

Project 1: The MiniJava Scanner

Due: Friday, October 19, 12:30 pm.

In this assignment you will extend the initial MiniJava scanner with the extensions described in the [course project description handout](#). Specifically, MiniJava's lexical structure should be extended as follows:

- Comments are now allowed, and ignored. Two styles of comment are supported: “//” to end of line, and /* . . . */ unnested block comments.
- Underscores (“_”) are now allowed in identifiers wherever letters are allowed.
- Floating-point literals are allowed. A floating-point literal is an integer part followed by either a fractional part, or an exponent part, or both. If both, then the fractional part precedes the exponent part. The integer part is one or more digits. A fractional part is a decimal point followed by one or more digits. An exponent part is the letter E or e, followed by an optional + or –, followed by one or more digits. (This is a restricted version of full Java's floating-point literal syntax.)
- "double", "for", "break", and "length" are new reserved words.
- "||" is a new operator.

In this assignment, you should only extend the scanner. You do not need to check for syntactic or typechecking errors, nor do you need to extend the parser or any other parts of the MiniJava compiler.

Do the following:

1. Extend [this specification](#) of MiniJava's lexical structure to describe the extended language, in the same style.
2. Extend `Parser/minijava.cup` and `Scanner/minijava.jflex` to scan the new tokens and comments. (You can follow [these directions](#) to get a copy of the initial MiniJava compiler implementation and set up your own SVN system.)
3. Develop test cases that demonstrate that your extended scanner works, both in cases that should now be lexically legal and in cases that should still be lexically illegal. (Since the scanner quits at the first error, you'll likely need several illegal test case files to test the different illegal cases.) The `SamplePrograms` directory contains some files that should scan after you make your changes; some of the files should scan successfully with the initial version of the MiniJava compiler.

You can use the `-scan -printTokens` options to the MiniJava compiler to just run the scanning phase and print out the tokens that it scans. See the `test_scanner` target in the `Makefile` for an example, and feel free to make your own target(s) to make running the tests you like easier and more mechanical.

Turn in the following:

1. Your extended MiniJava lexical specification.
2. Your modified `minijava.cup` and `minijava.jflex` files. Clearly identify your changes using comments.
3. Your test cases, with names of the form `name.legal.java` for test cases that should scan successfully and `name.illegal.java` for test cases that should trigger lexical errors.
4. A file `name.txt` containing names of group members

Submit the above files using the `turnin` command. The project name is `proj1`. Here's an example:

```
# turnin -c cse401 -p proj1 minijava.cup minijava.flex float.legal.java ...
```

See the [man page](#) for additional details on `turnin`.