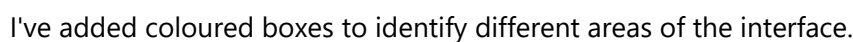


These exercises will get you set up with Visual Studio Code and GitLab.

We will be using Visual Studio Code (VSC or VSCode) to write, edit, debug and run our Java code. VSC is, at its heart, a text editor that can be extended in many different ways. It can be set up to work with many different programming languages by use of *Extensions*. We will be installing Extensions to allow VSCode to work with Java. These extensions will allow you to compile, debug and execute Java code. They will also help you manage Java projects and make it possible to work with GitLab from within VSCode.

Where is it?

The VSCode window will look something like this :-



The menu bar is highlighted with a red box at the top of the window, it has menus.

1 / 13

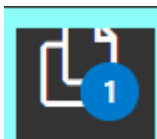
The majority of the window is taken up by the editor window(s). You can have multiple windows simultaneously. By default they appear in tabs, you can drag and drop tabs to tile windows.

Icon Bar/Context Window

The strip down the left hand side of the window (highlighted cyan above) is the Icon Toolbar. It can be used to switch between different modes in the Context Window (highlighted magenta) to the right of the Icon Bar. We will be mostly using three different modes.

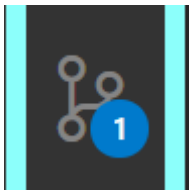
Explorer

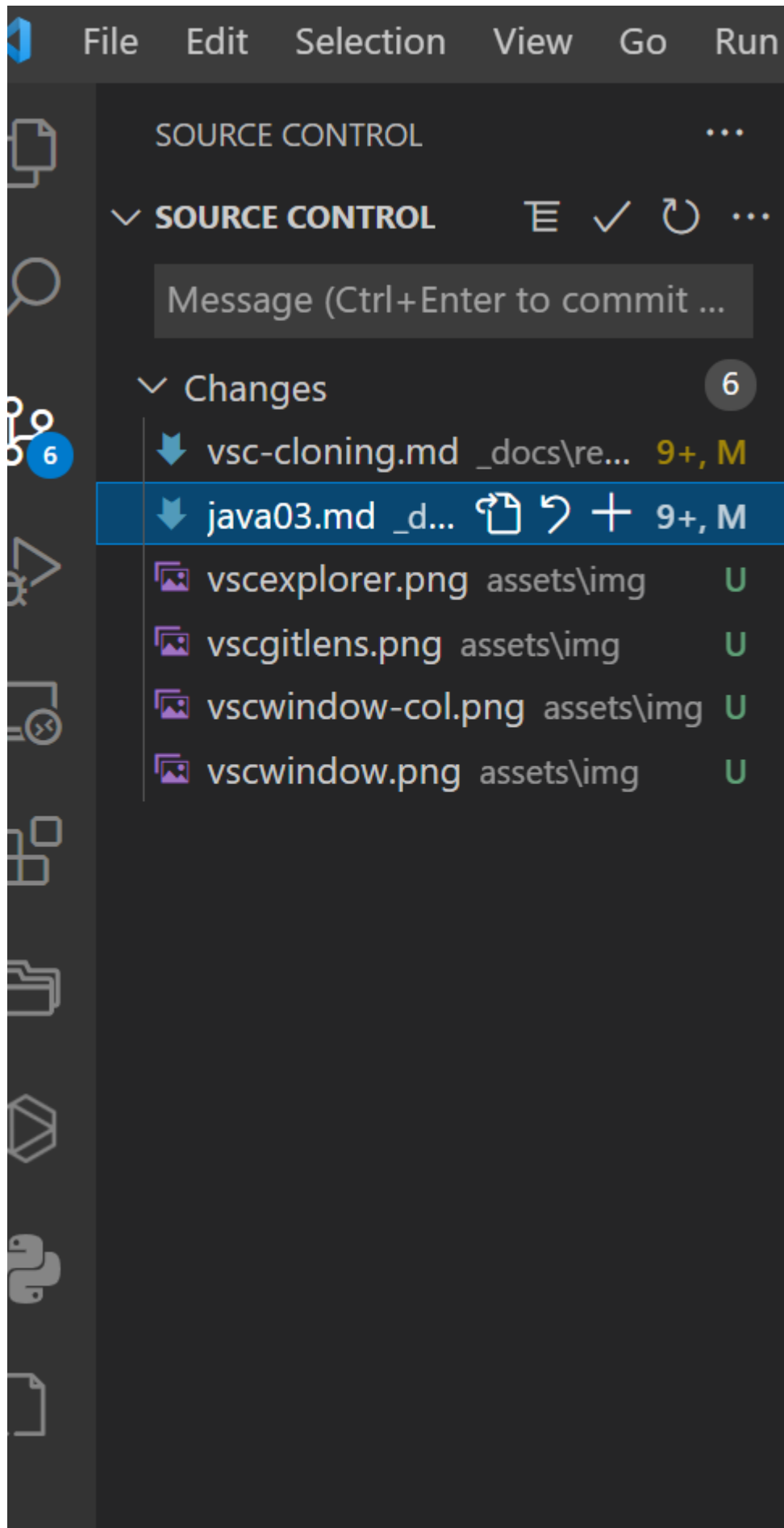
The Explorer icon is used to open the Explorer window in the context window. This is shown in the main image above. The Explorer Window allows you to navigate the folders and files of the current project.



