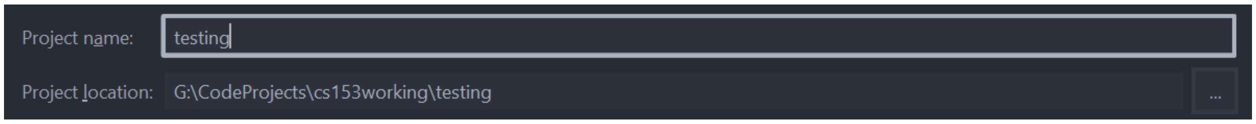
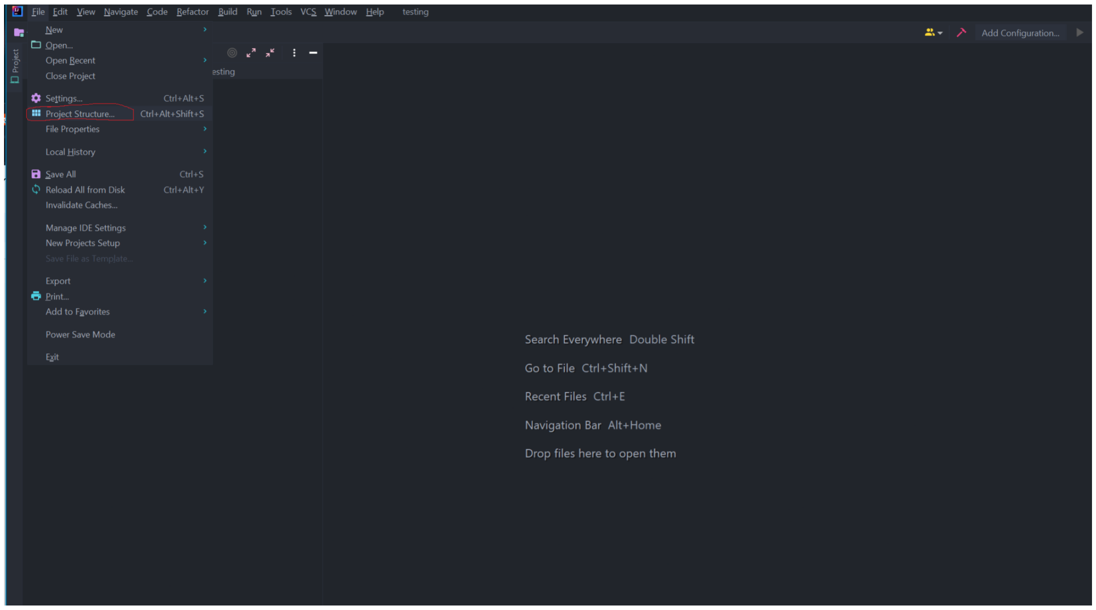
NOTES: This instruction was written before the finalization of the sample programs, so if there are some files that don't show up like in the pictures I provided it should be ignored such as DemoTestIf.cpp.
The main thing that should be present and accounted for is everything that is in src, cpp.java, cpp.g4.

1. Download the zip file, then open the folder called "Project"
2. Inside should contain everything related to the work 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.

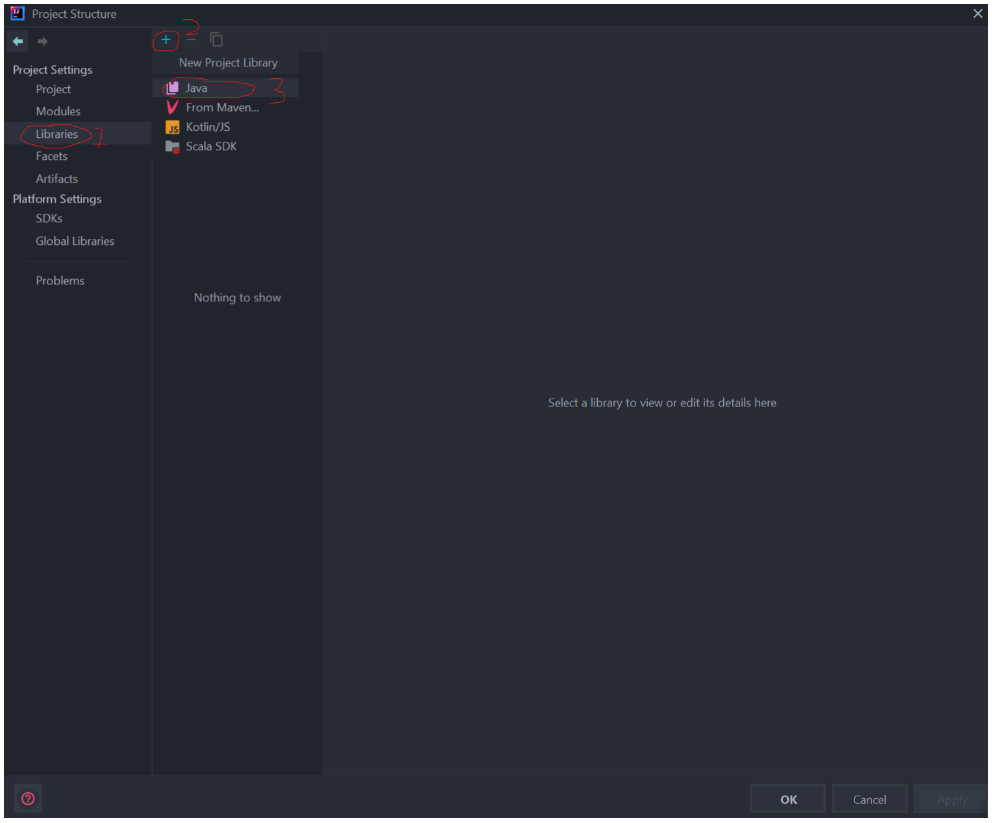


Project name: testing
Project location: G:\CodeProjects\cs153working\testing

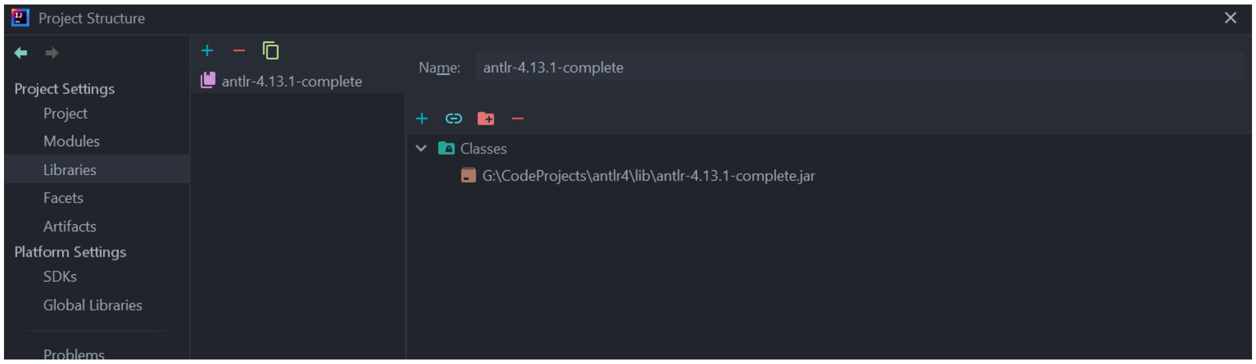
4. After, select **File**, on the top left side upper corner then select **Project Structure**



