**From Zero to Code:**

**Getting Started with Java and VS Code**

**Welcome to RUNI’s *Introduction to Computer Science* course!** Three tools that will be used extensively in this course are *Java*, *VS Code*, and *GitHub*. Java is a powerful and widely-used programming language; VS Code is a popular Software Development Environment (IDE) for many programming languages, including Java; GitHub is a cloud-based platform that helps developers store, collaborate, and manage their software versions. This document guides you on how to set up Java and VS Code on your computer; we’ll deal with GitHub later. Because you are a computer science student, we view this document not only as a technical guide but also as part of your computer science education.

**Background**

Java was conceived and developed by the pioneering company Sun Microsystems, which was later acquired by the software giant Oracle. As of 2024, Java is owned by Oracle.

Following Java’s tremendous success, Microsoft introduced a competing Java-like language called C#. Microsoft also developed the VS Code IDE. Microsoft made a wise decision to let people use VS Code to develop programs in any language, including Java. This decision was consistent with Microsoft’s vision to build a global developers ecosystem. This is also why Microsoft bought and now owns the *GitHub* platform, which we will discuss and use later in the course.

Java, VS Code, and GitHub are freely available and can be used on almost any computer platform. Why are companies willing to invest money and resources in developing give-away software? That’s an interesting question that you are welcome to think about.

**Installing Java**

Java comes in two main configurations: The JRE and the JDK.

**To run Java programs**, you need to install the **JRE (Java Runtime Environment)** on your PC. This package provides thousands of pre-built libraries that support input/output operations, networking, and user interface management. The result is a runtime system that allows users to smoothly run Java applications without needing development tools.

**To develop Java programs**, you need the **JDK (Java Development Kit)**.The JDKincludes the JRE, plus a *Java compiler* and other development tools. As a CS student at RUNI, you will write many Java programs, so you have to install the JDK on your PC.

Start by downloading to your computer the JDK 23 (the latest JDK version) from the [Oracle Java Downloads page](https://www.oracle.com/java/technologies/downloads/#java23).

**Intel-based Mac users**: Select the *MacOS* option and download the *x64 DMG Installer*.

**Apple Silicon (M1/M2/M3) Mac users**: Select the *MacOS* option and download the *ARM64 DMG Installer*.

**Windows Users:** Select the *Windows* option and download the *x64 Installer***.**

Execute the downloaded installer program on your computer and follow the installation steps. This action will install the JRE and the Java compiler on your computer. The Java *compiler* is a program that translates (compiles) Java programs into binary code; The JRE is a runtime system that runs (executes) compiled Java programs on your computer.

Once the JDK is installed on your PC, you're all set. No additional configuration is required.

**Installing VS Code**

Download and install VS Code for your operating system (Windows, macOS, or Linux) from [Microsoft’s Visual Studio Page](https://code.visualstudio.com/).

Locate VS Code in your computer’s applications folder, or use your OS search bar to find it. Launch (start executing) VS Code on your computer.

VS Code is designed for writing and developing programs in many different programming languages. Before using it in our course, customize it to work with Java. To do so, click the sidebar’s *Extensions* icon, search “*Extension Pack for Java*,” and install it. This pack configures VS Code for Java development and includes helpful tools for writing, debugging, and running Java programs. For visual help, refer to Figure 1 in the Appendix section below.

**Java program development: Writing, Compiling, Running**

**0. Assumptions**

We assume that you’ve installed Java, VS Code, and the Java Extension Pack.

In the first few weeks of the course, you will not write Java programs from scratch. Instead, we will give you skeletal programs (תכניות שלדיות) that you will have to complete. Each program, say MyProg, is stored in a separate file, say MyProg.java. Keeping files well organized is essential. In a nutshell, we use folders to keep together programs that have something in common. For example, all the programs that belong to Homework Assignment 1 are given in a folder named HW1. These programs are available for distribution on the course website in Moodle as a single compressed file named HW1.zip.

