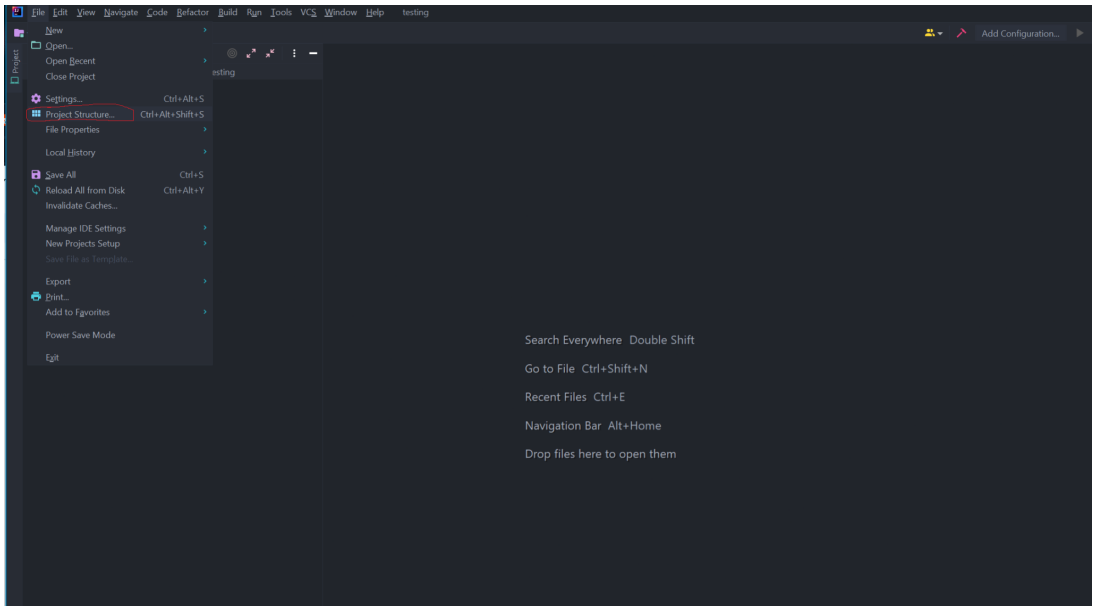
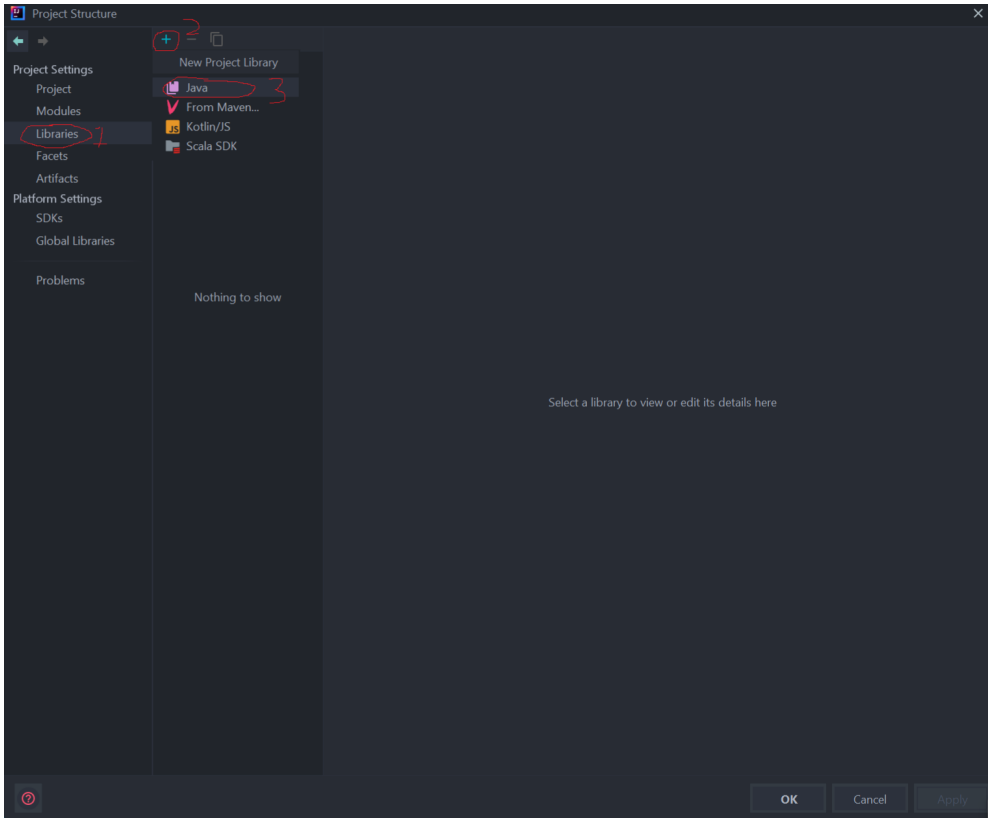
1. Download the zip file, then open the folder called “Project and Sample Programs”
2. Inside should contain codes and sample programs related to the work that we did for the project.
3. Using Intellij (the IDE that we worked with), create a “new project”, then choose java, then name the project and choose a directory to save the project in.