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



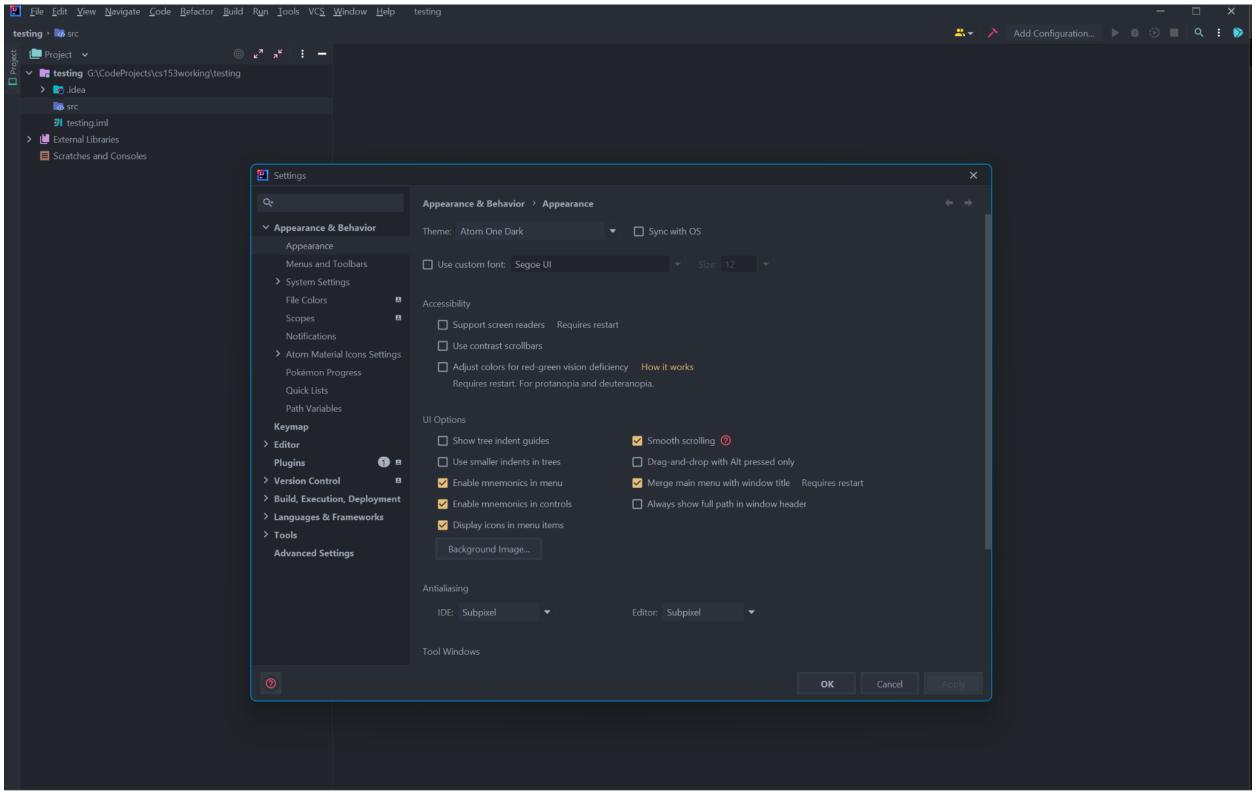
6. Afterwards, choose the directory that contains the antlr-4.13.1-complete.jar, then click ok. The picture below sure be what we have so far, last click **apply** (at the bottom right) then click **ok**



Name: antlr-4.13.1-complete

G:\CodeProjects\antlr4\lib\antlr-4.13.1-complete.jar

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



Theme: Atom One Dark
☐ Sync with OS
☐ Use custom font: Segoe UI Size: 12

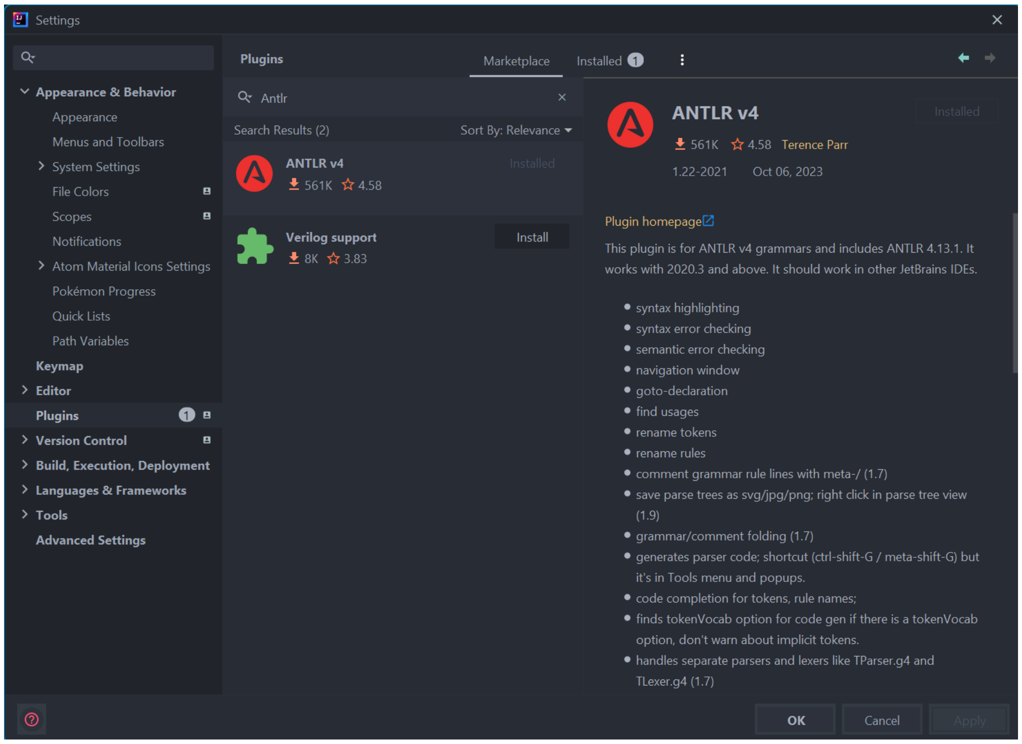
Accessibility
☐ Support screen readers Requires restart
☐ Use contrast scrollbars
☐ Adjust colors for red-green vision deficiency How it works
Requires restart. For protanopia and deuteranopia.