Create a folder named intro2cs on your PC. Download HW1.zip from the “Week 1” section on the course website and extract it under the intro2cs folder.

Next, start VS Code on your PC. All the operations described below should be performed in VS Code.

**1. Open the relevant folder**

To start working on an existing Java program, open the folder where the program’s code is stored. For example, let’s open the folder intro2cs/HW1/code:

Select *File*>*Open Folder*;

Navigate to the intro2cs/HW1/code folder on your computer;

Click *Open* to load the folder files into VS Code.

For visual help, refer to Figure 2 in the Appendix.

**2. Select a file to work on**

Once a folder has been opened in VS Code, the files in the folder will appear in the sidebar’s pane that Microsoft calls *Explorer*.Click on the file you want to work on. For example, if you wish to edit the Demo0 program, click the Demo0.java file name.

For visual help, refer to Figure 3 in the Appendix.

**3. Open a new terminal session**

The terms “*terminal*”, “*command com*”, or “*shell*” are used to describe the same thing on different platforms: An interactive, Unix-like, command-driven program that allows developers to manage files, compile, and run programs on their computers. We use terminal programs to enter textual commands and get textual outputs. The names of these commands are cryptic and telegraphic. For example, if you enter the command “ls” the terminal will respond by listing the files in the current folder. If you type “pwd” you’ll get the *path* of the current folder (in Unix, folders are called directories). There are numerous such Unix-like commands, but for the purpose of this course, you need to know only a few of them.

VS Code has a built-in terminal program, which is accessed as follows:

Click *Terminal* > *New Terminal*

A terminal window will open at the bottom of the VS Code window.

As a sanity check, type the command "java -version" in the terminal and press Enter. If the first line in the terminal output is something like "java version 23", then the JDK was installed successfully on your computer.

For visual help, refer to Figure 4 in the Appendix.

**4. Editing**

To edit an existing file, use the VS Code editor. The VS Code editor provides color-coded syntax, helpful editing hints, and syntax error detection. These features make program editing easier.

To practice editing, locate in the loaded file the statement int i = 0;

Enter, just below this statement, the new statement System.out.println(i);

To save your edits, Click *File* > *Save* (or type Ctrl+S in Windows or ⌘+S in Mac).  
Curiously, VS Code does not feature a traditional file save icon.

**5. Compiling**

Let’s compile the edited Demo0.java program:

In the terminal pane, enter the command "javac Demo0.java".

This command invokes the Java compiler (a program named javac) and applies it to the file Demo0.java. If the file includes error-free Java code, the compiler will translate it into lower-level code written in a language called *Bytecode*. The Bytecode will be stored in a file named Demo0.class.

If there are syntax errors, the compiler will print error messages.

For visual help, refer to Figure 5 in the Appendix.

**6. Running**

If your program was compiled successfully, you can now run (execute) it. To do so, enter the terminal command "java Demo0".

This command applies the Java runtime system (a program named Java) to the compiled file Demo0.class. The bytecode in this file will start running, and the user will see on the screen what this program is supposed to do. Curiously, the command "java Demo0.class" results in a runtime error.

For visual help, refer to Figure 6 in the Appendix.

