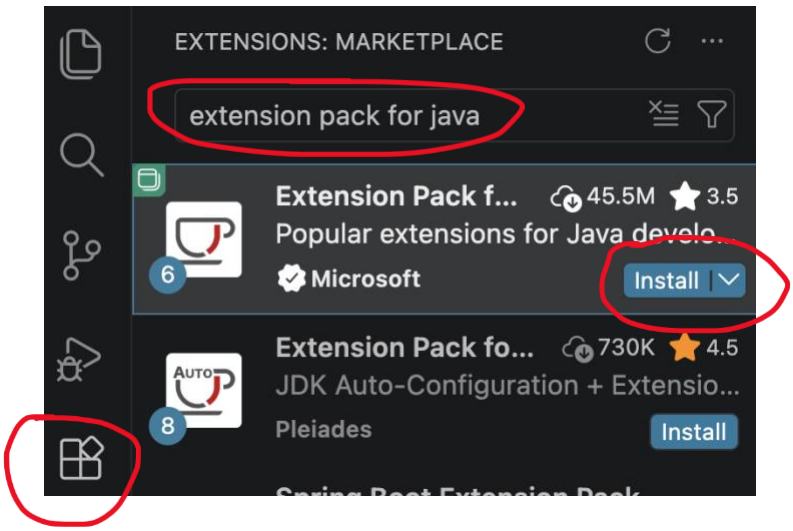
CSC 360, Sofware Development with AI
Course Setup

VS Code and Java Installation

1. Download and install Visual Studio Code (VS Code) from the following site: https://code.visualstudio.com/download

2. Open Visual Studio Code and install the Extension Pack for Java extension from within Visual Studio Code. Extensions can be installed by clicking the Extension icon in the panel on the left. Search for "Extension Pack for Java", and click Install (see red ovals) to install this extension.



3. You will need a JDK installed to run Java using VS Code. Go to View → Command Palette and select Java: Install New JDK. Click Download and follow the prompts to install the JDK. Restart VS Code.

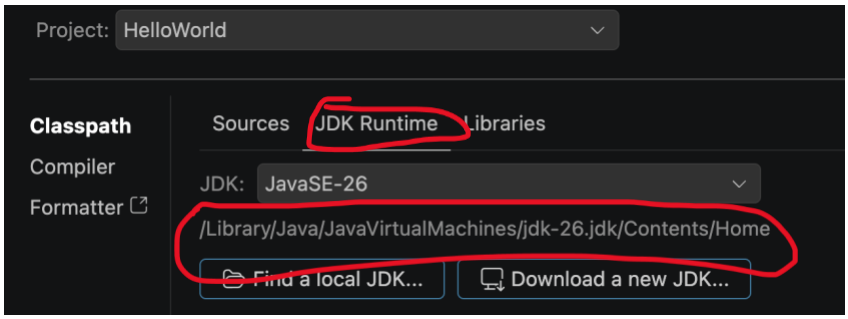
4. Let's test things to make sure Java is correctly installed. Go to View → Command Palette and select Java: Create Java Project. You will have the option to create several types of projects. Select No build tools. Then select your project location.

5. At the top of the screen, you will be prompted for a project name. Type HelloWorld and press Enter. If asked if you Trust the authors, click Yes.

6. If you do not see your files, click on the Explorer icon on the top left of the sidebar. Your project will contain a *src* folder with the file App.java, which contains the program. You can run your code by clicking Run above the *main* method or the arrow on the top right.

Your code will run in a terminal that will open below the code pane. Once the program is finished, you can click the trash can icon to close the terminal.

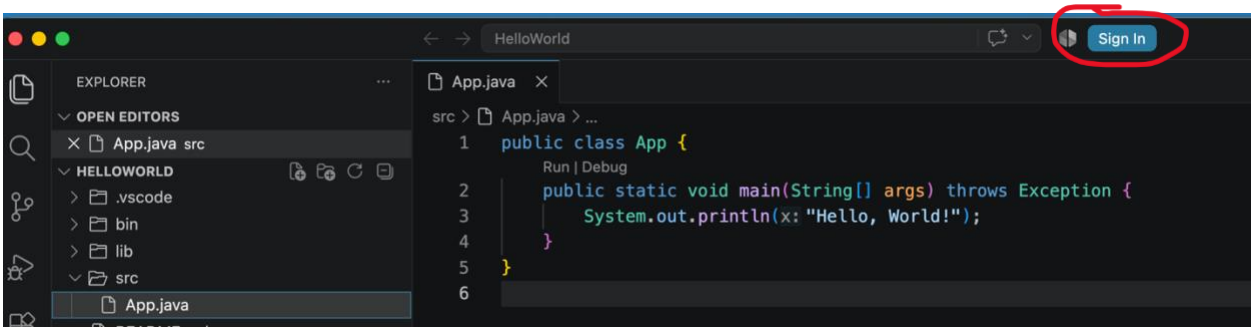
7. Set the environmental variable JAVA_HOME to the JDK installation path, e.g., C:\Program Files\Eclipse Adoptium\jdk-25.0.4.7-hotspot). You can get the JDK installation path by opening App.java in your HelloWorld project, going to View → Command Palette and selecting Java: Configure Java Runtime. An example on my Mac is shown below (your path will likely be different). This will be necessary later in the semester when using JavaFX.