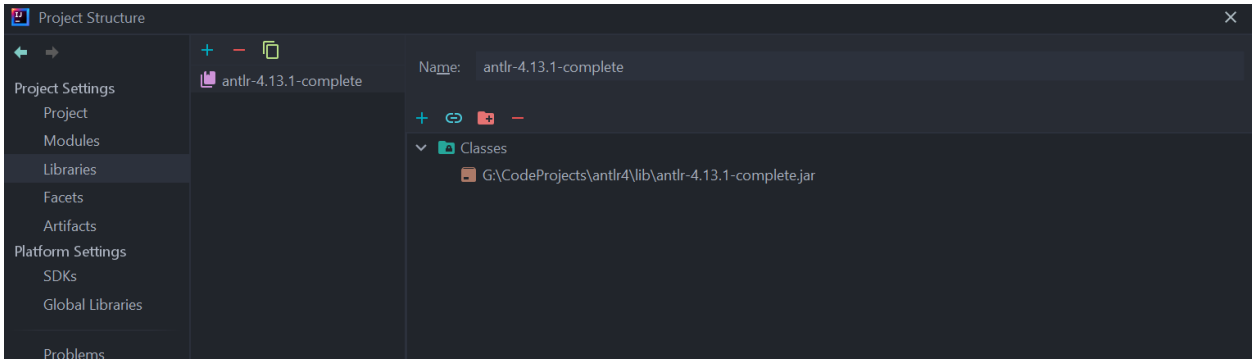
4. After, select **File**, on the top left side upper corner then select **Project Structure**



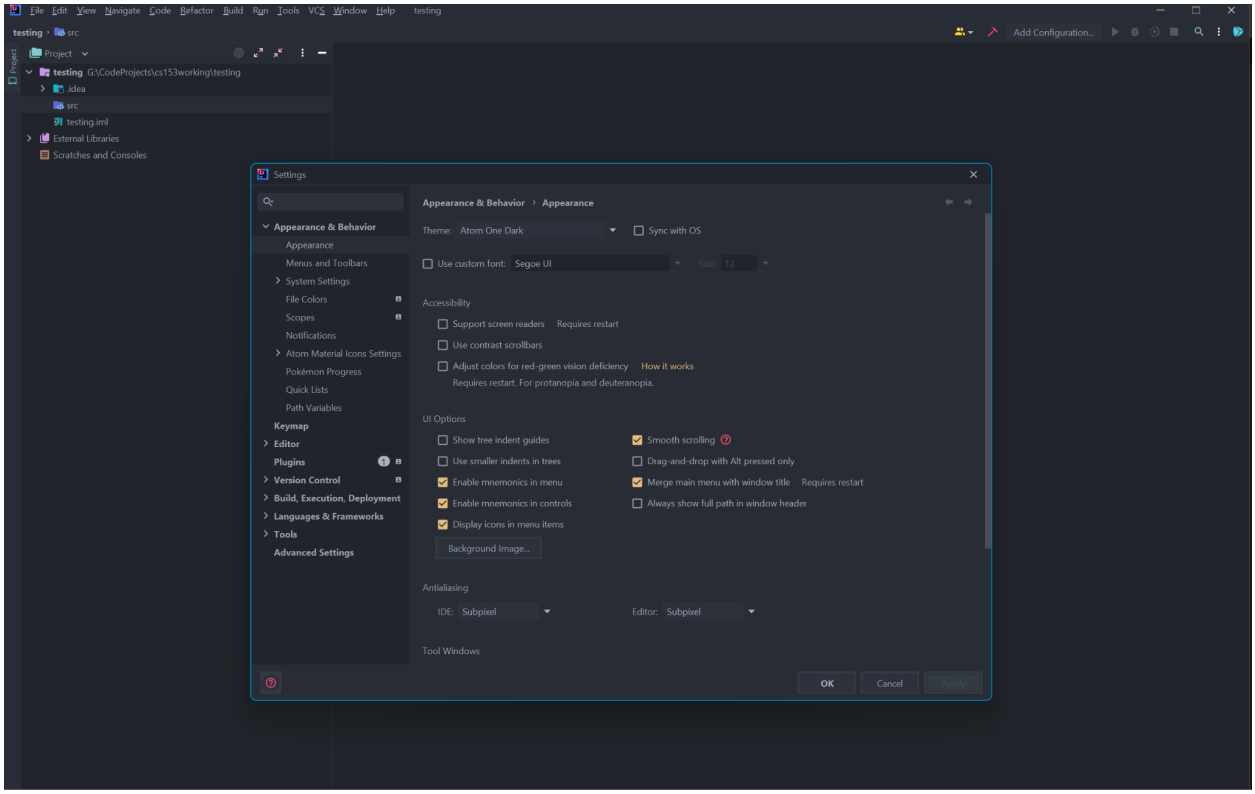
5. Inside the project structure tab, click on **Libraries,** then click on the **+**button, then inside the drop-down menu, choose **java**



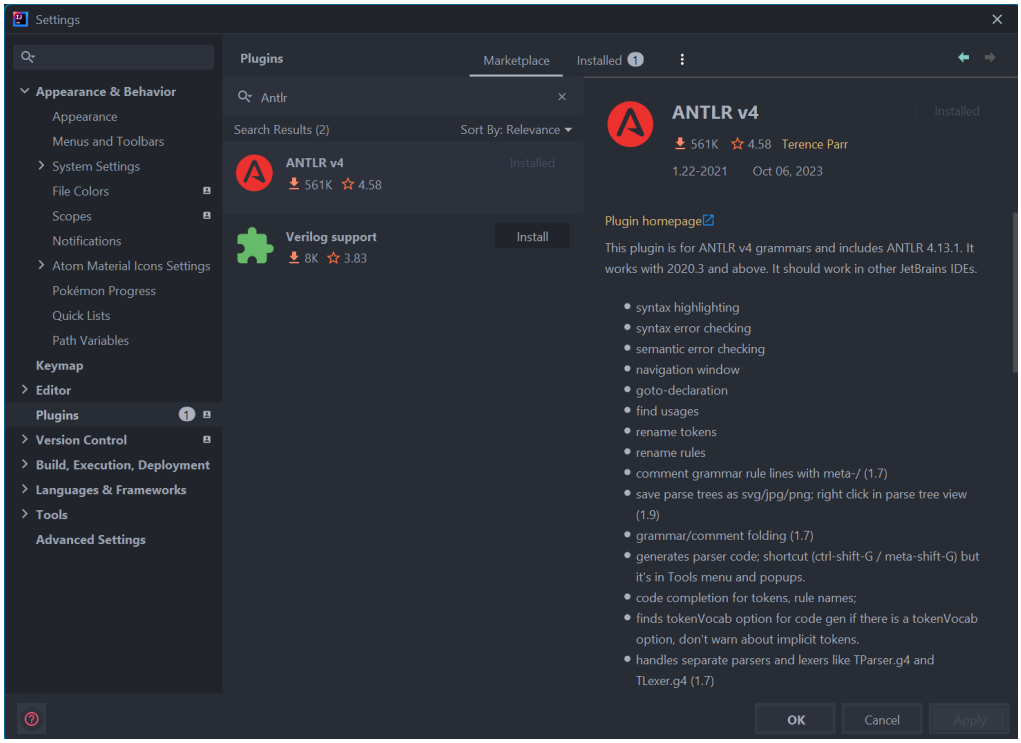
6. Afterwards, choose the directory that contains the **antlr-4.13.1-complete.jar**, then click ok. The picture below sure be what we have so far, last click **apply** (at the bottom right) then click **ok**