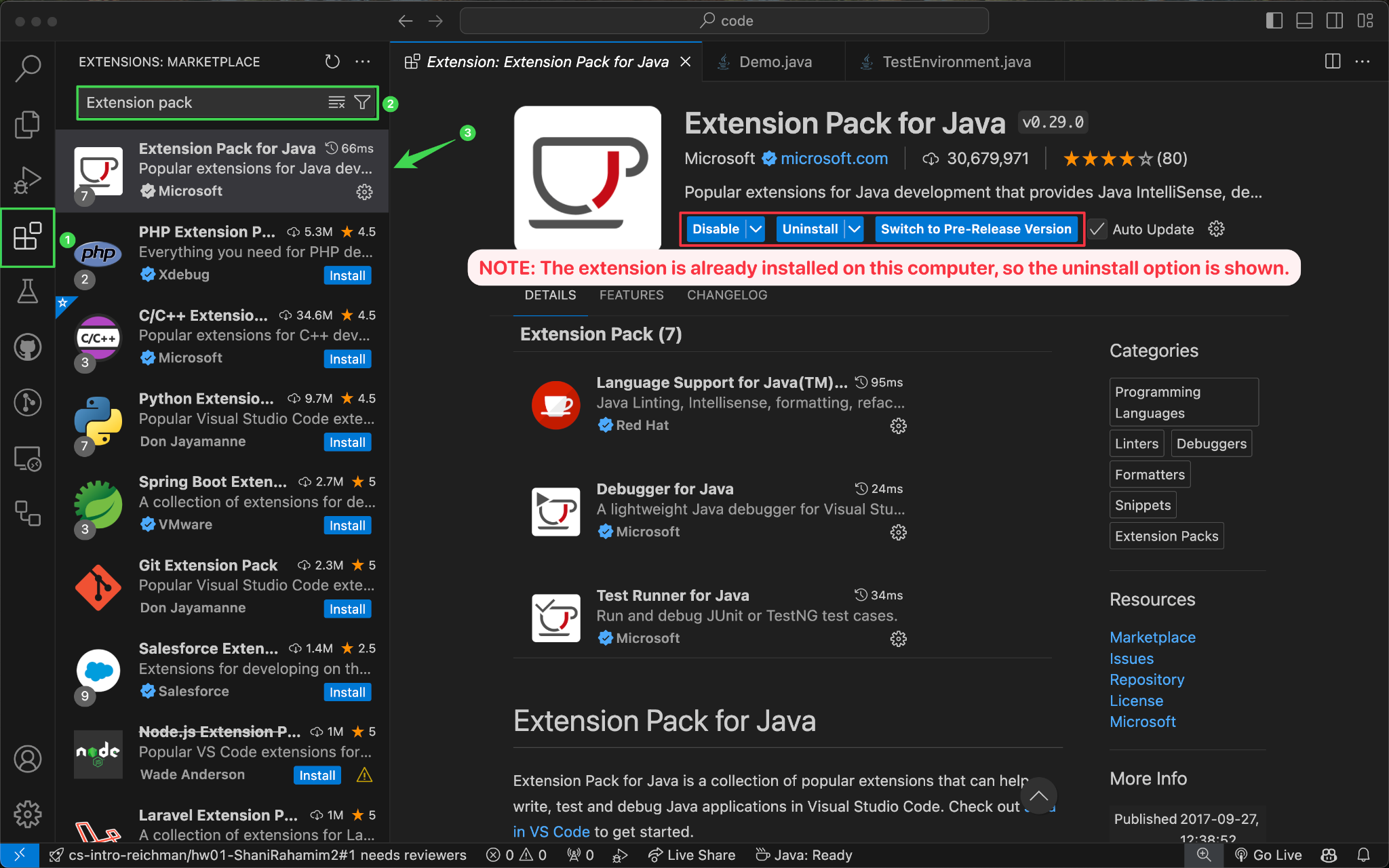
**Did you manage to compile and execute your first Java program?**

If so, Mazal Tov! You made the first step in your software developer journey.