GitLens

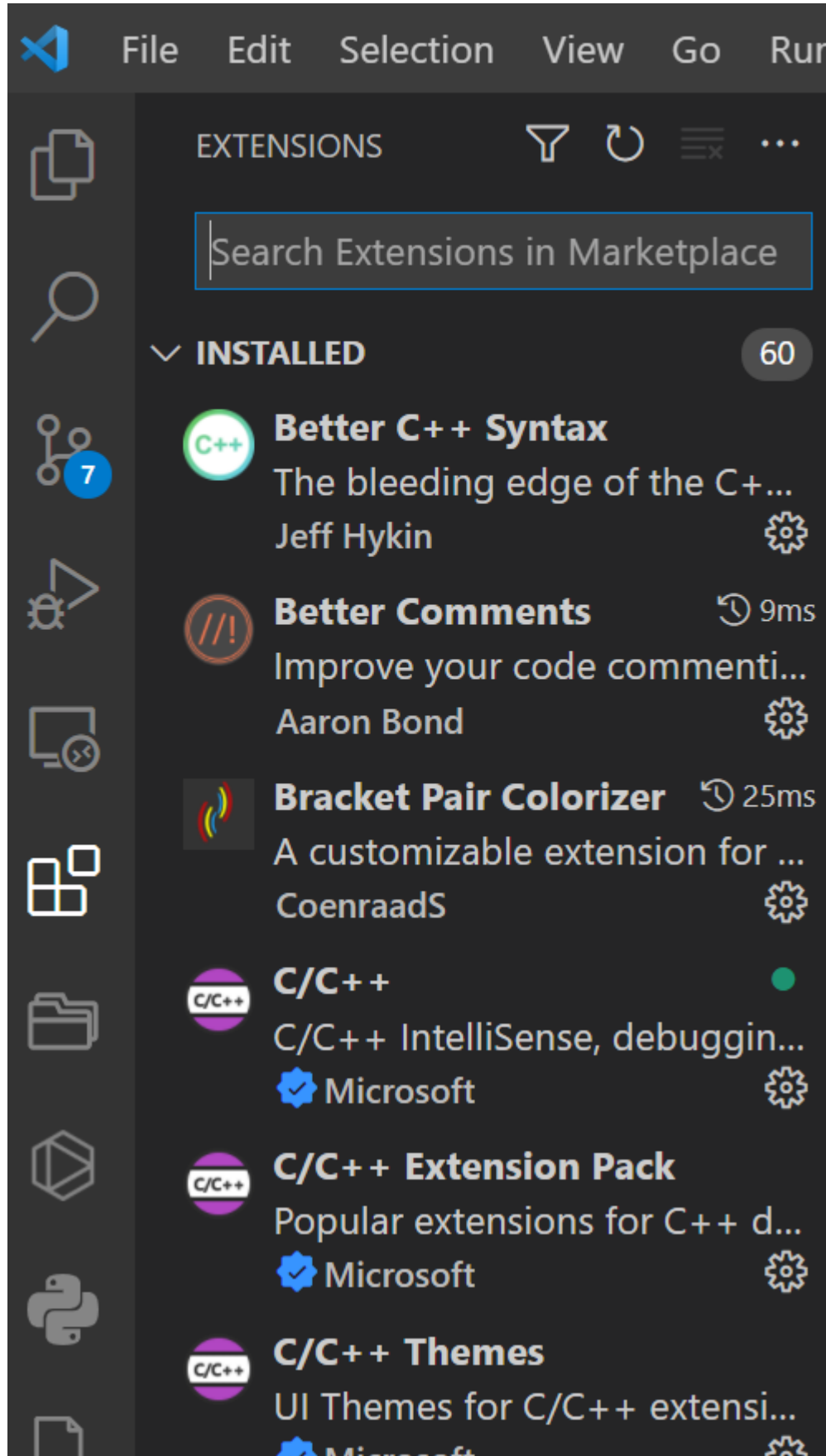
The GitLens icon is used to open the Git window in the context window. This is shown in below. The Git Window allows you to clone (download) git repositories and upload your own code to your git repositories.