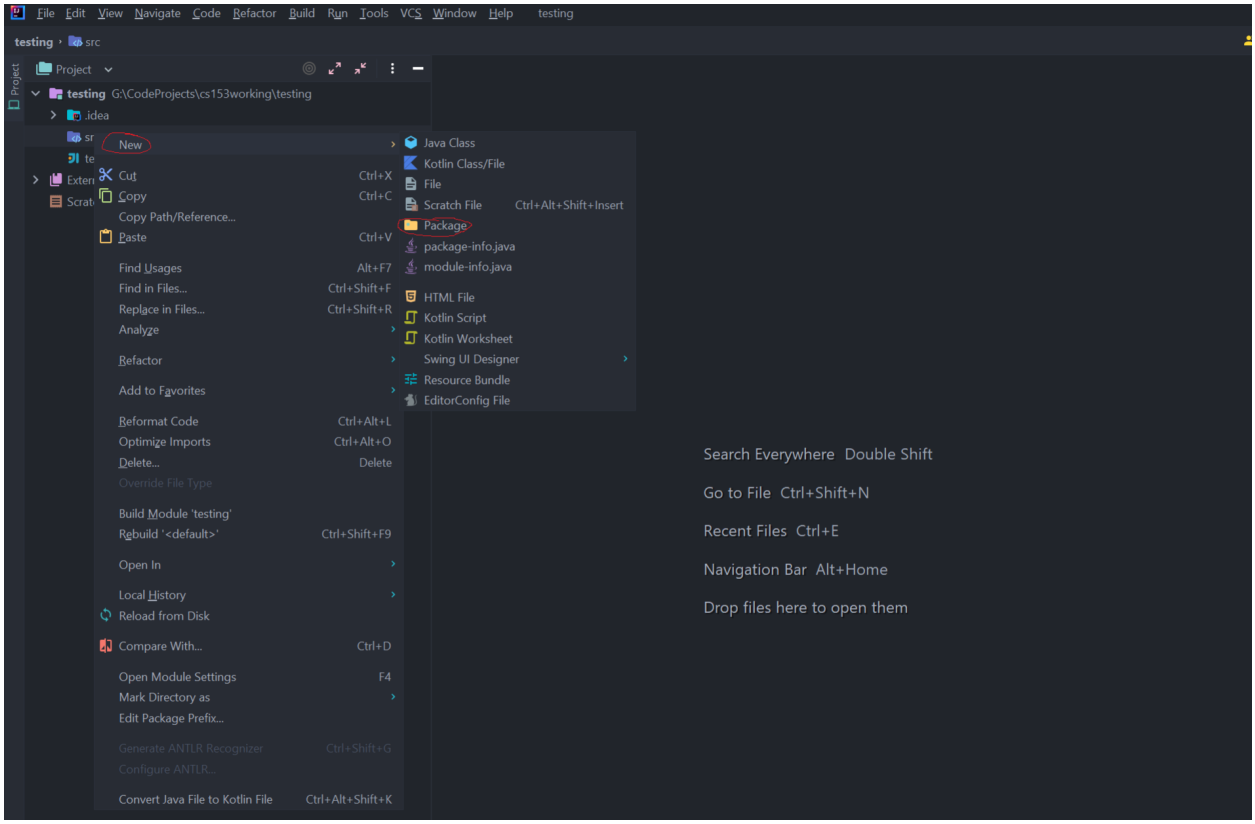
7. After importing the antlr-4.13.1-complete.jar, go navigate back to the screen as seen in step 4, then Do press [**CTRL ALT S**] on the keyboard to open the setting menu in intellij



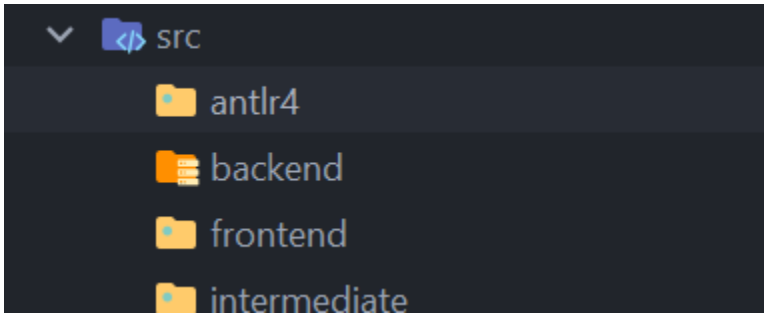
8. Inside the setting menu, Click on **Plugins,** then on Marketplace and in the search text form – type Antlr4. A plugin called Antlr4 should pop up so click installed