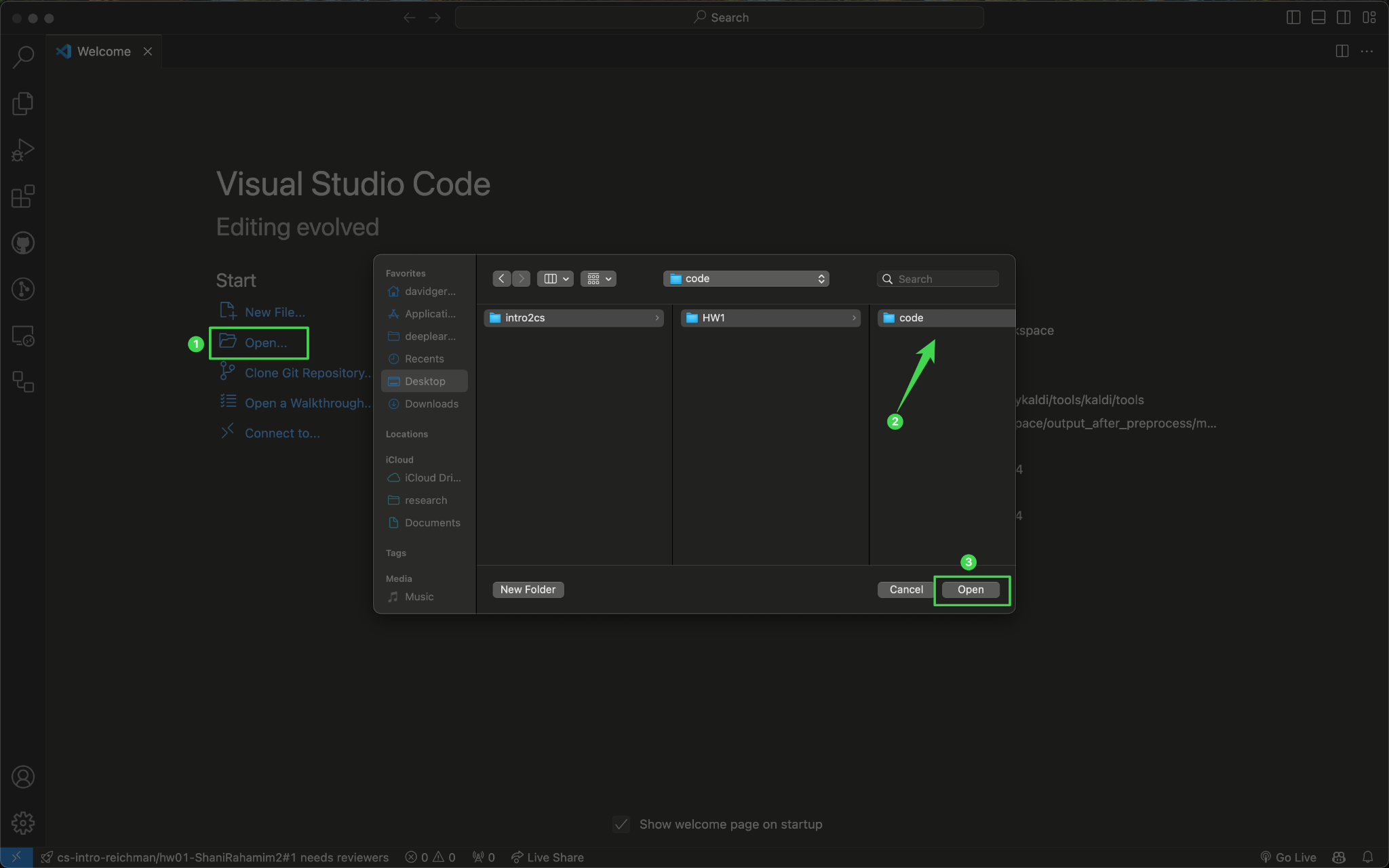
If you encounter errors and problems along the way, don’t worry. You just joined the millions of developers who struggle with software bugs at any given moment on this planet. So, welcome to the club! During the first week of the course, we’ll discuss installation problems and debugging and help you sort them out. If you need assistance, post a question in the Week 1 Q&A forum on the course website or raise it during the first week’s workshop (סדנה). Remember: Troubleshooting is a key part of learning—so don’t hesitate to ask for help!

