Lab 1
Using the IT 168 Lab Environment with VS Code

Welcome to your first lab session!

Completion of the lab exercises is very important in this course. The labs give you an opportunity to make sure you understand the course material and they give you a chance to practice new skills before using them to write your larger lecture programs. Therefore, you should take your lab work seriously.

Note:
This set of instructions assumes that you are familiar with the Windows operating systems and can copy files, run programs, etc. If that is not the case, please get individual help from your lab instructor.

Preparing for lab

In most cases, there will be one or more pre-lab exercises: work that you are expected to complete **before** coming to lab. You will submit your pre-lab work via ReggieNet prior to lab time. Late pre-lab work will not be accepted. In some cases, your lab instructor will provide the answers for your pre-lab. In other cases, you will figure out the answers during lab. Be sure to check ReggieNet well before your lab so that you can complete it in good time and be well-prepared. Pre-lab assignments will be made available by the end of the day on Thursday the week before lab.

Connecting to the Java Pool

We have provided a virtual computer for you to do your IT 168 work on. If you wish, you may install Java and VS Code on your laptop and work there outside of lab time. Instructions for that can be found at https://code.visualstudio.com/docs/languages/java. Scroll down to the Install heading and click on the appropriate button, depending on whether you use Windows or MacOS. You can also find a how-to video at https://youtu.be/gQ9qRzocCOc. However, in lab you'll be using the School of Information Technology's Java Pool.

Your computer log-in

You will be using your ULID and password to access the Java Pool and an I: drive to use for 168 work only. ***Do not share it with anyone.***

The I: drive is a portion of a networked hard disk that you can access from virtually any computer on which you can access ADILSTU. If you live off campus, you must use the VPN client to access the ISU network. Detailed instructions are posted on the website in the files: "Working on Your Personal Computer" and "Accessing IT Drives".

The purpose of the I: drive is to give you a place to store your course work that is accessible from multiple places. You must not store anything else on this disk (particularly not multi-media files) of any kind.

- Log in to the computer in front of you using your ULID and password.
- On the left hand side of the window, you will see a Software folder. Open that.
- Open the VMWare Horizon Client
- If there is not a server named vdi.it.ilstu.edu, create a New Server.
    - Enter vdi.it.ilstu.edu for the server
- Choose the vdi.it.ilstu.edu server
- Choose the Java Pool (this is the pool you will always use for IT 168)
- Log in to the virtual Windows desktop that appears using your ULID and password.

Explore a little. If you double-click on This PC, you'll see your I: drive under Network Locations. That is where you will store your work. The only thing that will be saved when you logout is what is on the I: drive.

- Create three folders in your I: drive:
    - work
    - Preferences
    - Lab01

Open a browser and go to ReggieNet. Follow your lab instructor's instructions to open the correct course site; then go to Assignments. Open the Lab01 assignment. Download the AddTwoNumbers.java file to your I: drive in the Lab01 folder you just created. Save the CodeFormatterSettings.xml file to the Preferences folder on your I: drive.

Opening Visual Studio Code (VS Code) in the Lab

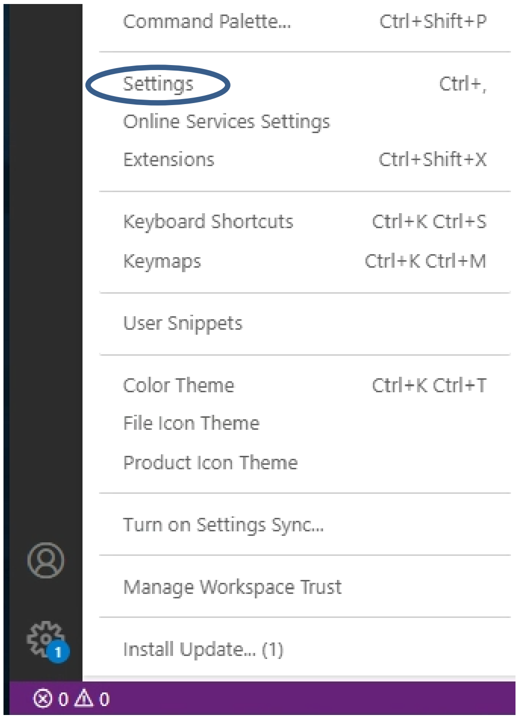
On your desktop in the Java Pool, you will see an icon on the left labeled Visual Studio Code. Double-click on that icon.