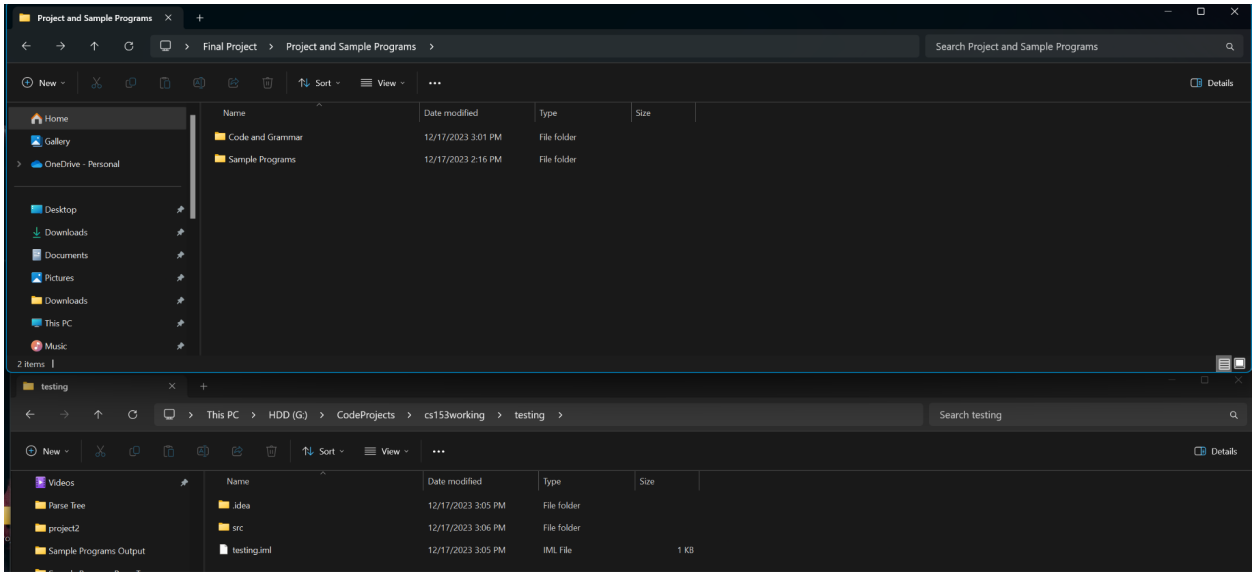
9. After installation is complete, restart the IDE and make sure that the plugin is enabled.
10. Now, right click on **src,** then click **new,** then click **Package.**



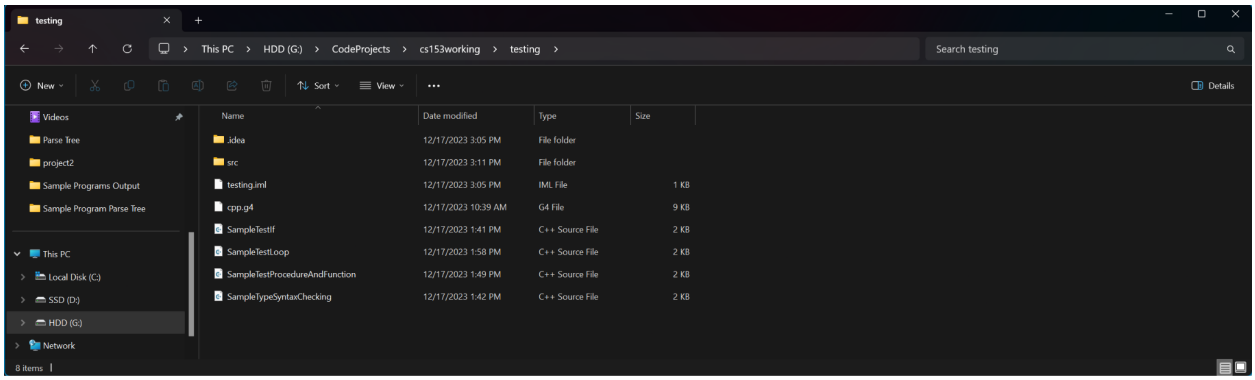
11. Make 4 packages and named each one antlr4, backend, frontend, intermediate



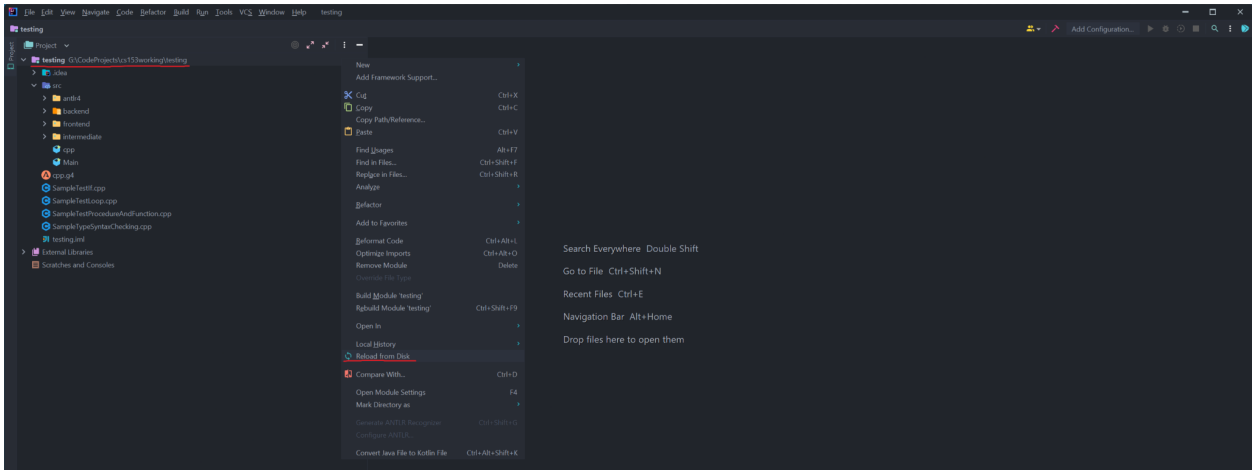
12. Afterwards, open the folder called “Project and Sample Programs” (the folder that has everything in step 1), and also open the directory where we made the “new project” (in step 3)