Setup for Coding with AI

1. Install the Github Copilot Chat extension from within VS Code.

2. In order to use AI features you must login using a Github, Google, or an Apple account. After opening VS Code, click the Sign In button the top right, and follow the prompts to sign in. You will only need to do this once.



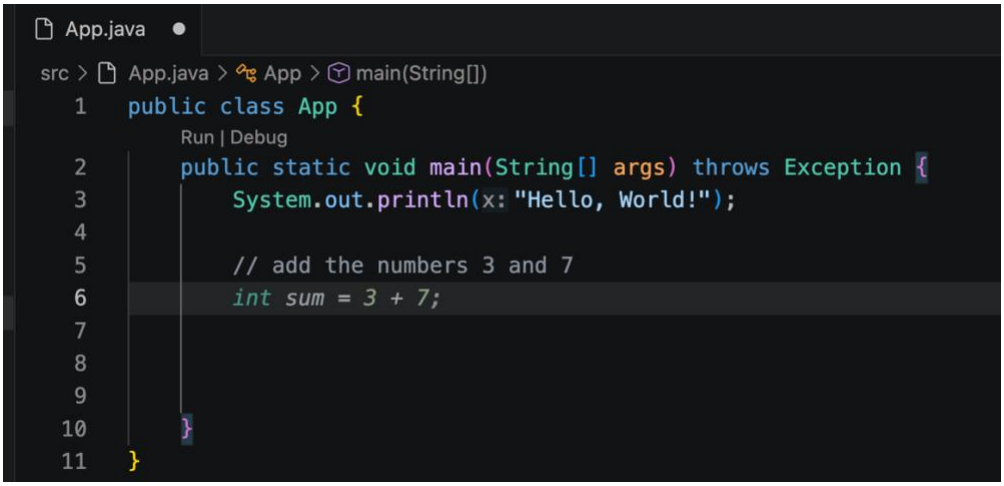
3. To open up a Chat window, select View → Command Palette and select Chat: Open Chat. You may click the X on the top right to close the Chat window. I suggest keeping the chat closed when not actively using it, so that you have more space for writing and running code.

How to use AI to write and understand code

We cover three basic use cases for using AI features to write and understand code.

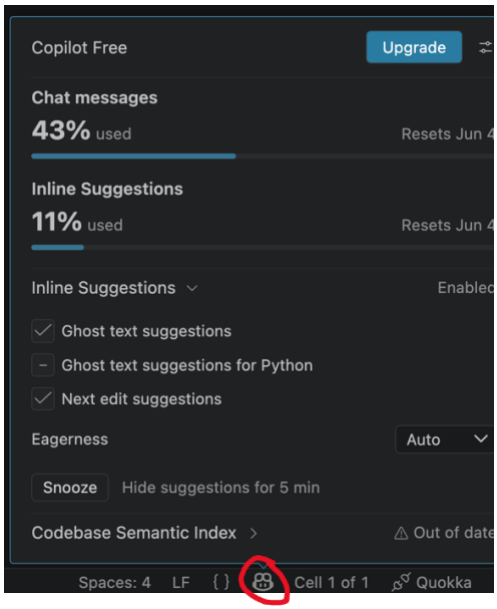
1. ***Use AI to autocomplete code for you***. When writing code AI will suggest autocompletions of code, which will appear in a lighter font (called "ghost text suggestions"). Be careful! Suggested code may not be correct. I recommend writing comments, and then having AI autocomplete the code for you. The AI may suggest one line of code at a time, or multiple lines at a time. When code is suggested, press tab to accept the code; type anything else to discard the suggestion.

   For example, after writing the comment to add the numbers 3 and 7, AI recommended the line `int sum = 3 + 7;`



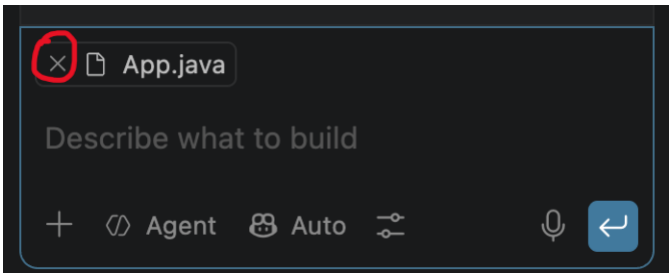
Turning off autocomplete

Sometimes the AI autocomplete suggestions can be annoying and overwhelming. There are also usage limits with the free version of CoPilot (2,000 per month). You can turn these off by clicking on the Copilot icon on the bottom right of the page, and unchecking Ghost text suggestions. This will also let you see your usage.

