Computer Science
Engineering School

Software
Engineering

Lab 03
Lexical Analysis

Francisco Ortín Soler

University of Oviedo

Objective

- Implement a C-- lexical analyzer with ANTLR

Lexical Specification

- Open the description.txt file to see a lexical description of C--
- Open input.txt to see some sample tokens to be recognized

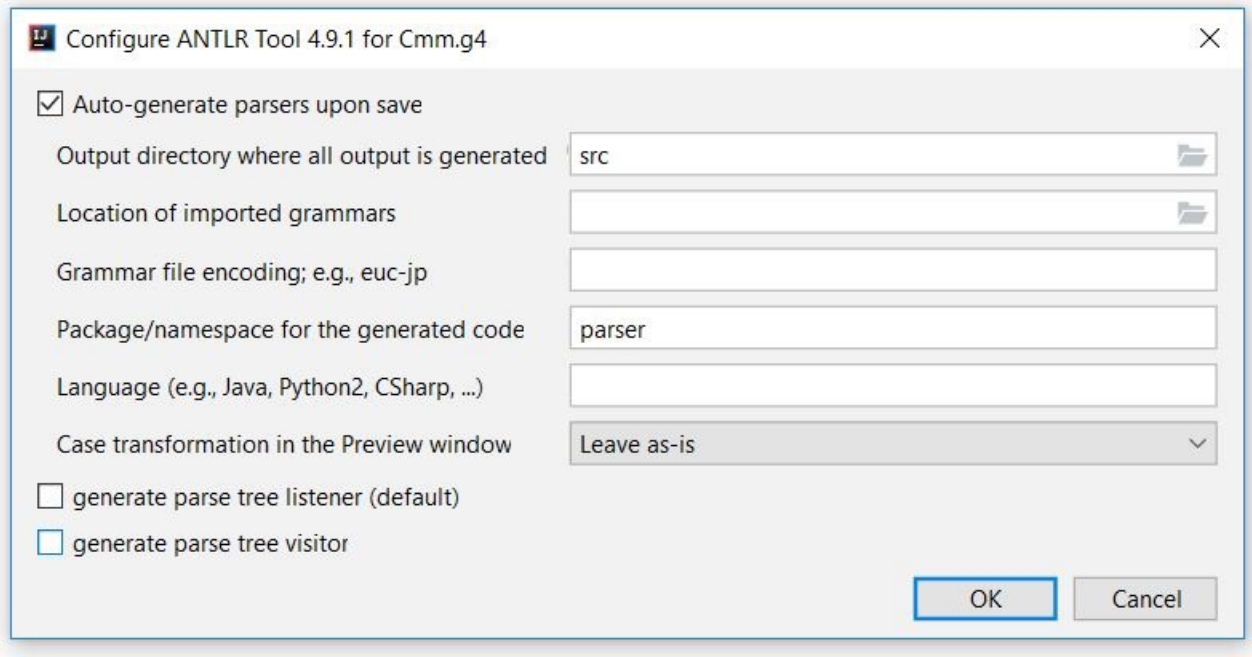
Scanner specification with ANTLR

Integrating ANTLR with the IDE

- In this course, I encourage the use of **IntelliJ**
  - It provides a wonderful **ANTLR plugin**
  - It will help you to save time
  - You can install IntelliJ **Community** for free
- You can use any other IDE, though

Demo

1. Create a new **`dlp`** Java project
2. Copy all the files to the project folder
3. Right-click over antlr-4.x-complete.jar and select "add as library"
4. Install ANTLR v4 grammar plugin by selecting File | Settings | Plugins | Marketplace
5. Right-click over Cmm.g4 and select "configure ANTLR"



Demo

6. Generate the Lexer by right-clicking over Cmm.g4 and selecting "Generate ANTLR recognizer"
7. Run the Main::main method passing small-input.txt as an input and enable asserts (-ea parameter for the VM)
   - The INT_CONSTANT token must be recognized
8. Go to Cmm.g4 and add

```
program: INT_CONSTANT
       ;
```

9. Right-click over program and select "test rule program"
   See the results

Autonomous work

1. Complete Cmm.g4 to recognize all the lexical patterns described in description.txt
   - Test them with the "test rule program" option described in the previous slide
2. Complete the LexerHelper class to obtain the semantic values of real and char constants
3. Test your lexer
   - Remove the TODO comment in Main::main
   - Rename LexerText.java.rename to LexerText.java to test your lexer
   - Run LexerText.java
     - **Remember** to enable asserts passing –ea to the VM upon execution

- Your lexer must be implemented for next lab
- Upload it to VC 5 minutes before the lab ends