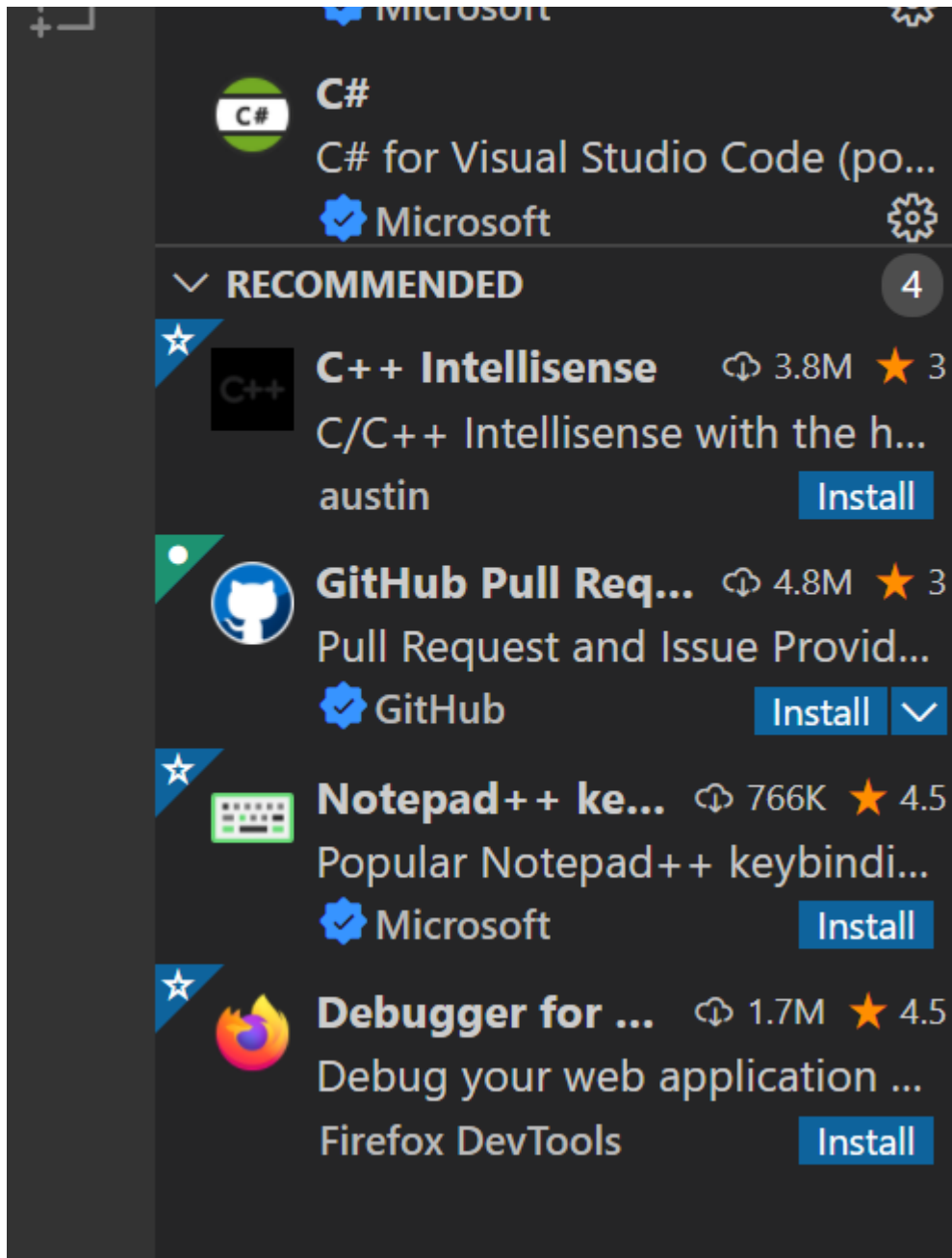




Extensions

The Extensions icon is used to open the Extensions window in the context window. This is shown in below. This window allows us to add and remove extensions, and to change their settings.





This window is divided into three sections. At the top of the window is a search bar that you can use to search for extensions. When you search for an extension the two sections below are replaced by the search results. If you wish to install an extension simply click on the blue **Install** button under the extension. You may need to reload after installing an extension.

Below this is the **Installed** section that shows extensions which have been installed. You can access the settings for these extensions by clicking on the little cog icon next to them.

The lower section shows recommended extensions. These are extensions that VSCode thinks you might find useful. Don't install them all, you don't need them.

Problems/Output/Terminal Window

The window at the bottom of VSCode, below the editor windows, is used for output, debugging and the terminal. If any of these windows is not open/showing in the tabs, you can add them from the *View* menu.

3. Install Extensions

1. Click on the Extension icon in the Icon toolbar.
2. Install the following extensions
 - Extension Pack For Java
 - Test Runner for Java
 - Code Runner
 - PlantUML

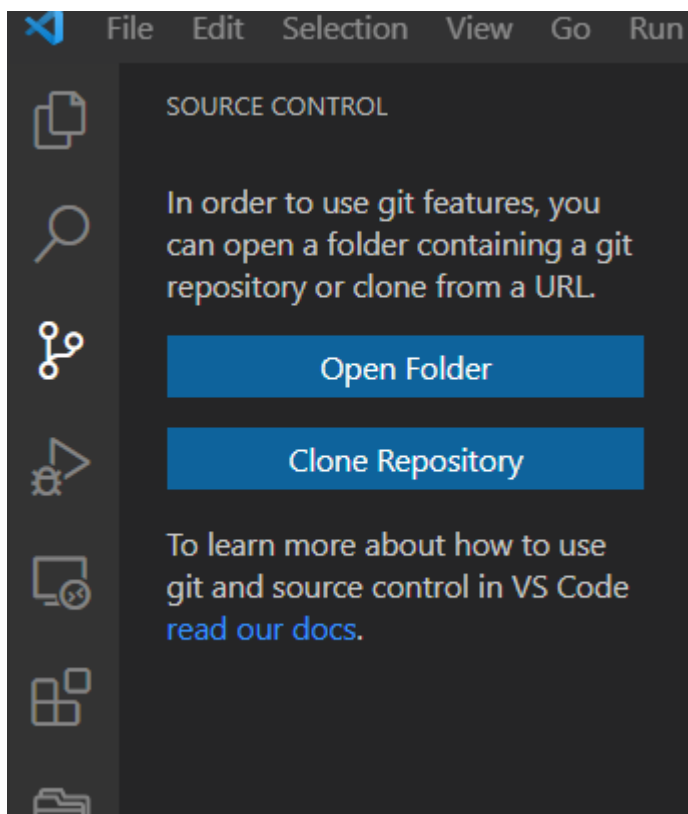
Extension Pack For Java is a collection of extensions that will allow us to work with Java. **Test Runner for Java** will simplify running unit tests. **Code Runner** will make compiling and executing the code easier. **PlantUML** will allow us to make UML diagrams. We will be doing this later in the module.