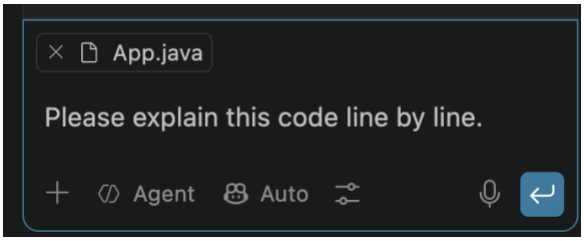


2. ***Use AI to help explain code***. Open the Chat window if it is not already open. Note that you can interact with this chat in the same way you would any chatbot. However, it is designed to help with writing code, with options that allow the chatbot to read files from your project. To specify a file to read, simply click on it, and it will be listed in the chat. Click the plus (+) sign to add the file to the *context*, which the AI will access when responding to your prompt.

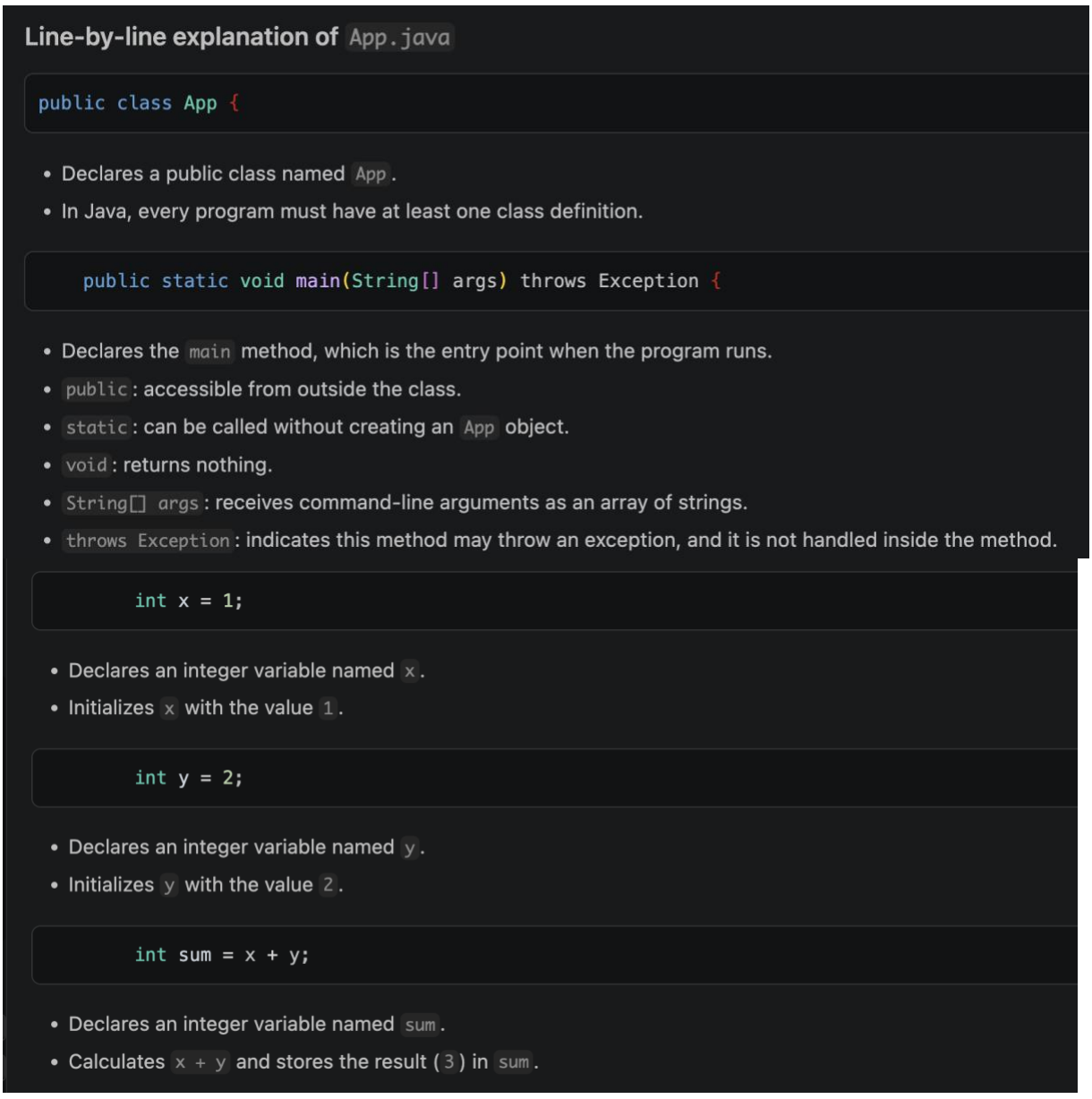
   Once added to the context, there will be an *x* next to the file, instead of a *+*. If you want to remove a file from the context, click on the *x*.