UI Options
☐ Show tree indent guides
☐ Use smaller indents in trees
☑ Enable mnemonics in menu
☑ Enable mnemonics in controls
☑ Display icons in menu items
☑ Smooth scrolling
☐ Drag-and-drop with Alt pressed only
☑ Merge main menu with window title Requires restart
☐ Always show full path in window header
Background Image...

Antialiasing
IDE: Subpixel
Editor: Subpixel

Tool Windows

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



Antlr
Search Results (2) Sort By: Relevance

ANTLR v4 561K ☆ 4.58 Installed
Verilog support 8K ☆ 3.83 Install

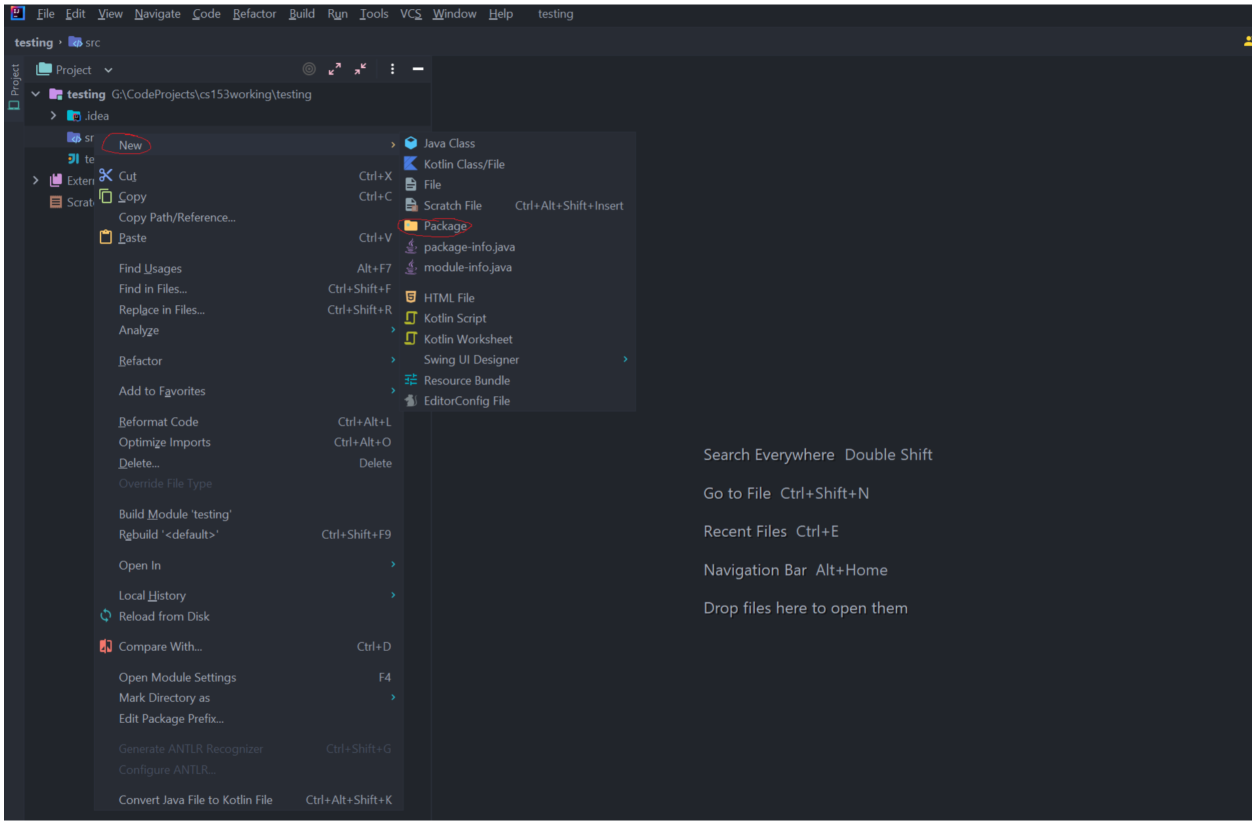
ANTLR v4 Installed
561K ☆ 4.58 Terence Parr
1.22-2021 Oct 06, 2023

Plugin homepage

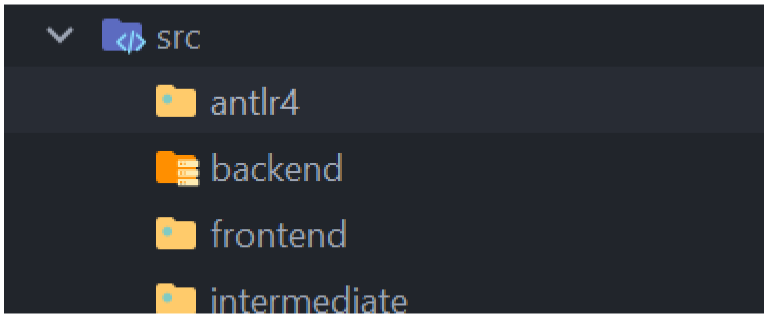
This plugin is for ANTLR v4 grammars and includes ANTLR 4.13.1. It works with 2020.3 and above. It should work in other JetBrains IDEs.

- syntax highlighting
- syntax error checking
- semantic error checking
- navigation window
- goto-declaration
- find usages
- rename tokens
- rename rules
- comment grammar rule lines with meta-/ (1.7)
- save parse trees as svg/jpg/png; right click in parse tree view (1.9)
- grammar/comment folding (1.7)
- generates parser code; shortcut (ctrl-shift-G / meta-shift-G) but it's in Tools menu and popups.
- code completion for tokens, rule names;
- finds tokenVocab option for code gen if there is a tokenVocab option, don't warn about implicit tokens.
- handles separate parsers and lexers like TParser.g4 and TLexer.g4 (1.7)

