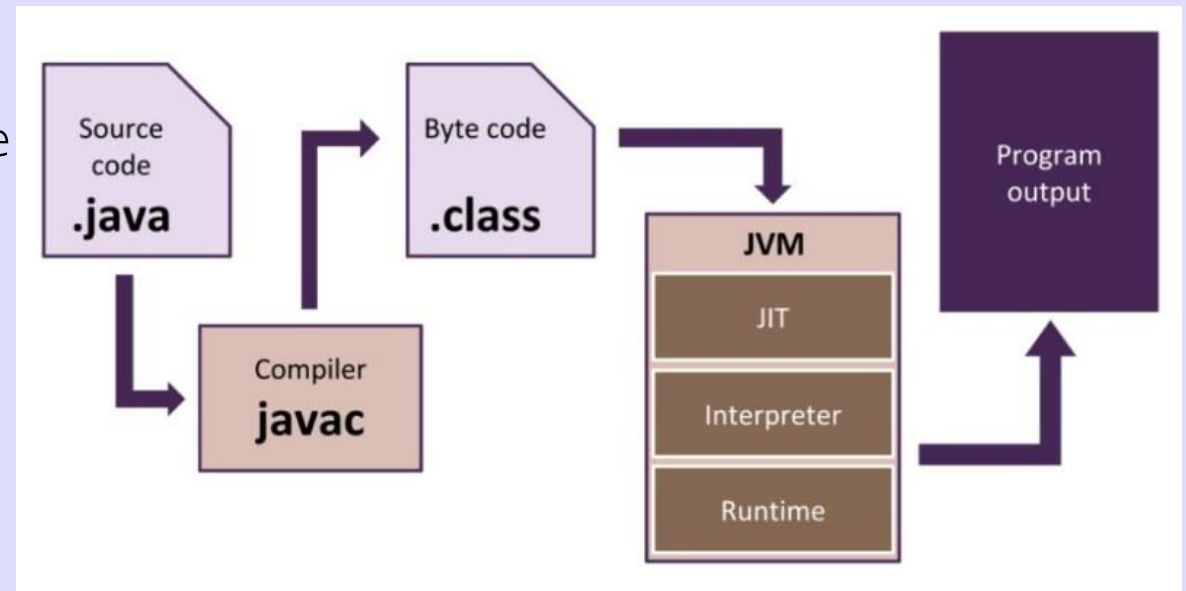
Creating a source code file (4)

- Input a source code in HelloWorld.java.

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Welcome to Java Programming!");  
    }  
}
```

Run .java

- type "**javac filename.java**": This will compile your code. If there are no errors in the code, the command prompt will take you to the next line.
- If there are no errors, It converting the code written in **filename.java** into bytecode (a **filename.class** file will be generated), which the JVM (Java Virtual Machine) can execute.
- The bytecode is saved in a **.class** file.
- Now, type "**java filename.java**" to run the file



Run .java

The screenshot shows the Visual Studio Code interface with the following components:

- EXPLORER:** Shows a project named **WEEK01** containing **HelloWorld.class** and **HelloWorld.java**.
- EDITOR:** Displays the **HelloWorld.java** file with the following code:

```
1 public class HelloWorld {  
2     Run | Debug  
3     public static void main(String[] args){  
4         System.out.println(x:"Welcome to Java Programming!");  
5     }  
}
```
- TERMINAL:** Shows the execution of the program in a PowerShell window:

```
PS C:\Users\Lenovo\Desktop\week01> javac HelloWorld.java  
PS C:\Users\Lenovo\Desktop\week01> java HelloWorld.java  
Welcome to Java Programming!  
PS C:\Users\Lenovo\Desktop\week01> |
```

The commands **javac HelloWorld.java** and **java HelloWorld.java** are highlighted with red dashed boxes.

Run .java

The screenshot shows an IDE interface with the following components:

- EXPLORER:** Shows a project named **WEEK01** containing a file **HelloWorld.java**.
- Editor:** Displays the code for **HelloWorld.java**. The code is:

```
1 public class HelloWorld {  
2     public static void main(String[] args){  
3         System.out.println(x:"Welcome to Java Programming!");  
4     }  
5 }
```

A red box highlights the **Run** button (a play icon) next to line 2. A red arrow points from this button to the terminal.
- Terminal:** Shows the command prompt output:

```
PS C:\Users\Lenovo\Desktop\week01> & 'C:\Program Files\Eclipse Adoptium\jdk-17.0.13.11-hotspot\bin\java  
.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Lenovo\AppData\Roaming\Code\User\workspa  
ceStorage\66b5db44a151f561fff1877a78830c6d\redhat.java\jdt_ws\week01_f6e31805\bin' 'HelloWorld'  
Welcome to Java Programming!  
PS C:\Users\Lenovo\Desktop\week01>
```

Declaring a Class & Class Body

Class Declaration

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Welcome to Java Programming!");  
    }  
}
```

Class Body

- `public class HelloWorld` begins a class declaration for class HelloWorld.
- Every Java program consists of at least one class that you (the programmer) define.
- The `class` keyword introduces a class declaration and is immediately followed by the class name (HelloWorld).
- A `public class` must be placed in a file that has a filename of the form `ClassName.java`, so class HelloWorld is stored in the file HelloWorld.java
- `Class names` begin with a `capital letter` and capitalize the `first letter of each word` they include (e.g., SampleClassName).
- A class name is an identifier—a series of characters consisting of letters, digits, underscores (`_`) and dollar signs (`$`) that does not begin with a digit and does not contain spaces.
- Class Body, A left brace, `{`, begins the `body` of every `class declaration`. A corresponding right brace, `}`, must end each class declaration.

Declaring a Method

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Welcome to Java Programming!");  
    }  
}
```

Method Declaration

method body

- `public static void main(String[] args)` is the starting point of every Java application.
- The **parentheses** after the identifier `main` indicate that it's a program building block called a method.
- Java class declarations normally contain one or more methods.
- Keyword `void` indicates that this **method will not return** any information.
- The `String[] args` in parentheses is a **required part of the method main's** declaration
- The left brace begins the body of the method declaration. A corresponding right brace must end it.

Performing Output with `System.out.println`

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Welcome to Java Programming!");  
    }  
}
```

- Method `System.out.print` displays (or prints) a line of text in the command window."
- Method `System.out.println` displays (or prints) a line of text by a new line in the command window."
- The string in the parentheses `"Welcome to Java Programming!"` is the argument to the method.
- including `System.out.println`, the argument `"Welcome to Java Programming!"` in the parentheses and the semicolon (;), is called a statement.
- A method typically contains one or more statements that perform its task.
- Most statements end with a semicolon.

Escape Sequences

Sequence	Description	Example	Output
\n	New Line	System.out.println("Hello\nWorld");	Hello World
\t	Tab	System.out.println("Name:\tJohn");	Name: John
\"	Double Quote	System.out.println("She said \"Hi\".");	She said "Hi".
\'	Single Quote	System.out.println('\A');	'A'
\\	Backslash	System.out.println("Path: C:\\dir");	Path: C:\dir
\r	Carriage Return	System.out.print("123\rABC");	ABC
\b	Backspace	System.out.print("Test\bA");	TesA

Exercise

- Create Welcome.java
- Write a Java program for Printing a line of text with multiple statements.
- Display the output of the program as follows:

```
Welcome to  
ENG23 2032 - OBJECT ORIENTED TECHNOLOGY
```

- Compile&Run
 - `javac -d . week01/lecture/Welcome.java`
 - `java week01.lecture.Welcome`