When you first start VS Code, you will have a chance to choose a theme. Feel free to use the theme of your choice. The screenshots here will use the Light+ Theme, because that is more appropriate for printing on paper.

Update VS Code Settings

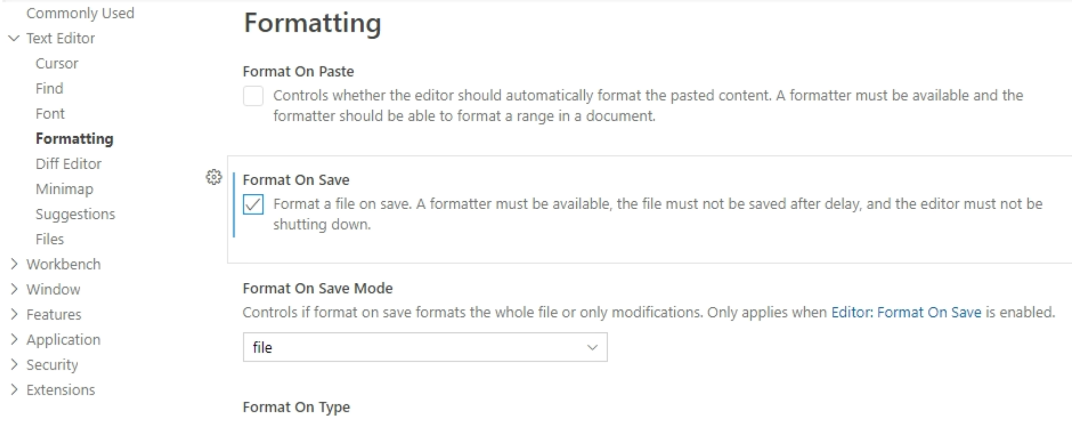
There are some settings you want to adjust in VS Code to save you time later and make sure your code is formatted according to IT 168 expectations. We're going to spend a little time now to set this up in order to save you time later on your lab assignments and lecture programming assignments.

- Click on the gear icon in the lower left of the VS Code window and click on Settings



First, we're going to make sure our code gets formatted whenever we save.

- Expand Text Editor
- Click on Formatting
- Check the Format on Save option



Formatting

Format On Paste

□ Controls whether the editor should automatically format the pasted content. A formatter must be available and the formatter should be able to format a range in a document.

Format On Save

☑ Format a file on save. A formatter must be available, the file must not be saved after delay, and the editor must not be shutting down.

Format On Save Mode

Controls if format on save formats the whole file or only modifications. Only applies when Editor: Format On Save is enabled.