You may install any extensions you wish, just don't install too many or VSCode performance will suffer.

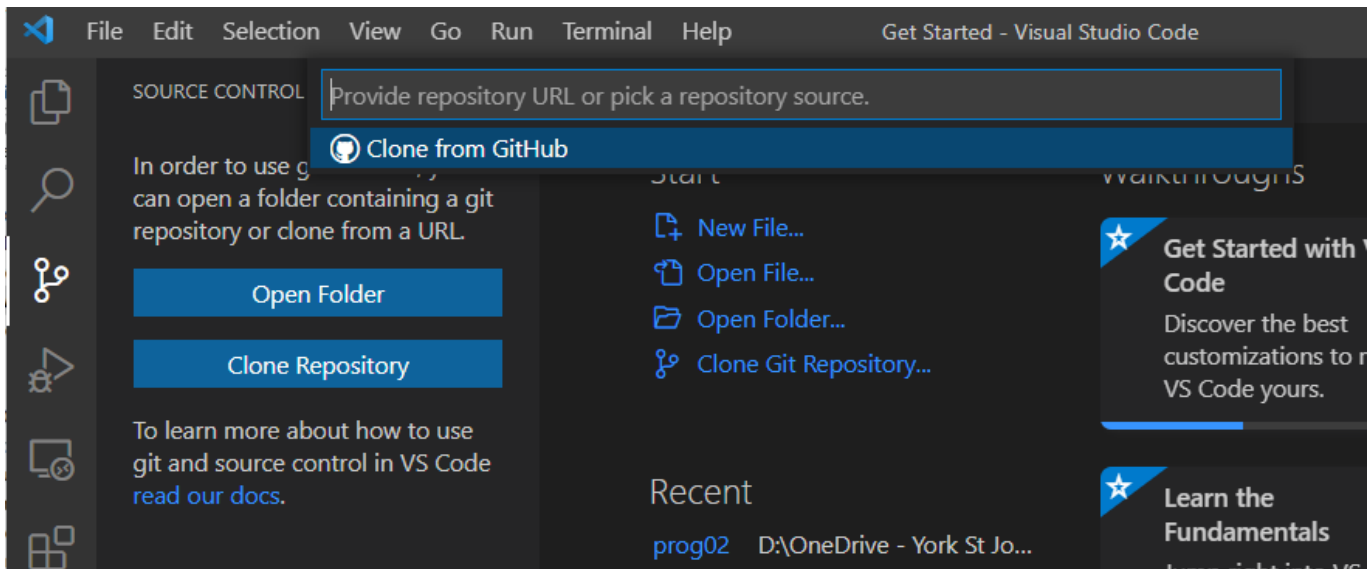
4. Clone JavaHelloWorld Repository

Now we should be able to get some Java running. I've created a Java Hello World application and uploaded it to our GitLab server. You can use VSCode to download a copy of this programme to your machine and run it.