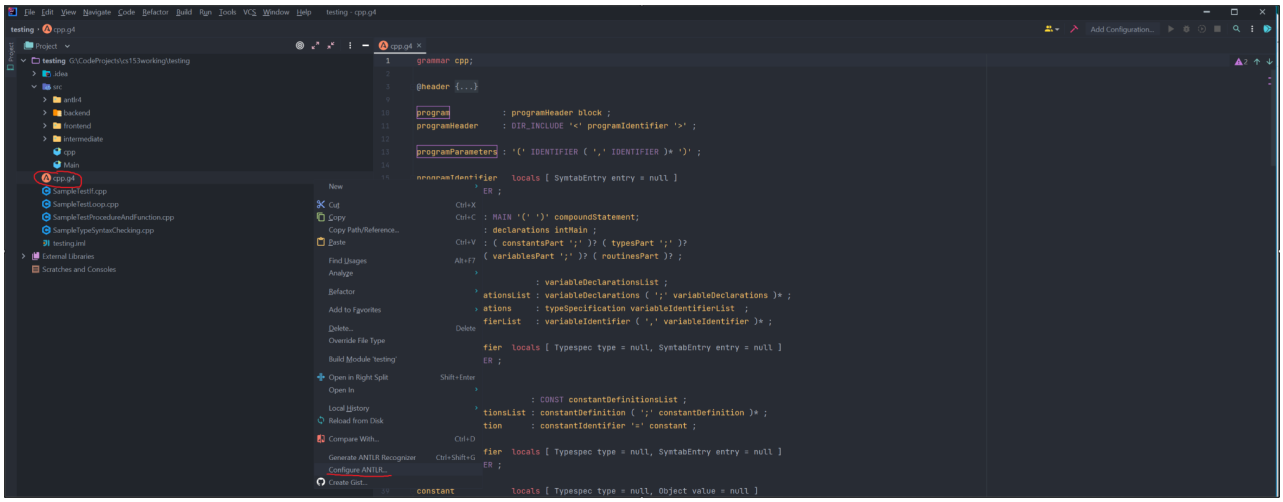
13. Open the “**Code and Grammar”** folder which contains the src code and grammar file called cpp.g4. Drag the **cpp.g4** file and drag the **src folder** over to the directory where we made the “new project” (in step 3). Renavigate back to the “Project and Sample Programs” folder. Then open the **“Sample Programs”** Folder, then open the “**Sample Programs Code”** which contains the sample code for our project then also drag those into the directory where we made the “new project” (in step 3). The picture below shows what our “new project” should look like now.



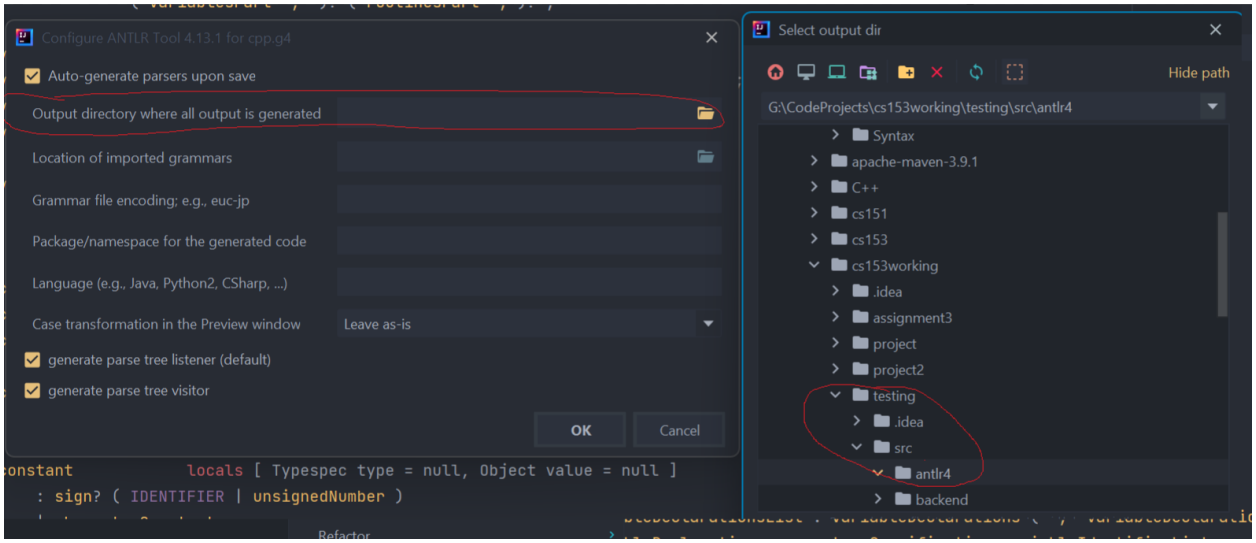
14. Navigate back to intellij, then right click on the “new project” (mine is named testing), then click **Reload from Disk**