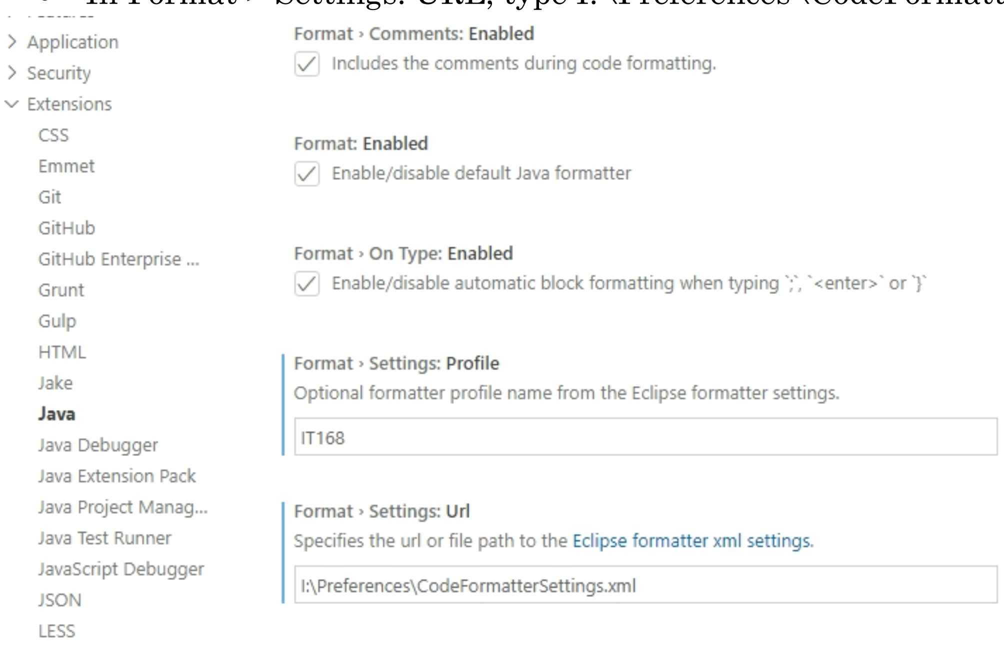
file

Format On Type

Next, we're going to set up the code formatting.

- Expand Extensions
- Select Java
- Scroll down in the settings until you come to the Format settings as shown below.
- Make sure Format>Comments: Enabled is checked
- Make sure Format: Enabled is checked

- Make sure Format> On Type: Enable is checked
- In Format > Settings: Profile, type IT168 (no spaces) as shown below
- In Format > Settings: URL, type I:\Preferences\CodeFormatterSettings.xml



Format › Comments: Enabled

☑ Includes the comments during code formatting.

Format: Enabled

☑ Enable/disable default Java formatter

Format › On Type: Enabled

☑ Enable/disable automatic block formatting when typing `;`, `<enter>` or `}`

Format › Settings: Profile

Optional formatter profile name from the Eclipse formatter settings.

IT168

Format › Settings: Url

Specifies the url or file path to the Eclipse formatter xml settings.

I:\Preferences\CodeFormatterSettings.xml

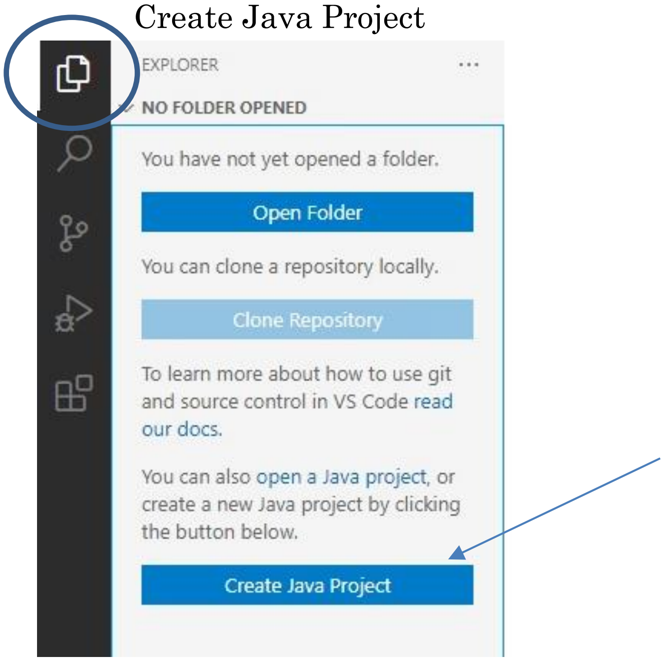
There are a few adjustments that we will now make. First, we want to make sure that we are showing line numbers. That setting doesn't get saved in our preferences file.

Now you can close the Settings tab. Feel free to close the Welcome tab as well.

Create a Java Project

Now we will create our first Java Project.