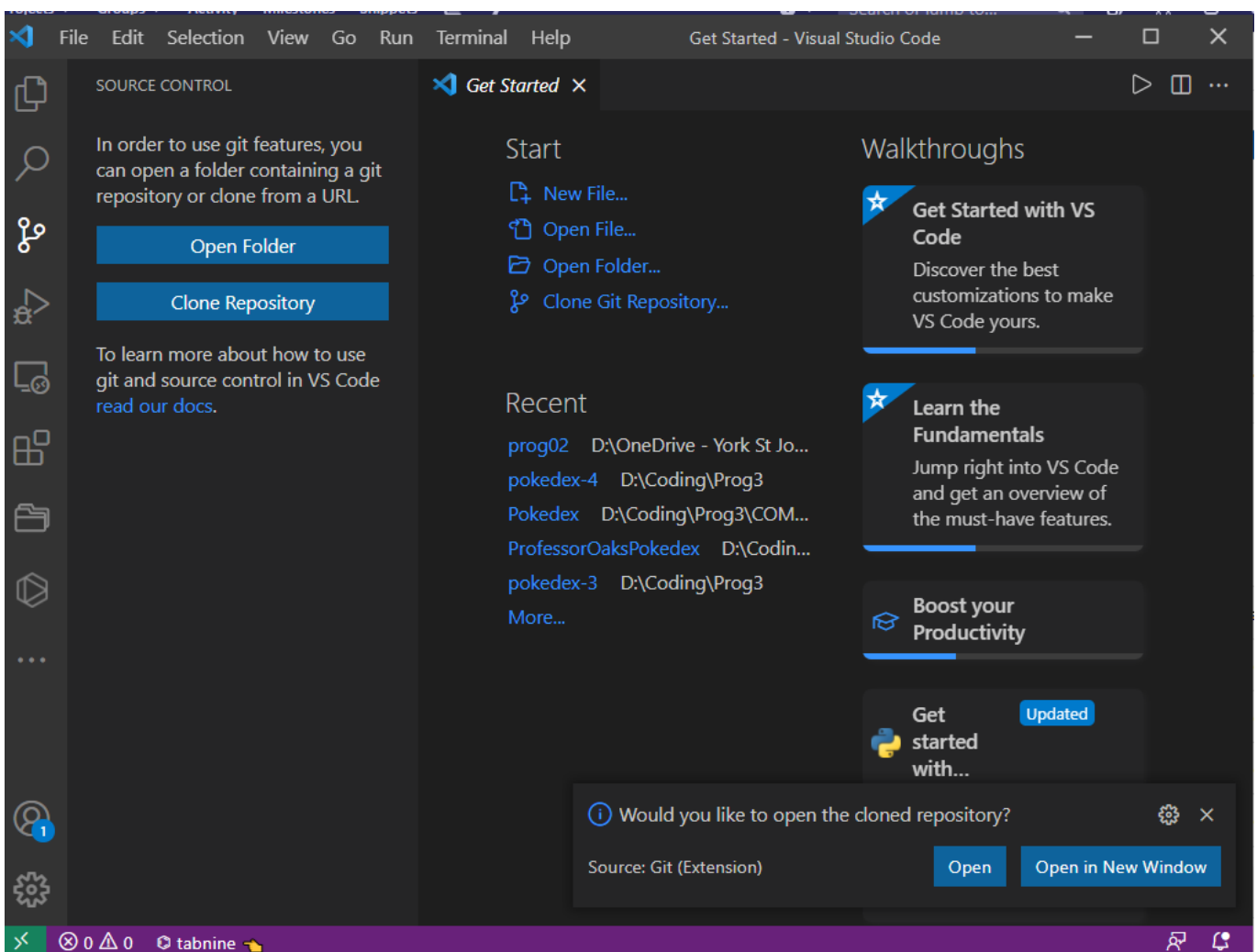
- Ensure you have no windows open in VSCode
- Click the GitLens icon to open the Git/Source Control Window (shown below)



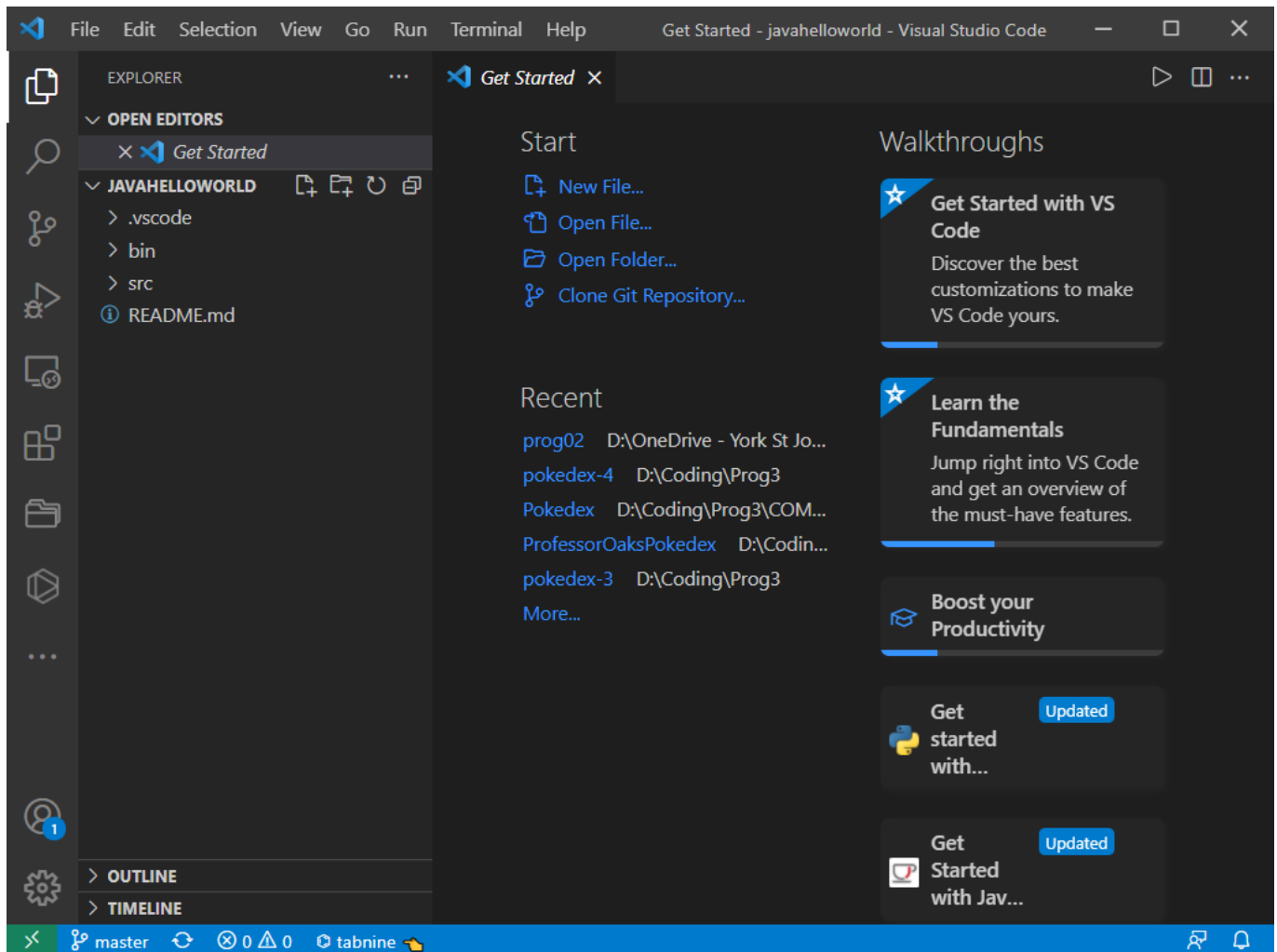
- Click the **Clone Repository** button
- In the popup window enter the repository link <https://git.yesjcs.net:8888/a.guest/javahelloworld.git> - you can type it in or copy and paste it, and press return.