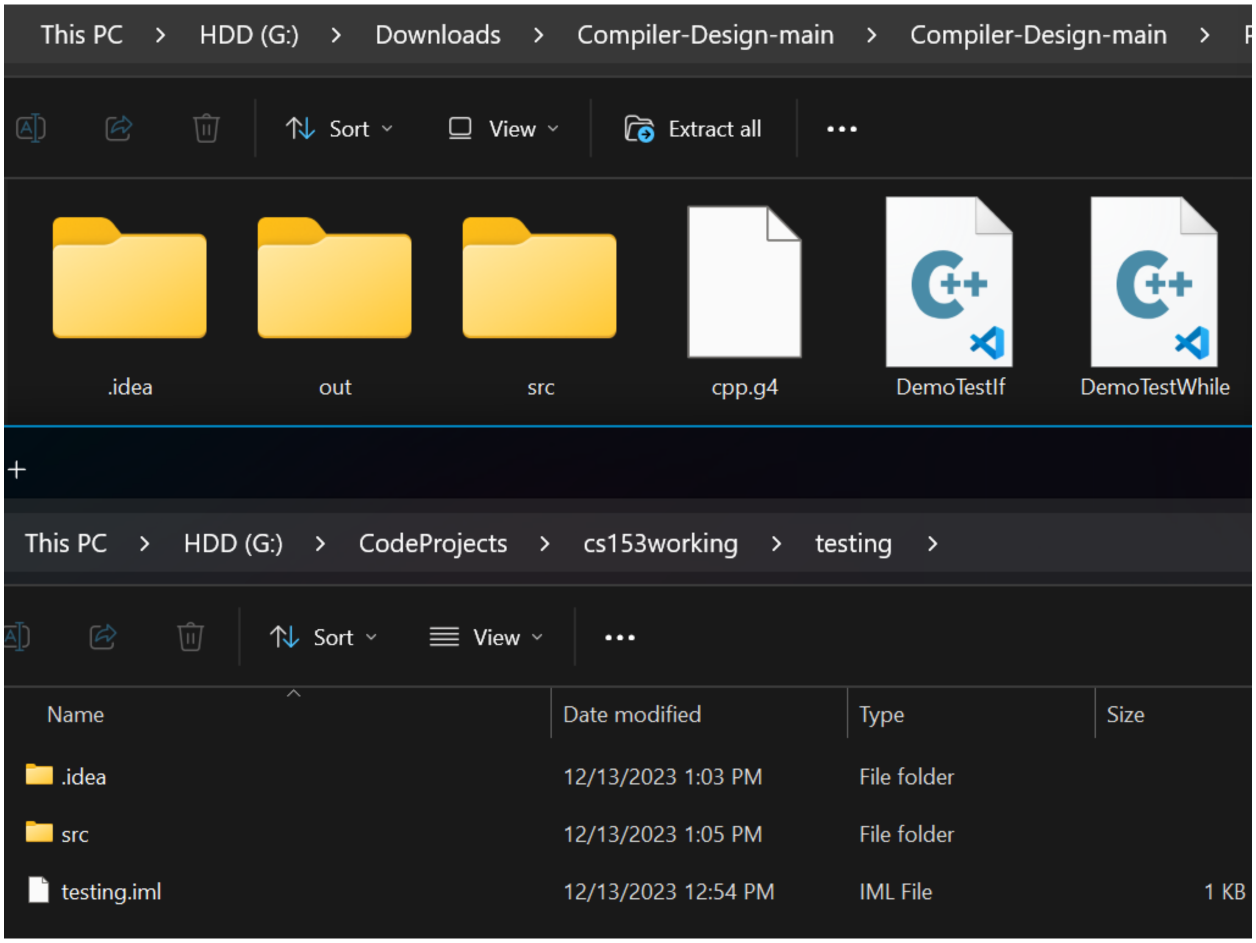
9. After installation is complete, restart the IDE and make sure that the plugin is enabled.
10. Now, right click on **src,** then click **new,** then click **Package.**



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

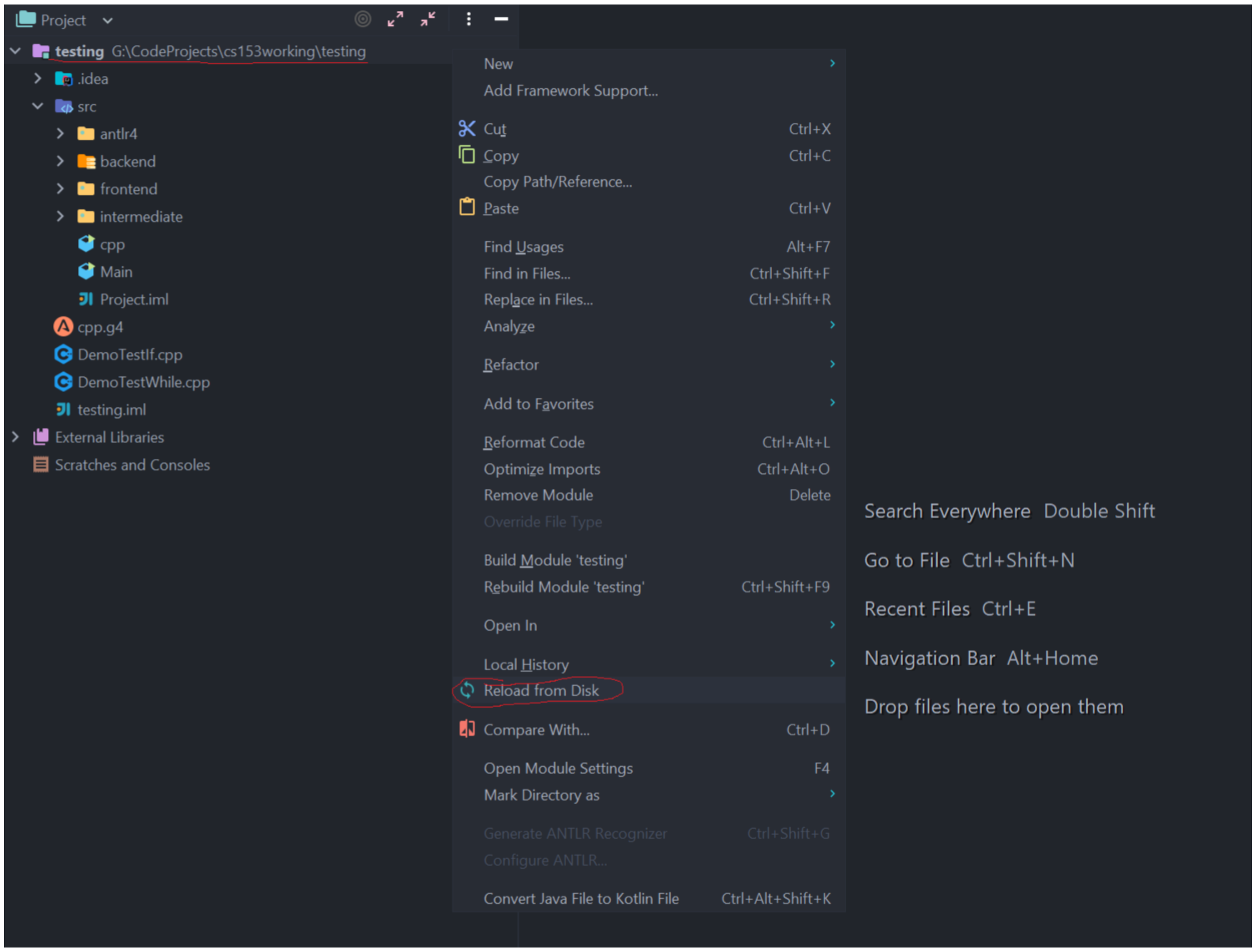


12. Afterwards, open the folder called “Project” (the folder that has everything in step 1), and also open the directory where we made the “new project” (in step 3)