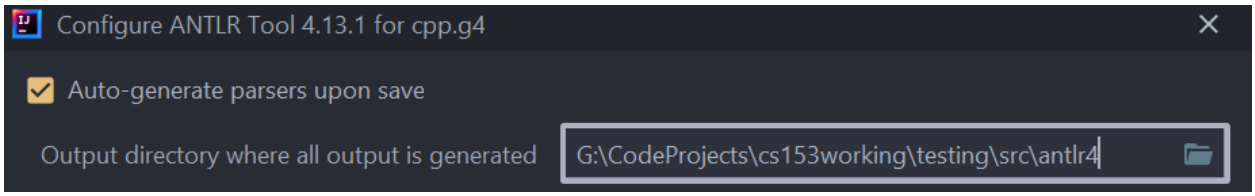


15. Afterwards, right click cpp.g4 then click Configure ANTLR

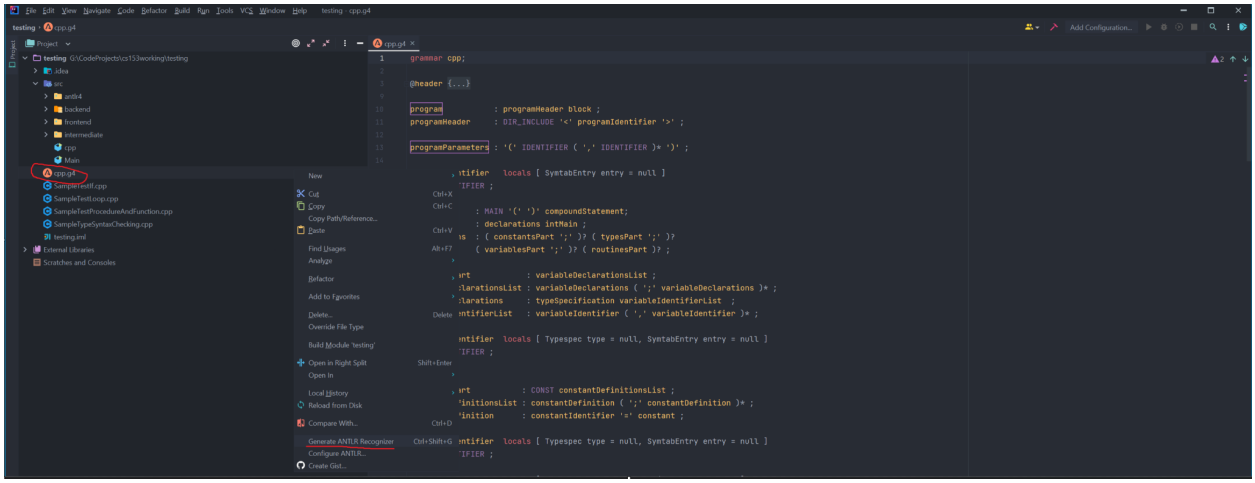


16. In the menu, click on the **output direct directory where all output is generated**, then located the “new project” directory then choose src then select antlr4 and click ok”

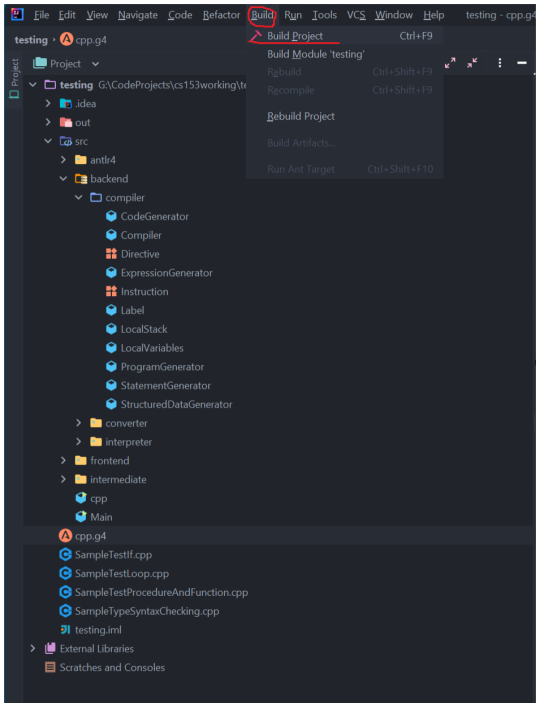




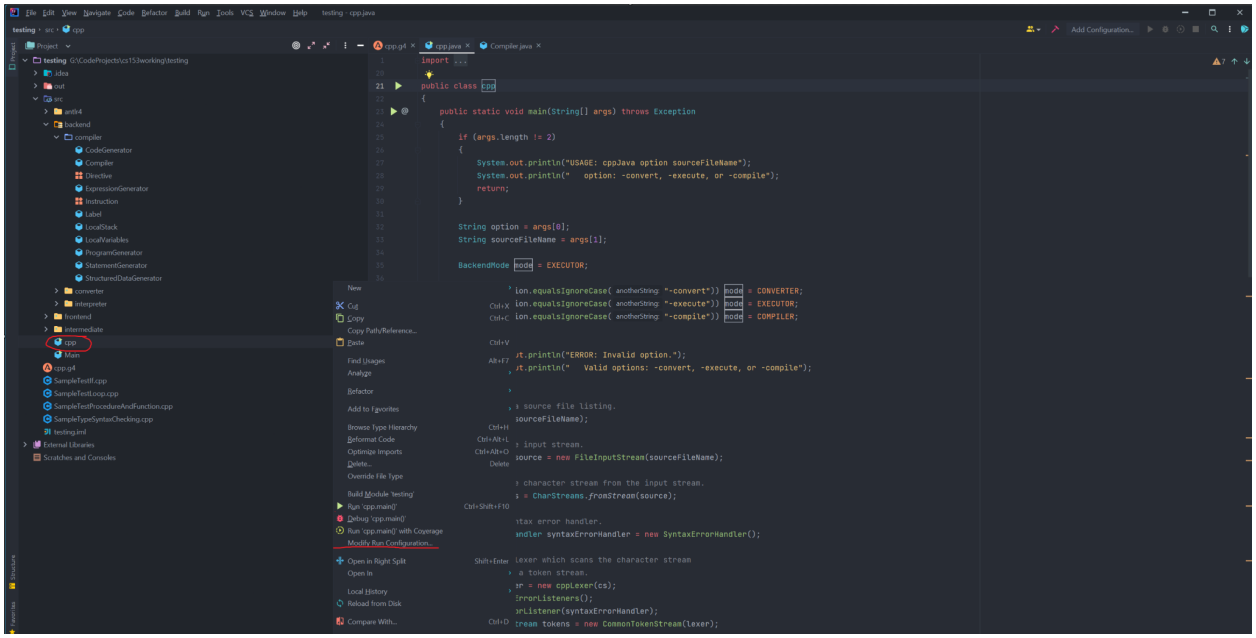
17. Afterwards, right click on the cpp.g4 file again and this time, click on **Generate antlr4 recognizer**