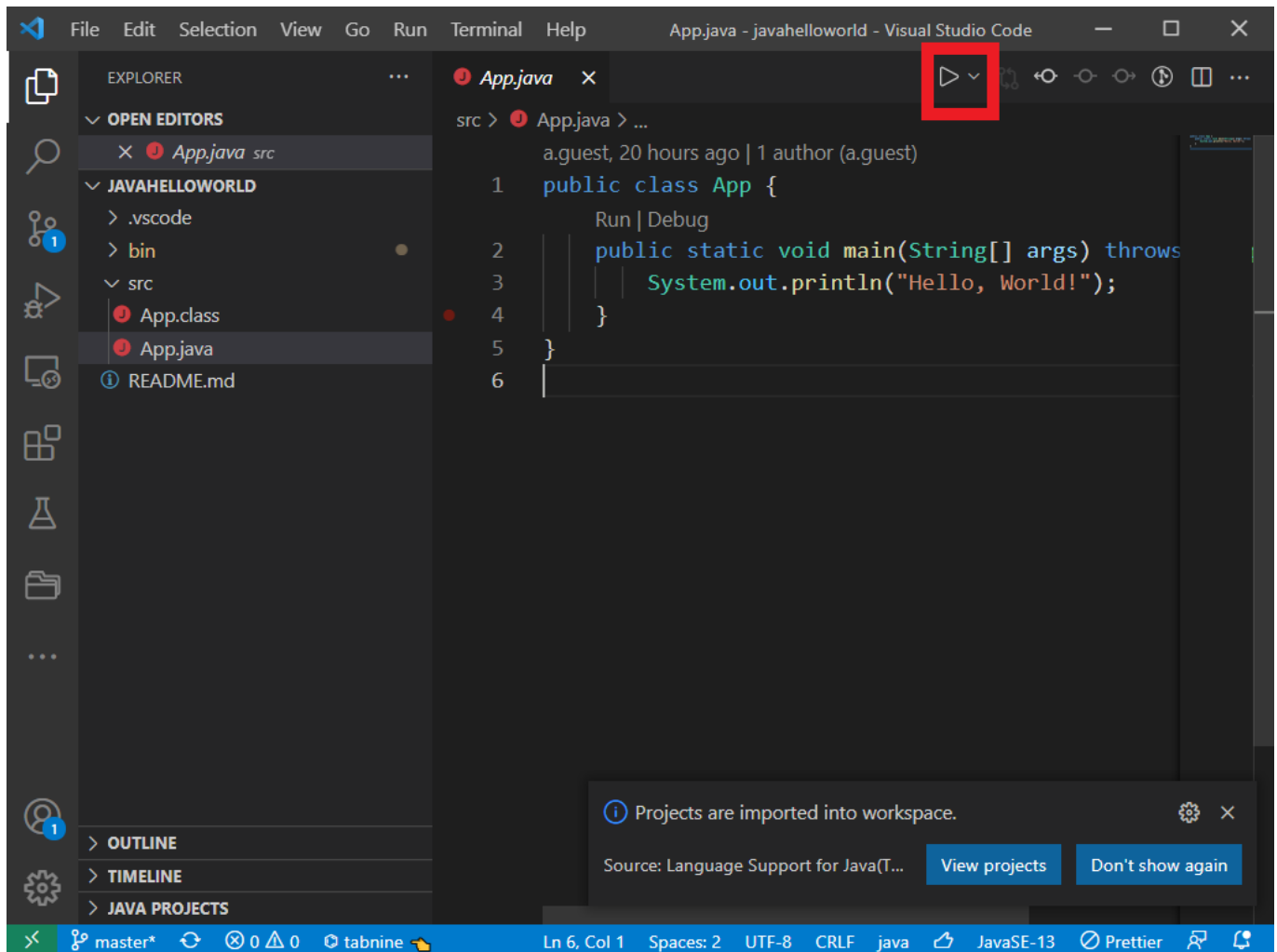
- Select a folder to clone the repository to. Note the process will create a folder called javahelloworld in the folder you select, so select the folder you want to keep all your code for this module in.
- Once the repository has been successfully cloned you should see a popup window appear. Click the **Open** button.