- Click on the Explorer button in the top left of the VS Code Window, and then click on Create Java Project



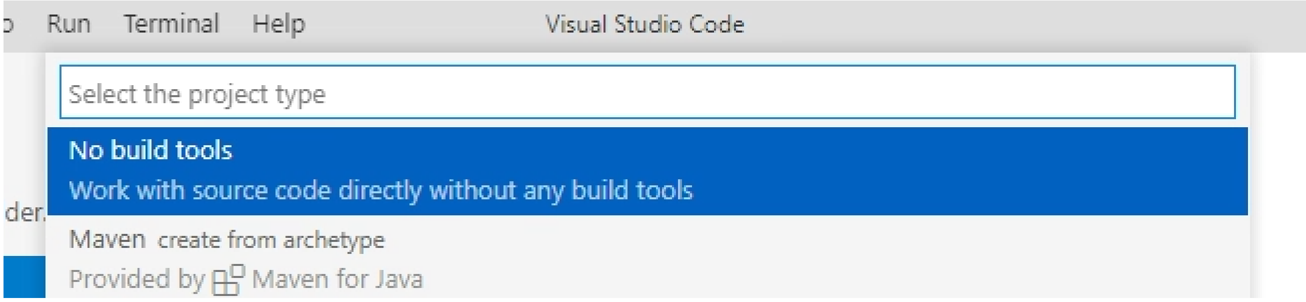
You have not yet opened a folder.

You can clone a repository locally.

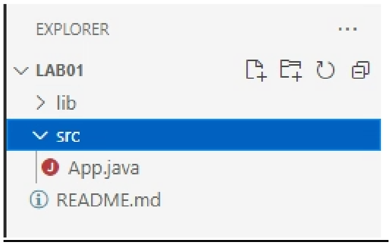
To learn more about how to use git and source control in VS Code read our docs.

You can also open a Java project, or create a new Java project by clicking the button below.

- Choose the No Build Tools option.


Run
Terminal
Help
Visual Studio Code
Select the project type
No build tools
Work with source code directly without any build tools
Maven create from archetype
Provided by Maven for Java
der.

- Browse to your work folder on the I: drive and Select Project Location with the work folder selected.
- Then name the project Lab01 in the textbox near the top of the window.
- Go ahead and trust the authors of the project, since you are the author.
- Location
    - You want to make sure you are aware of where you are storing your project.
    - Your default location should be I:\work
        - On your own computer you have 3 choices
            - Save work on your hard drive
            - Create a workspace on a flash drive to carry between computers.
            - Log in to the ISU network and use your assigned I: drive.
- Click on the src folder in the left-hand panel. You'll see that there is an App.java file there. Right-click on the file and delete it.



Create a Package

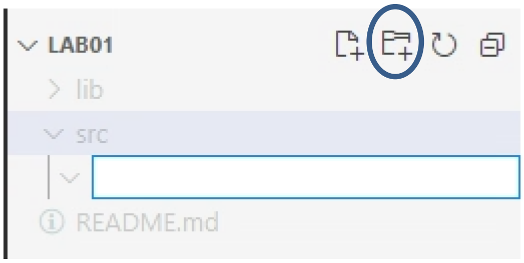
When you have a small project with few files, things are fairly simple. However, when working on a large project with many files, it is a good idea to have a way to organize them. This is done in your workspace by using *packages*, similar to organizing your hard drive with folders.

Advantages:

- Makes it easier to find and use types
- Avoids naming conflicts
- Can be used to control access

To create a package:

- Make sure the *src* folder in your project is highlighted.
- Click on the new folder icon to the right of the project name



- Type edu into the textbox
- With that folder still highlighted create another new folder
- Name this folder `ilstu`

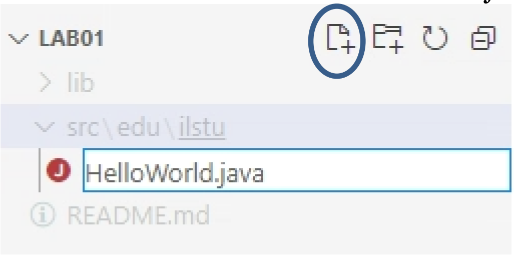
If you do not use a *package* statement, your java files end up in an unnamed (default) package. Generally, an unnamed package is only for small or temporary applications.