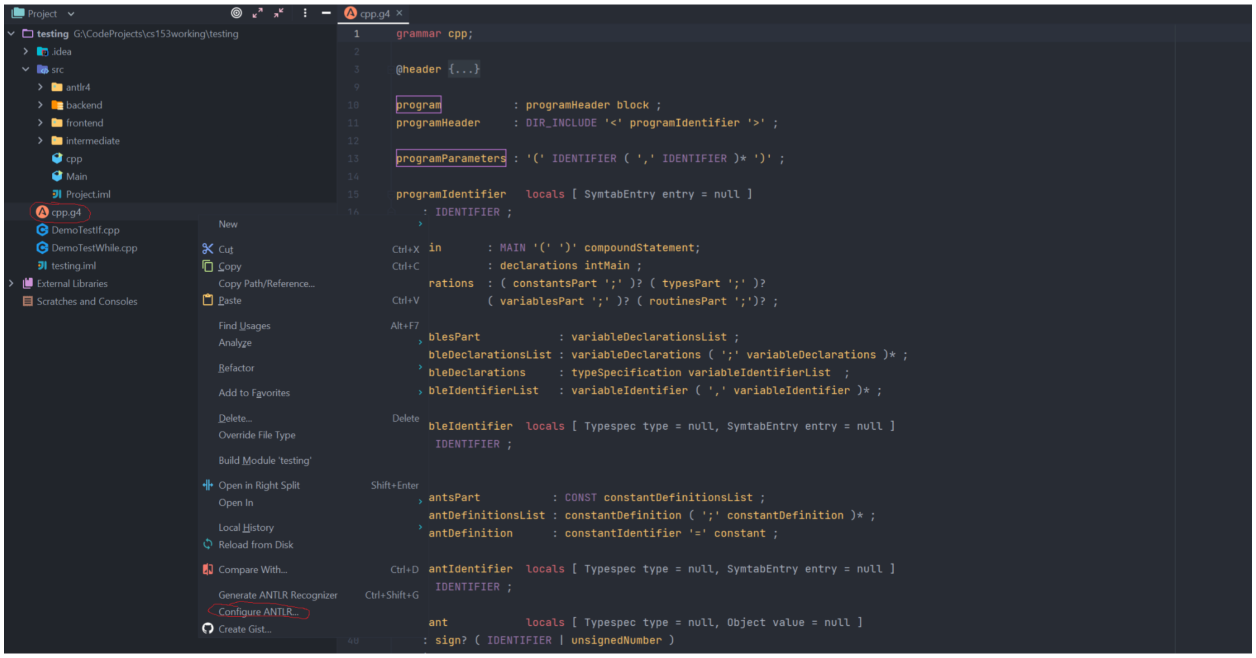

This PC > HDD (G:) > Downloads > Compiler-Design-main > Compiler-Design-main >
Sort
View
Extract all
.idea
out
src
cpp.g4
DemoTestIf
DemoTestWhile
This PC > HDD (G:) > CodeProjects > cs153working > testing >
Sort
View

Name	Date modified	Type	Size
.idea	12/13/2023 1:03 PM	File folder	
src	12/13/2023 1:05 PM	File folder	
testing.iml	12/13/2023 12:54 PM	IML File	1 KB

13. Drag the **cpp.g4** file over and any of the **demo/sample programs over** and lastly drag the **src folder** over too.
14. Navigate back to intellij, then right click on the “new project” (mine is named testing), then **Reload from Disk**

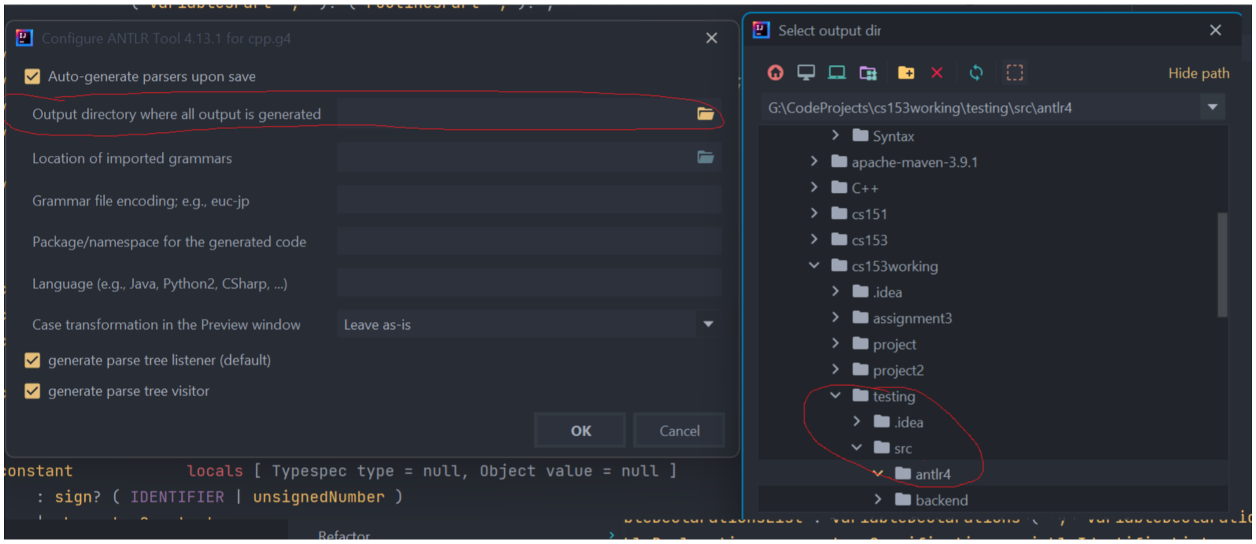


15. Afterwards, right click cpp.g4 then click Configure ANTLR



```
grammar cpp;

@header {...}

program             : programHeader block ;
programHeader       : DIR_INCLUDE '<' programIdentifier '>' ;

programParameters : '(' IDENTIFIER ( ',' IDENTIFIER )* ')' ;

programIdentifier   locals [ SymtabEntry entry = null ]
    : IDENTIFIER ;

in          : MAIN '(' ')' compoundStatement;
            : declarations intMain ;
rations     : ( constantsPart ';' )? ( typesPart ';' )?
            ( variablesPart ';' )? ( routinesPart ';')? ;

blesPart              : variableDeclarationsList ;
bleDeclarationsList : variableDeclarations ( ';' variableDeclarations )* ;
bleDeclarations       : typeSpecification variableIdentifierList  ;
bleIdentifierList     : variableIdentifier ( ',' variableIdentifier )* ;

bleIdentifier  locals [ Typespec type = null, SymtabEntry entry = null ]
 IDENTIFIER ;

antsPart             : CONST constantDefinitionsList ;
antDefinitionsList : constantDefinition ( ';' constantDefinition )* ;
antDefinition        : constantIdentifier '=' constant ;

antIdentifier  locals [ Typespec type = null, SymtabEntry entry = null ]
 IDENTIFIER ;

ant            locals [ Typespec type = null, Object value = null ]
: sign? ( IDENTIFIER | unsignedNumber )
```