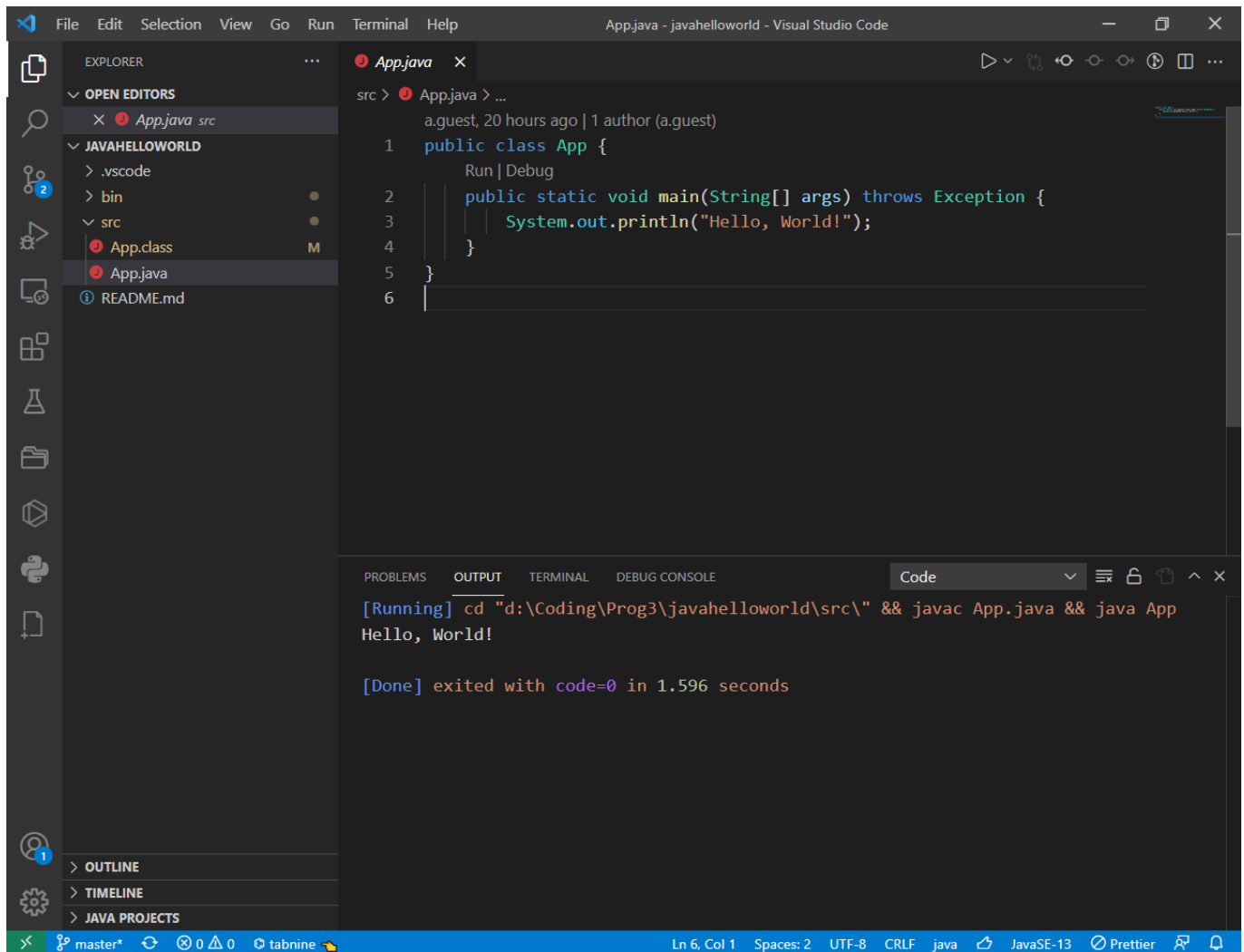
- You should now see the window below.



- In the Explorer window click on > **src** to expand the src folder. Now click on **App.java** to view the code



- Press the play button to run the code (highlighted in the image above). You should see the **Output** window appear, looking something like the image below.



```
File Edit Selection View Go Run Terminal Help
App.java - javahelloworld - Visual Studio Code

EXPLORER
OPEN EDITORS
App.java src
JAVAHELLOWORLD
.vscode
bin
src
App.class
App.java
README.md

src > App.java > ...
a.guest, 20 hours ago | 1 author (a.guest)
1 public class App {
   Run | Debug
2     public static void main(String[] args) throws Exception {
3         System.out.println("Hello, World!");
4     }
5 }
6

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
[Running] cd "d:\Coding\Prog3\javahelloworld\src\" && javac App.java && java App
Hello, World!

[Done] exited with code=0 in 1.596 seconds

Ln 6, Col 1 Spaces: 2 UTF-8 CRLF java JavaSE-13 Prettier
```

Getting Started With GitLab

GitLab is our source control software. It allows us to upload code we are working on so we can access it anywhere. It also tracks the changes we make to the code, so if we screw up we can go back to an earlier version before the mistakes were made.