Package names are written in all lowercase letters. Convention is for a company to use their reversed Internet domain name to begin their package names. The one we will use is slightly modified, but similar. More qualifiers can be added as needed to distinguish packages from each other.

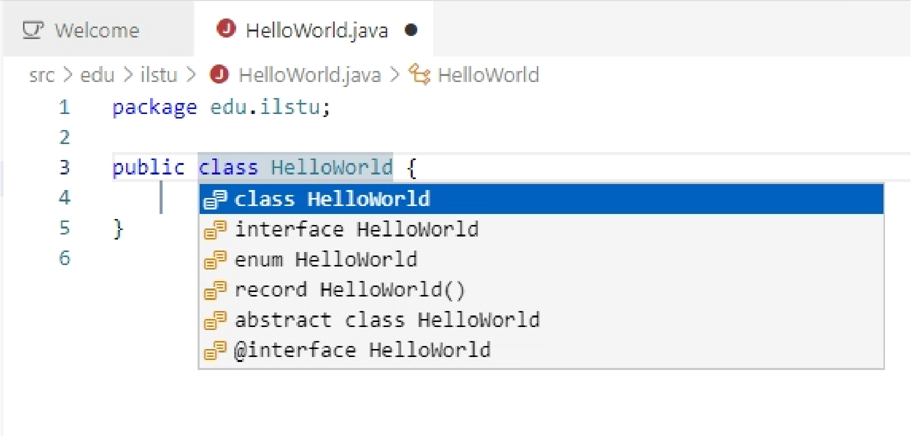
In IT 168, we will use the edu.ilstu package, such as the one you have created in this project.

Create a new Java file in a project

- With the src\edu\ilstu folder highlighted, click on the New File icon (to the left of the New Folder icon.
- Name the file HelloWorld.java (make sure there are no spaces in the name)



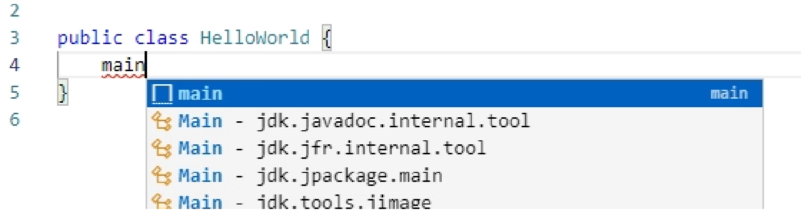
After you hit enter, you should see a new editing tab open that looks like this:



```
package edu.ilstu;

public class HelloWorld {

}

```

Notice that VS Code has automatically added a package statement to the top of your file and is providing a selection of kinds of files you might be wanting to create. In IT 168, we will always be making a class, the first and default option.

- Choose the class option
- Click in the empty line between the curly braces { }
- Type main
- You will see a context menu pop up



```
public class HelloWorld {
    main
}
```

main main
Main - jdk.javadoc.internal.tool
Main - jdk.jfr.internal.tool
Main - jdk.jpackage.main
Main - jdk.tools.jimage

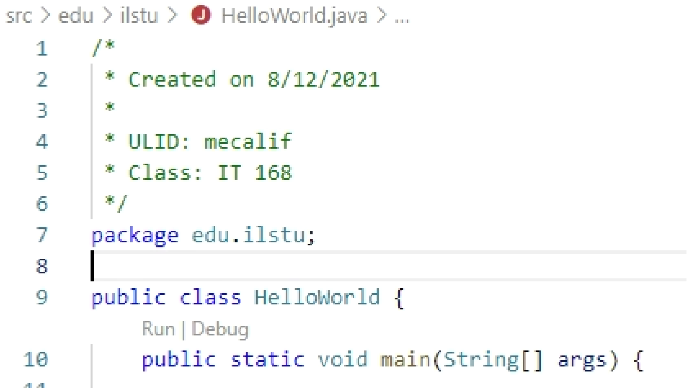
- Select that first option that starts with a square

This creates a special function called main that is the starting point for our programs (and initially where we write all of our code). Don't worry if the line looks a little confusing right now. By the end of the semester, you will know what everything in this line of code means.