**Appendix**

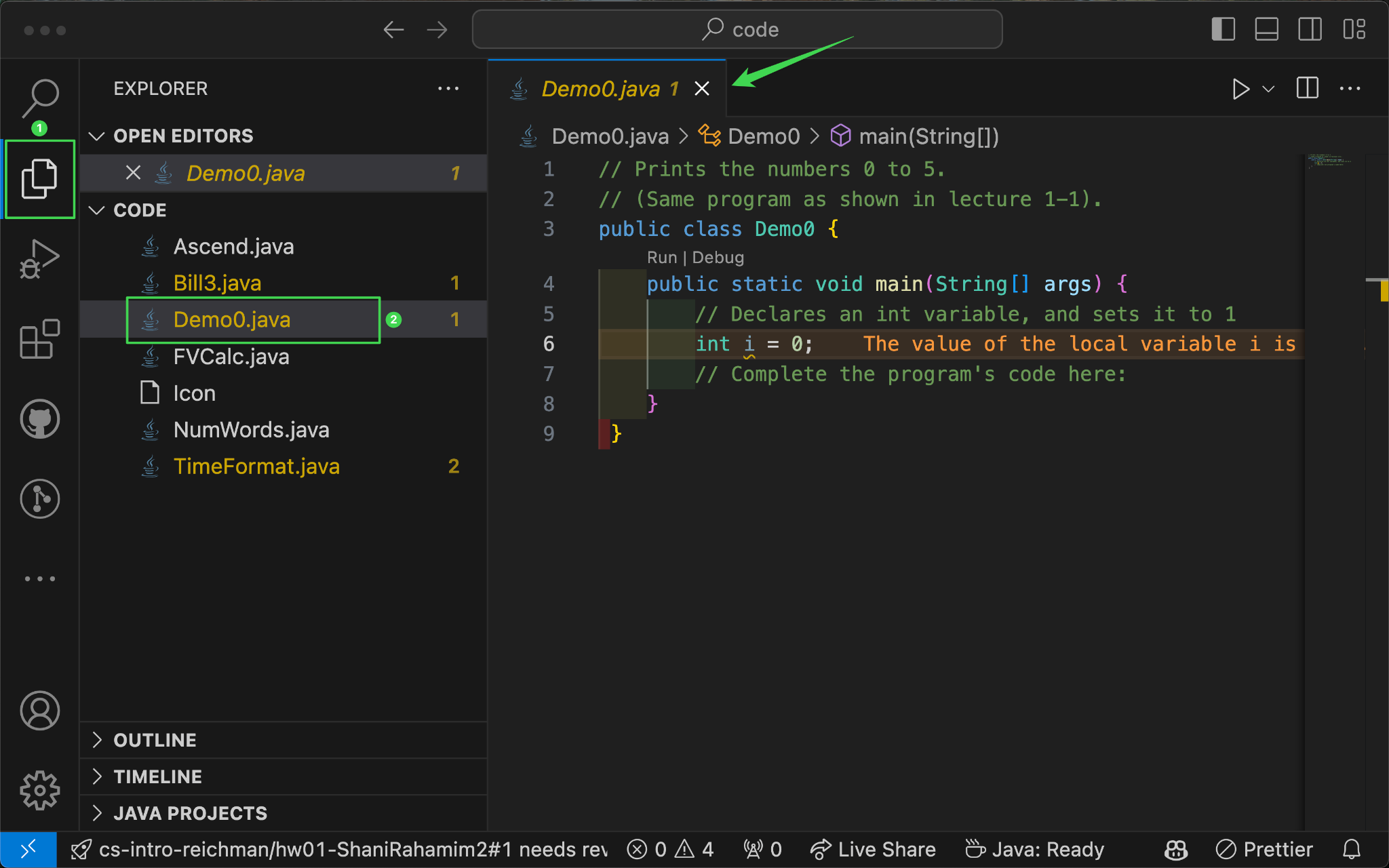
**Figure 1**