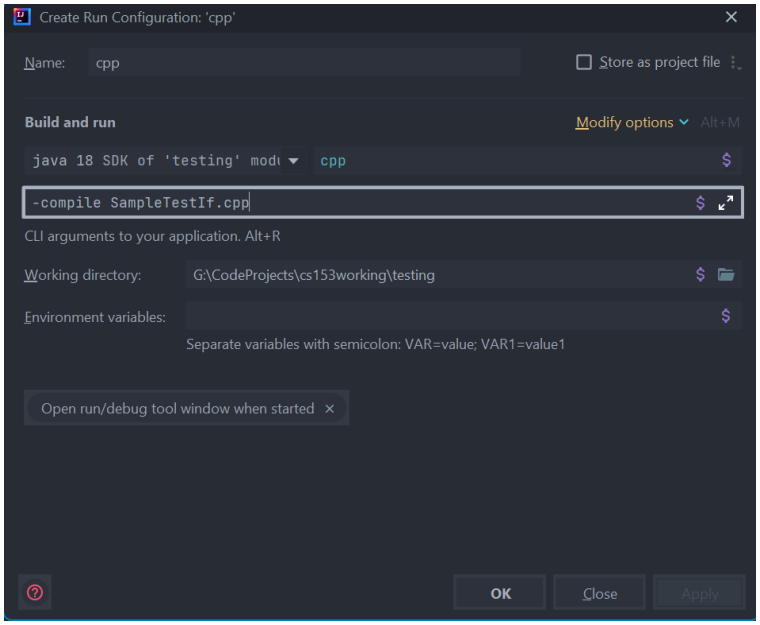
18. Now click on **Build**, in the upper area then click **Build Project**



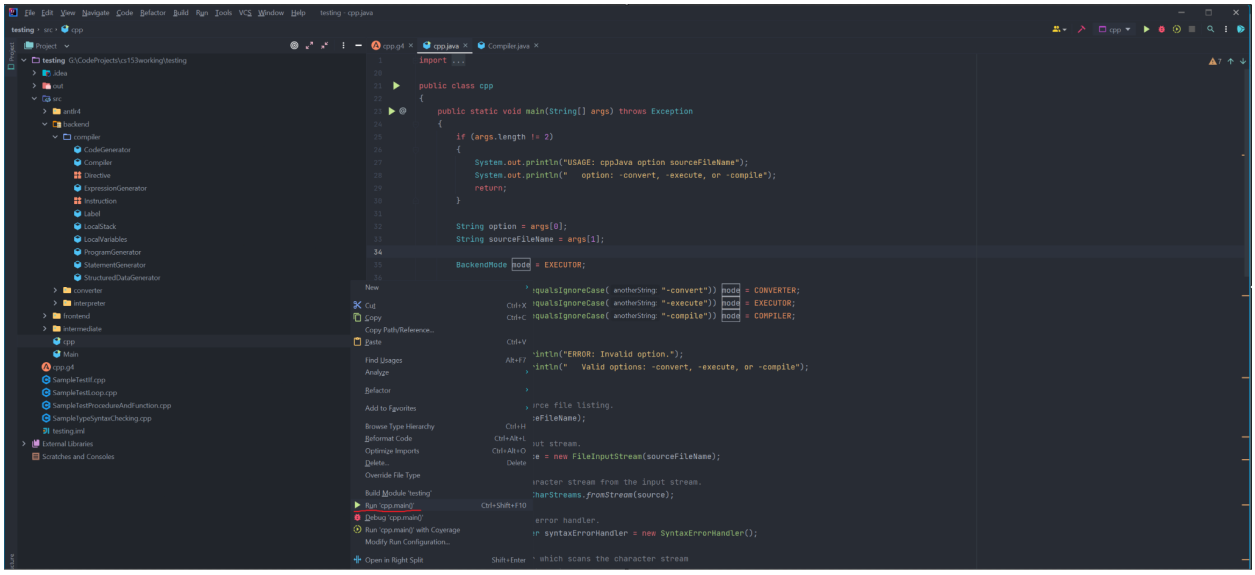
19. The program should now be set up and ready to compile.
20. To compile the sample programs, navigate to the cpp.java executable file and right click on it, and then click **modify run configuration**