Before we write the code to do something, we're going to add comments to our code. For every program you write this semester, you will have at least two comments (later programs will have additional comments).

The first comment will be a regular multi-line comment (enclosed in /* */).

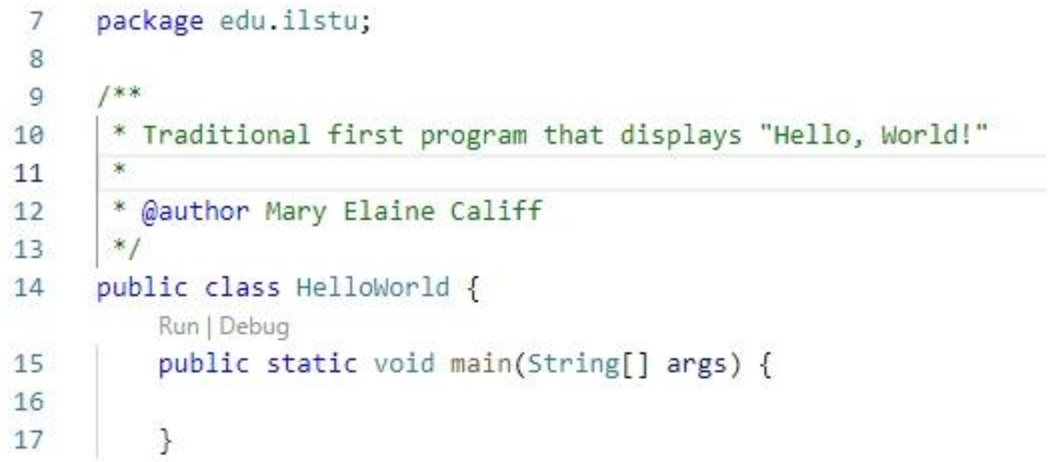
- Type it in the same form as you see in the picture below, but use the current date and your ULID.



```
/*
 * Created on 8/12/2021
 *
 * ULID: mecalif
 * Class: IT 168
 */
package edu.ilstu;

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

The second form will be a class comment that is aJavadoc-style comment (enclosed in /** */). It will have a brief description of the program and then an @author tag with your name (first and last).

- Type in this comment using the description
    - Traditional first program that displays "Hello, world!".
- Make sure your comment looks like the example below, but with your name
- Notice that as you type, VS Code will make suggestions to help you. Make use of them.



```
package edu.ilstu;

/**
 * Traditional first program that displays "Hello, World!"
 *
 * @author Mary Elaine Califf
 */
public class HelloWorld {
    Run | Debug
    public static void main(String[] args) {

    }
```

Be sure to notice the circle in the HelloWorld tab. That means the program has not been saved.

- Save your file. The dot should now be an x.

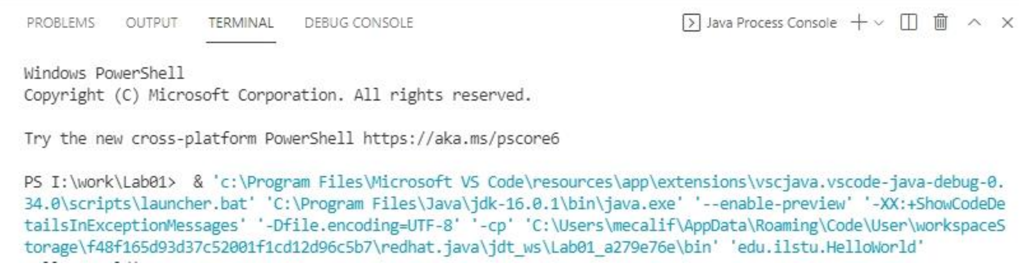
- Now we will write the code for the program:
  - In the blank line under main, type:

    ```
    System.out.println("Hello, world!");
    ```

    You need to be very precise with Java. Type the line exactly as you see it here and don't forget the punctuation!

Running a Java Program

- Immediately above main, you'll see a line with `Run|Debug` . Click on the word Run.
- Cancel the warning from Windows Defender.
- Notice that a new panel has appeared in the lower part of the VS Code window.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS I:\work\Lab01>  & 'c:\Program Files\Microsoft VS Code\resources\app\extensions\vscjava.vscode-java-debug-0.
34.0\scripts\launcher.bat' 'C:\Program Files\Java\jdk-16.0.1\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDe
tailsInExceptionMessages' '-Dfile.encoding=UTF-8' '-cp' 'C:\Users\mecalif\AppData\Roaming\Code\User\workspaceS
torage\f48f165d93d37c52001f1cd12d96c5b7\redhat.java\jdt_ws\Lab01_a279e76e\bin' 'edu.ilstu.HelloWorld'
```