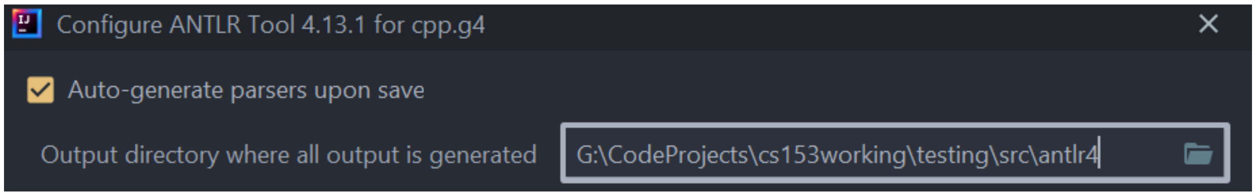
16. In the menu, click on the **output direct directory where all output is generated**, then located the “new project directory the choose src then select antlr4 and click ok”



- [✓] Auto-generate parsers upon save
- Output directory where all output is generated:
- Location of imported grammars:
- Grammar file encoding; e.g., euc-jp:
- Package/namespace for the generated code:
- Language (e.g., Java, Python2, CSharp, ...):
- Case transformation in the Preview window: Leave as-is
- [✓] generate parse tree listener (default)
- [✓] generate parse tree visitor

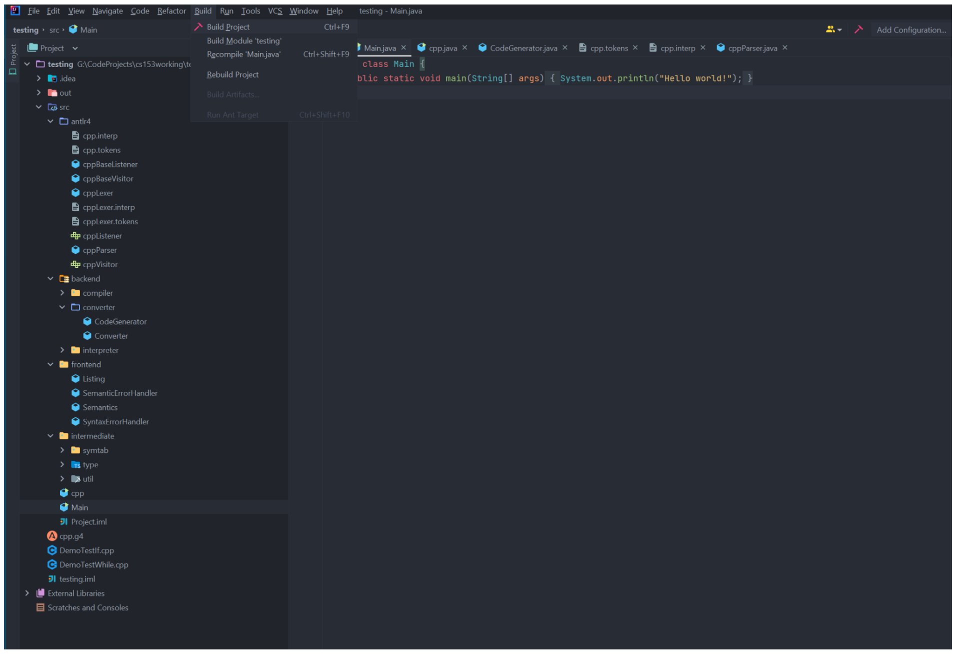
Select output dir: G:\CodeProjects\cs153working\testing\src\antlr4

```
constant        locals [ Typespec type = null, Object value = null ]
    : sign? ( IDENTIFIER | unsignedNumber )
```



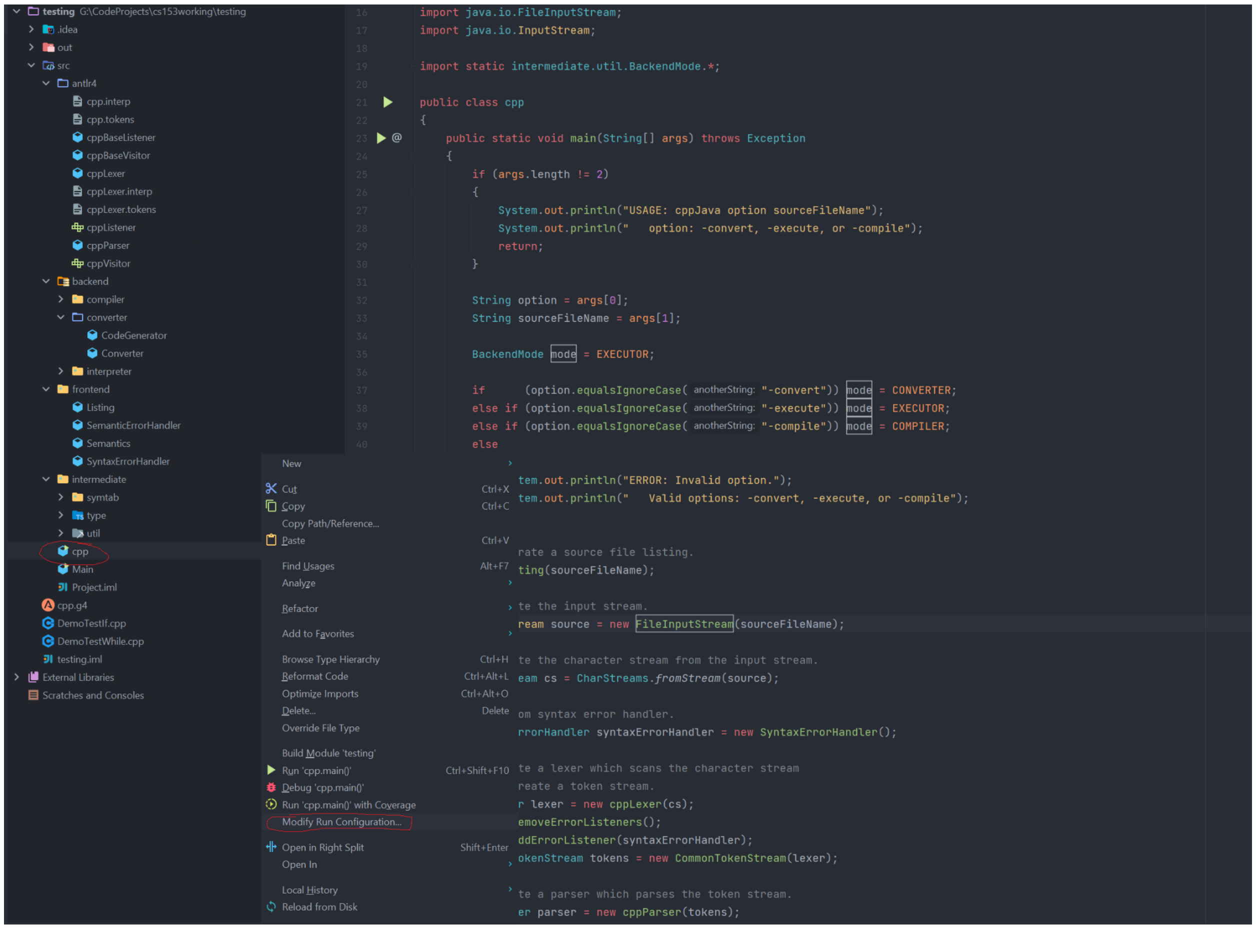
- [✓] Auto-generate parsers upon save
- Output directory where all output is generated: G:\CodeProjects\cs153working\testing\src\antlr4