We also use it as a secure place to upload code for assessments.

Source control is used in industry and much of it is Git based (GitLab or GitHub), it is essential you learn how to use it, not only for your degree but for your future career.

- Log in to GitLab. Use the link https://git.yjsjcs.net:8888/users/sign_in and login in to GitLab using your standard University username (the bit before @yorks.ac.uk in your email address) and your password for the Computing network. This is not your normal University password. If you don't know it, email Matt Baxter and he'll reset it and email you the new password (m.baxter@yorks.ac.uk).
- You should see something like this, but with less repositories 😊



Every repository you create will be stored here.

- We are going to make a copy of the javahelloworld repository in your GitLab account.

- First you have to create a repository in GitLab to upload the code to. Click on the green **New Project** button at the top right of the web page (for GitLab).
- Enter a name for the project. It doesn't need to be the same as the project you want to upload but it probably helps if it is informative and similar.
- Set the Visibility Level to **Public**
- Ensure the **Initialize repository with a README** is **NOT** ticked
- Click the **Create Project** button
- Click on the blue **Clone** button then click on the button to the right of the line starting "https://" to copy the URL to the clipboard. **Keep the browser open just in case**
- Return to VSCode
- Ensure you have the javahelloworld project open
- Click the GitLens icon
- Click on the ... in the Source Control window. (There are two if you can't follow step 3, try the other one!)
- Select Remote > then Remove Remote. Select the name from the list. (If there is more than one, delete them all one by one)
- Connect to your repository
 1. Click on the ... in the side bar.
 2. Select Remote > then Add Remote...
 3. Paste the URL of your repository - if you've followed the instructions correctly you should be able to just paste it
 4. Enter the name origin
 5. Click on the ... in the side bar.
 6. Select Push
 7. Click OK if a message box pops up

You may need to enter your username and your (computing) password in this process.

- You should see a progress window pop up. When it has gone you should have uploaded the project to your repository
- Return to your browser and the GitLab page and refresh it, you should see a file list for the uploaded project.

You will need to repeat this process any time you clone a repository and want to keep a copy of it.

Changing Your Code

Return to VSCode and have a look at the App.java code.

```
public class App {  
    public static void main(String[] args) throws Exception {  
        System.out.println("Hello, World!");  
    }  
}
```

This is our `Hello World` programme in Java.

The line `System.out.println("Hello, World!");` outputs the string "Hello, World!" to the screen. `println` is **Print Line**, it outputs the text then adds a new line character. We can also use `print` which will just output the text and leave the cursor at the end of the text.

The code

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

creates a function called **main**. In all Java programmes **main** is the entry point for the code. When the programme is run, it starts running the code in **main**. We'll have a look at what `public`, `static`, `void`, etc mean in a while, for now it's just important to learn that **main** is the code's starting point.

The code

```
public class App {  
  
}
```

creates the programme itself. In this case our programme is called `App`. Again we'll discuss what `public` and `class` mean later.

- Edit `App.java` to output different text and run it. Make sure it works and gives the expected output.

Updating Your Repository

Now you've made changes to your code and that means the code on your computer and the code in your repository are different. We'll look at how to update the repository.

This is a fairly painless process.

- Click on the GitLens icon if the Source Control window isn't open. You should see a little blue circle on the icon with a number in it. The number is the amount of changes you've made
- Under the **Source Control** heading is a text box. This text box is for entering *Commit* messages. These are little notes identifying what changes were made during the update. You must enter a commit message, do so now

- Above the text box is a little tick icon. Click it. The changes you have made are now committed to your local repository. This means some hidden files in your application folder have been updated to record the changes made since you last made a Commit (or initially created/cloned the repository).
- The changes in the Commit have not yet been uploaded to the repository. To upload them click on the "..." just above the text box (not the one higher up), and click "Push". You may need to enter your username and password. Once this completes your code should be uploaded to the repository.
- Return to your browser and refresh it to see the changes.

You should Commit & Push regularly while you are working on a project, not just at the end of a session.

Working From Home/Elsewhere

If you want to work from home on one of your own repositories you will need to clone it following the instructions at the top of this page but changing the repository URL to the URL from your repository.

Once you have local copy of a repository you can update it as shown above.

You only need to clone to a specific machine once. When you already have a repository cloned to your machine you can open the project, click the GitLens icon, click the "..." and select Pull to update your local code to match the repository.

Example process

1. Clone a repository in the lab to work on an exercise
2. Immediately create your own repository and upload the code to it
3. Work on the code in class
4. Commit the changes and Push them to GitLab
5. Get home, clone the repository from your GitLab to your home computer
6. Work on the code
7. Commit the changes and Push them to GitLab
8. Next time you are in the lab and want to work on the same project -
 1. Open the project in VSCode
 2. Pull the changes from the repository
 3. Change the code
 4. Commit the changes and Push them to GitLab

and so on.