You can now ask the AI questions about your code. Click the blue up arrow or hit enter to submit your prompt. I recommend asking for a line-by-line explanation of the code so that you can test your understanding.



Screen shots of the explanation for this code is shown below:



Line-by-line explanation of `App.java`

```
public class App {
```

- Declares a public class named `App`.
- In Java, every program must have at least one class definition.

```
    public static void main(String[] args) throws Exception {
```

- Declares the `main` method, which is the entry point when the program runs.
- `public`: accessible from outside the class.
- `static`: can be called without creating an `App` object.
- `void`: returns nothing.
- `String[] args`: receives command-line arguments as an array of strings.
- `throws Exception`: indicates this method may throw an exception, and it is not handled inside the method.

```
        int x = 1;
```

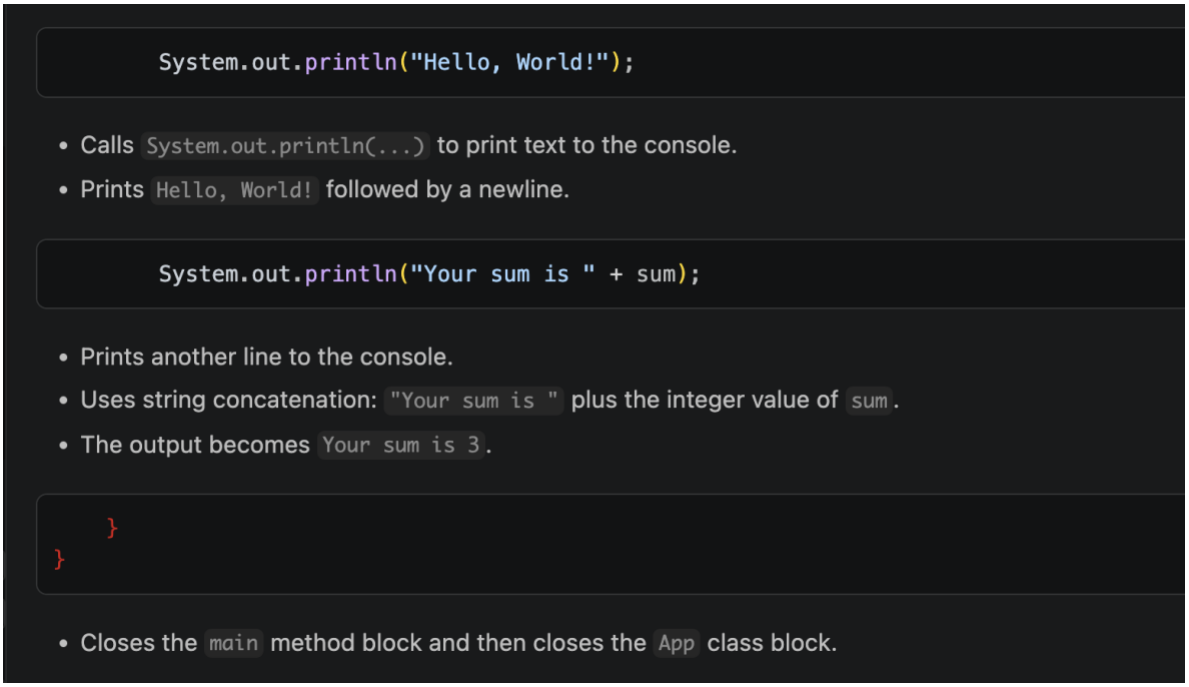
- Declares an integer variable named `x`.
- Initializes `x` with the value `1`.

```
        int y = 2;
```

- Declares an integer variable named `y`.
- Initializes `y` with the value `2`.

```
        int sum = x + y;
```

- Declares an integer variable named `sum`.
- Calculates `x + y` and stores the result (`3`) in `sum`.



Note: because the number of free "credits" for AI prompts is limited, I recommend using other AI tools such as ChatGPT or Claude to explain code.

3. Use AI to write code. Using the AI chat window, you can prompt the AI to generate code for you (e.g., Generate code for a console-based Tic-Tac-Toe game). We will discuss this use of AI to write code in more detail throughout the course.