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**



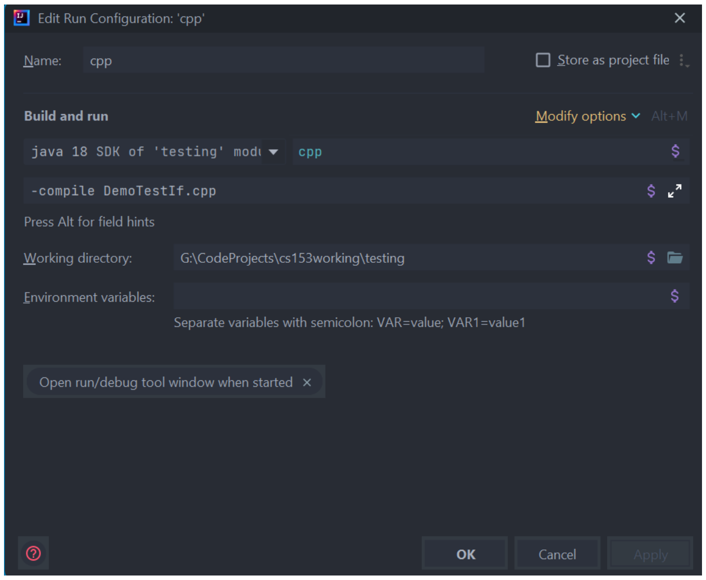
```
class Main {
blic static void main(String[] args) { System.out.println("Hello world!"); }
```

19. Everything should be working now.
20. To compile the sample programs/demo programs, navigate to the cpp.java executable file and right click on it, and then click **modify run configuration**



```java
import java.io.FileInputStream;
import java.io.InputStream;

import static intermediate.util.BackendMode.*;

public class cpp
{
    public static void main(String[] args) throws Exception
    {
        if (args.length != 2)
        {
            System.out.println("USAGE: cppJava option sourceFileName");
            System.out.println("    option: -convert, -execute, or -compile");
            return;
        }

        String option = args[0];
        String sourceFileName = args[1];

        BackendMode mode = EXECUTOR;

        if      (option.equalsIgnoreCase( anotherString: "-convert")) mode = CONVERTER;
        else if (option.equalsIgnoreCase( anotherString: "-execute")) mode = EXECUTOR;
        else if (option.equalsIgnoreCase( anotherString: "-compile")) mode = COMPILER;
        else

                    tem.out.println("ERROR: Invalid option.");
                    tem.out.println("    Valid options: -convert, -execute, or -compile");

                    rate a source file listing.
                    ting(sourceFileName);

                    te the input stream.
                    ream source = new FileInputStream(sourceFileName);

                    te the character stream from the input stream.
                    eam cs = CharStreams.fromStream(source);

                    om syntax error handler.
                    rorHandler syntaxErrorHandler = new SyntaxErrorHandler();

                    te a lexer which scans the character stream
                    reate a token stream.
                    r lexer = new cppLexer(cs);
                    emoveErrorListeners();
                    ddErrorListener(syntaxErrorHandler);
                    okenStream tokens = new CommonTokenStream(lexer);

                    te a parser which parses the token stream.
                    er parser = new cppParser(tokens);
```

21. In the new menu, input -compile [directory of sample programs]


Edit Run Configuration: 'cpp'
Store as project file
Build and run
Modify options Alt+M
Press Alt for field hints
Separate variables with semicolon: VAR=value; VAR1=value1
Open run/debug tool window when started
OK
Cancel
Apply

Name: cpp

java 18 SDK of 'testing' modu | cpp

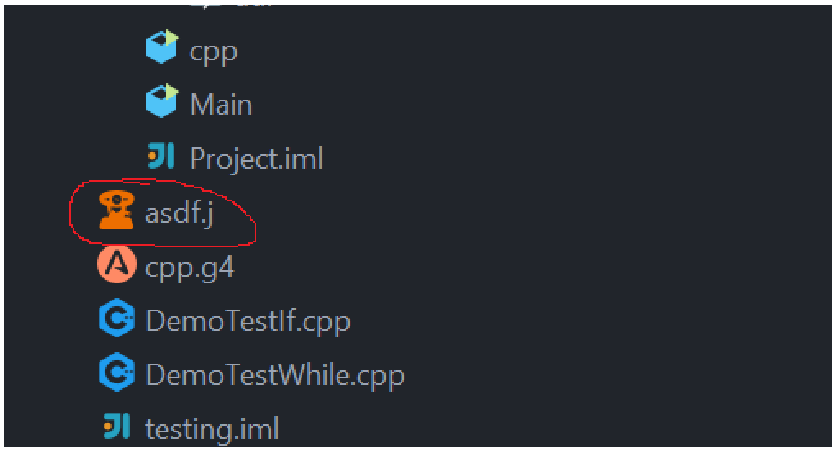
-compile DemoTestIf.cpp

Working directory: G:\CodeProjects\cs153working\testing

Environment variables:

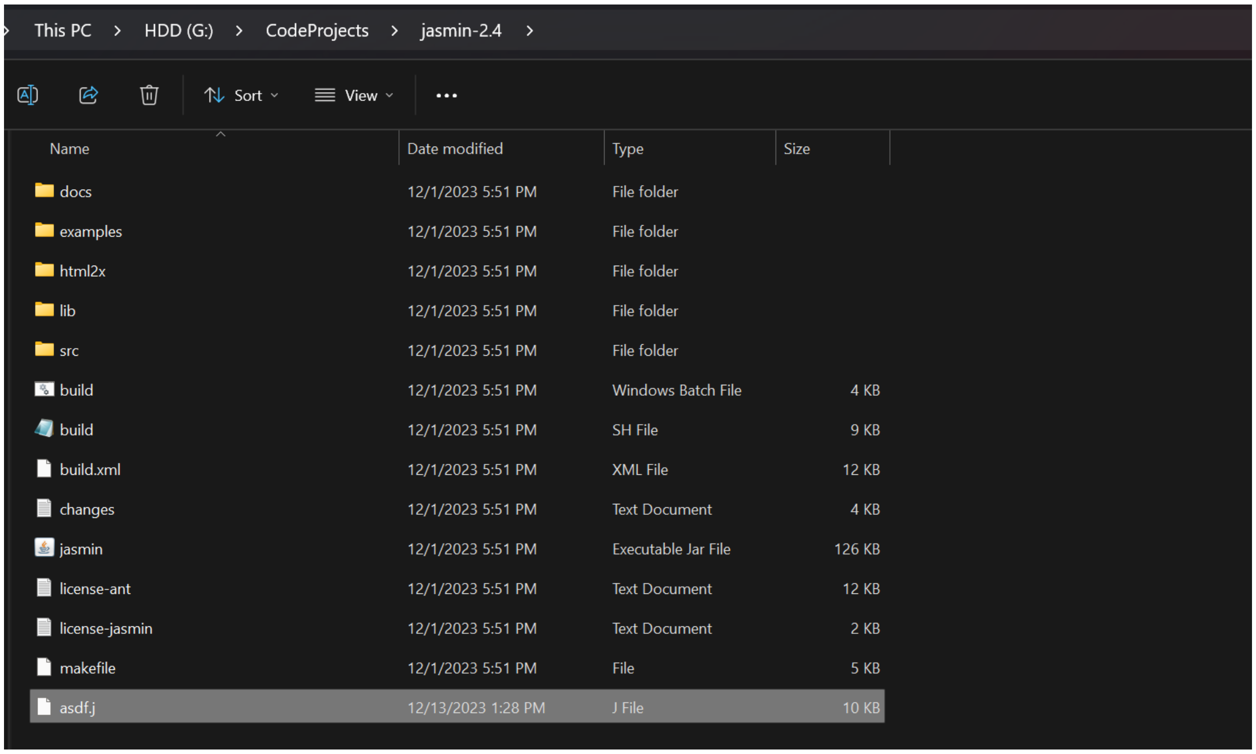
22. Click apply then click ok.
23. Afterwards, right click on the cpp.java file and click **Run cpp.main()**
24. Assuming that the file is written correctly based on our languages, everything should be in running order and the Object File jasmin file will be generated. If you do not see the

jasmin file, repeat step 14 in order to reload everything. My generated jasmin file is named asdf.j



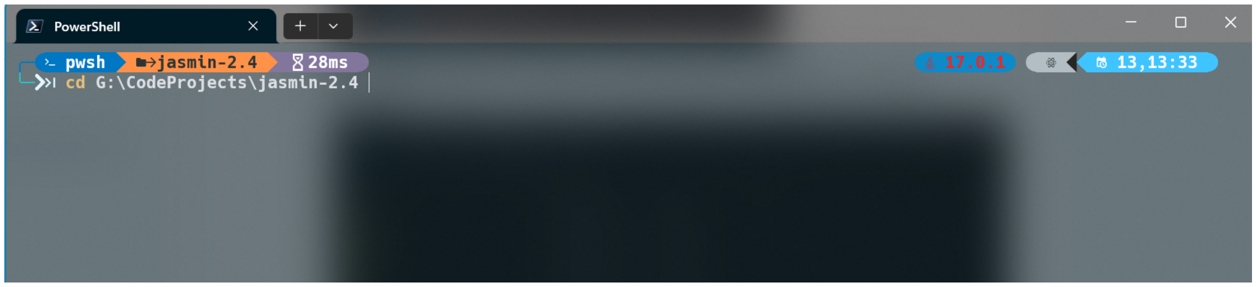
cpp
Main
Project.iml
asdf.j
cpp.g4
DemoTestIf.cpp
DemoTestWhile.cpp
testing.iml

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,



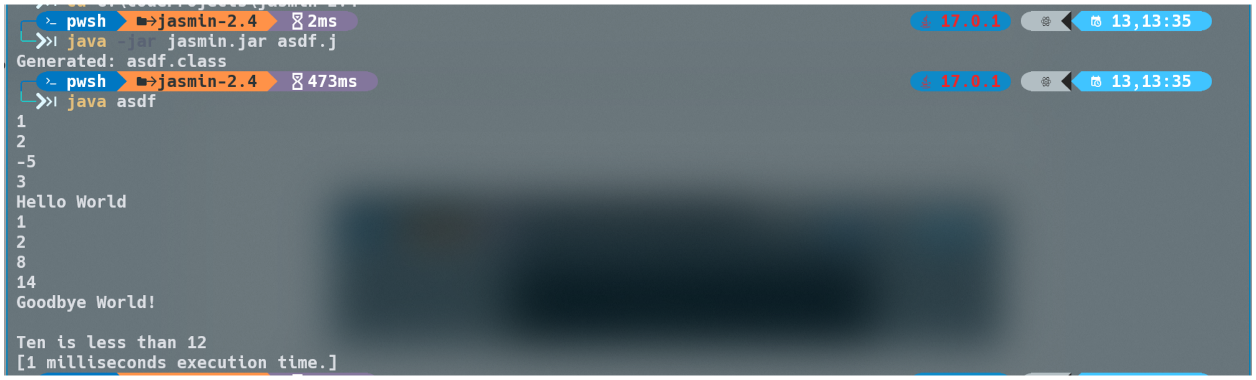
Name	Date modified	Type	Size
docs	12/1/2023 5:51 PM	File folder	
examples	12/1/2023 5:51 PM	File folder	
html2x	12/1/2023 5:51 PM	File folder	
lib	12/1/2023 5:51 PM	File folder	
src	12/1/2023 5:51 PM	File folder	
build	12/1/2023 5:51 PM	Windows Batch File	4 KB
build	12/1/2023 5:51 PM	SH File	9 KB
build.xml	12/1/2023 5:51 PM	XML File	12 KB
changes	12/1/2023 5:51 PM	Text Document	4 KB
jasmin	12/1/2023 5:51 PM	Executable Jar File	126 KB
license-ant	12/1/2023 5:51 PM	Text Document	12 KB
license-jasmin	12/1/2023 5:51 PM	Text Document	2 KB
makefile	12/1/2023 5:51 PM	File	5 KB
asdf.j	12/13/2023 1:28 PM	J File	10 KB

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



```
pwsh  jasmin-2.4  28ms                                     17.0.1    13,13:33
>  cd G:\CodeProjects\jasmin-2.4
```

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.



```
pwsh  jasmin-2.4  2ms                                      17.0.1    13,13:35
>  java -jar jasmin.jar asdf.j
Generated: asdf.class
pwsh  jasmin-2.4  473ms                                    17.0.1    13,13:35
>  java asdf
1
2
-5
3
Hello World
1
2
8
14
Goodbye World!

Ten is less than 12
[1 milliseconds execution time.]
```