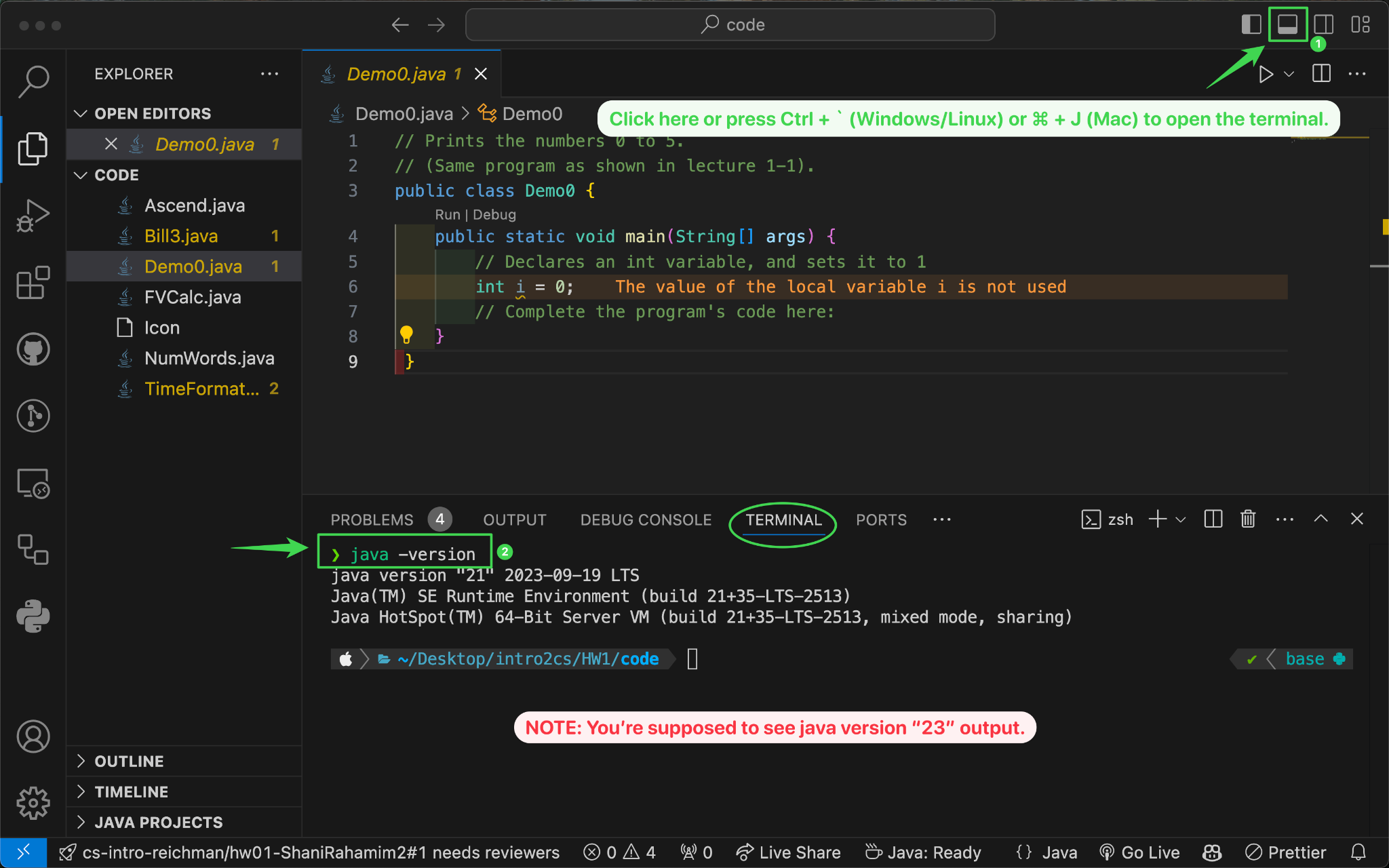


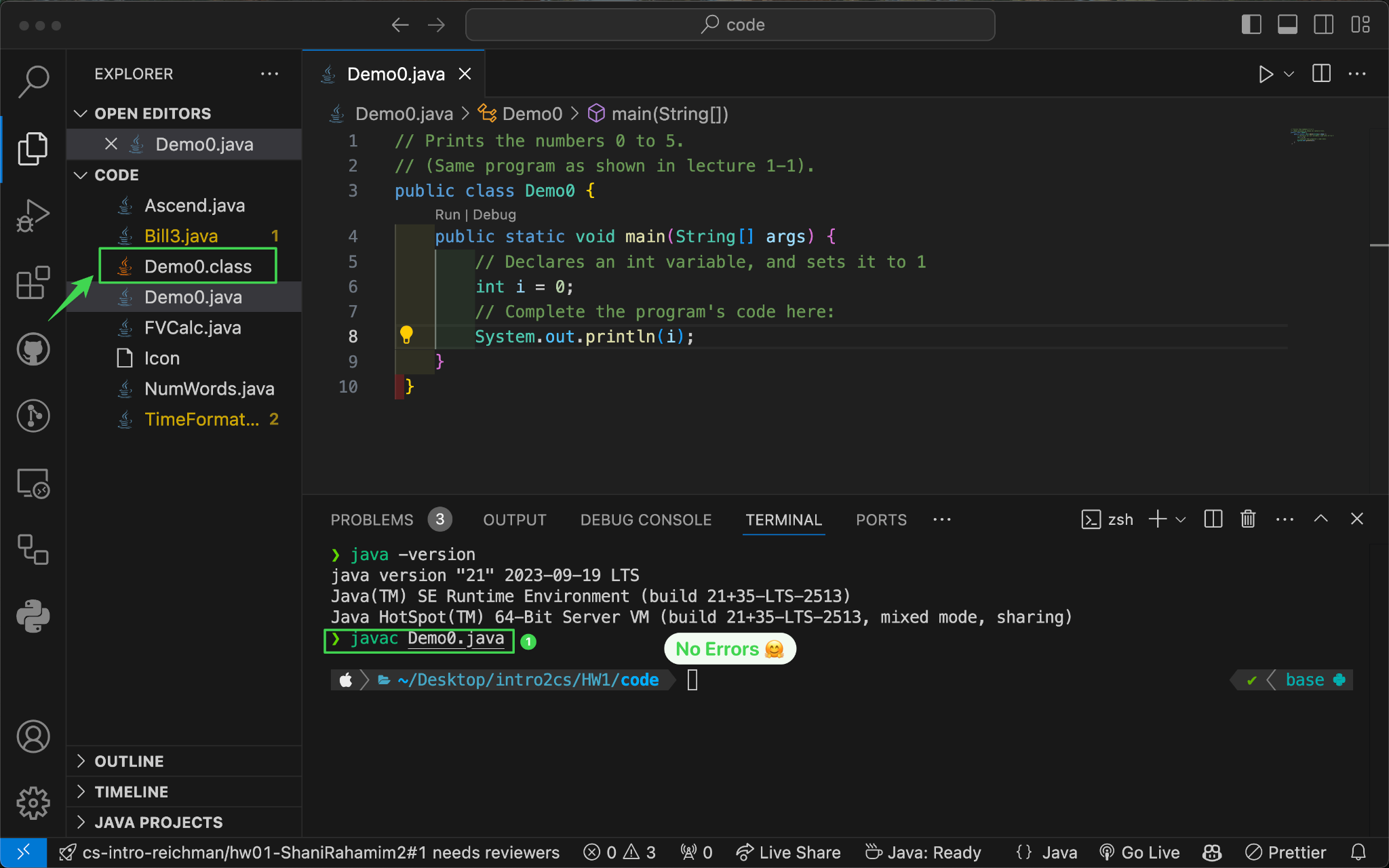
**Figure 2**



**Figure 3**



**Figure 4**

**Figure 5**

**Figure 6**