Below the commands to make the program run, you will see the output of the program.

- What is the output displayed inside the Console window?

- After the lab is complete, HelloWorld.java is one of the files that will be submitted.
  - All files for the lab will need to be put into a zip file to be submitted at one time.

Modifying a Java Program

- Make a copy of HelloWorld.java and save it with a new name.
    - Make sure you have the file selected.
    - Choose "Save As . . . " from the File menu
    - Name the file HelloWorld2 and save it into the ilstu folder.

You'll notice that the name of the file is red and there is a red square to the right as well as a red squiggly line under the class name HelloWorld. This is because the class name and the file name do not match, and Java requires that they do.

- Change the name of the class to HelloWorld2.
- Add a new line of code to the existing program after the "Hello World!" statement.
    - Type the following:

        -
          ```
          System.out.println("Welcome to IT 168!");
          ```
- Save your program
- Run the HelloWorld2 program

- The HelloWorld2 file will need to be submitted when the lab is completed.

Import a Java file into a project

Follow this procedure any time you are provided with a file that is to be part of your project.

- In Windows Explorer, open your Lab01 folder on the I: drive.
- Drag the AddTwoNumbers.java file from that folder onto the project folder in VS Code and drop it.

- Open the file and run the program
    - There will be a prompt in the Console window. Follow the directions and hit ENTER to get the result.

- **Modify** the program so that it computes and displays the difference between the first and second numbers as well as the sum of the two numbers. Also modify the class description appropriately and add your name to the @author clause.

Submit the Files using ReggieNet

Step 1: First you need to prepare your files for submission. Java file names must match the name given to the class definition. If the file name is changed, the class will not function. You will be required to zip all your submitted files into a single zip file. **You are to zip the files only. DO NOT zip a folder containing the files as this will make grading more time-consuming and your grade will be reduced for it.**

Follow these steps to zip your lab files:

- Typically you will be submitting multiple files for each lab or program. You are required to submit only one zip file so you must **wait until your entire lab or program is complete to submit.**
- Locate all files to be submitted in your `I:\work\Lab01\src\edu\ilstu` folder. Remember that you are to submit the **.java** file **NOT** the **.class** files found in the `bin` folder for all Java classes. Any **.class** files submitted will not receive credit.
- Copy them to the Lab01 folder you created.
- Select all files to be submitted. You can do this using **Ctrl-click**. The files will be:
    - HelloWorld.java (8 points)
    - HelloWorld2.java (8 points)
    - AddTwoNumbers.java (9 points)

- Right-click on the set of selected files and choose **Send To -> Compressed (zipped) Folder**.
- **Rename your zip file Lab01.zip.**

Step 2: Submit the zip file through ReggieNet to the Lab01 Assignment.

- Open ReggieNet and log in.
- Go to the IT 168 section that has your lab assignments (the same one you used at the beginning of the lab).
- Click **Assignments** button on the left and then open the assignment for lab 1.
- Upload your zipped file and proceed.
- When you are done, click *Submit*.
    - Labs are due by the beginning of the following lab. The assignment should be set up so that you can resubmit freely if needed until the beginning of the next lab.
- After submitting your assignment, you will receive an email notification which serves as a receipt. The email notification includes a Submission ID which can be used to verify that you submitted the assignment.
    - Keep this email until your grade has been posted.