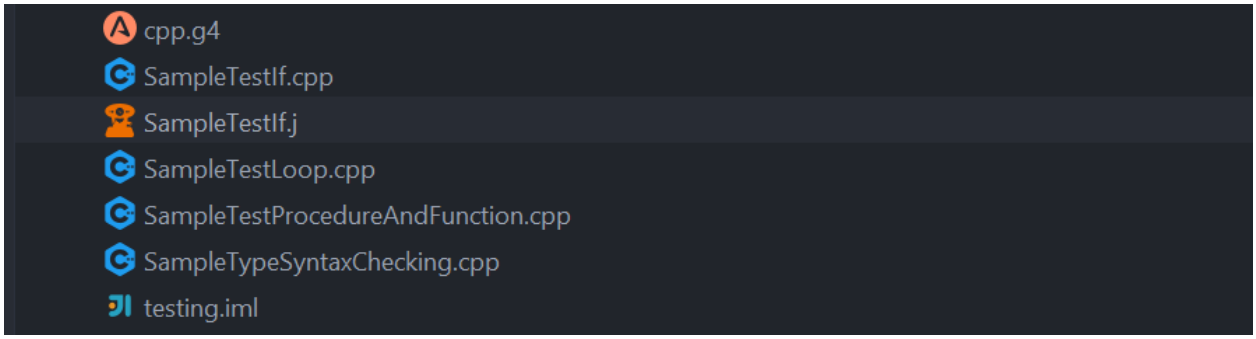
21. In the new menu, input **-compile [the sample programs here]**



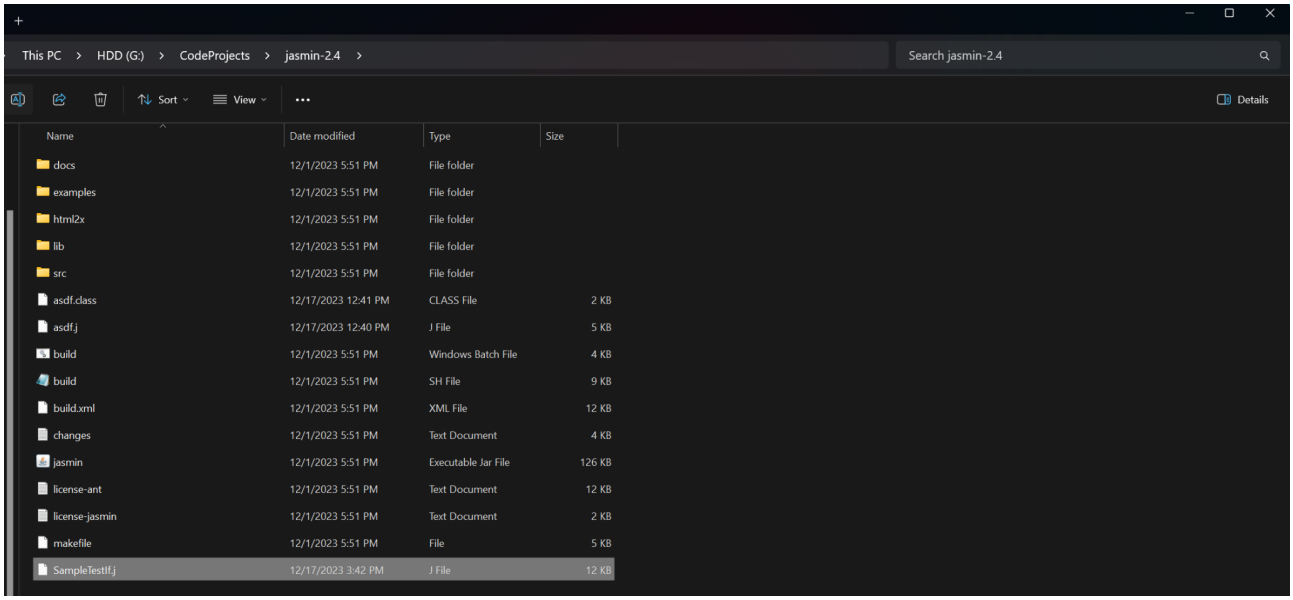
22. Click apply then click ok.
23. Afterwards, right click on the cpp.java file and click **Run cpp.main()**



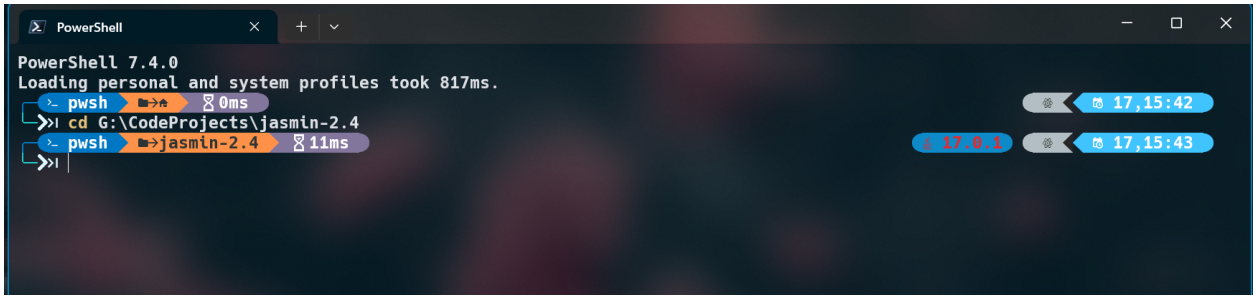
24. Assuming that the file is written correctly based on our language, everything should be in running order and the Object File jasmin file will be generated. If you do not see the jasmin file, repeat step 14 in order to reload everything from disk. My generated jasmin file is named SampleTestIf.j



25. Now, navigate to the folder where you have jasmin installed, mine is in G:\CodeProjects\jasmin-2.4. Keep note of this directory. Now drag over the generated jasmin file to the jasmin directory,



26. Open your terminal, I am using Powershell in this example, then type cd [Directory of Jasmin Here]



27. Lastly, to create the .class file type **java -jar jasmin.jar [jasmin file here]**, then type **java [generated class name]**. Then you should see that the program compiles and executes.