

CS 352 – Spring 2011

Project 2

Handed out: **Feb 3th, 2011**
Due: **Feb 24th, 2011 @ 11:30pm**

Abstract:

In this project, you will construct a parser for the MiniJava language specified in the attached BNF, using JavaCC.

Setting up your code skeleton:

The code skeleton consists of only one file named `MiniJavaParser.jj`. You are required to use this skeleton code as a starting point. And you should not modify the code before line 35: `PARSER_END(MiniJavaParser)`.

Running your code skeleton:

Compile using javacc: `javacc MiniJavaParser.jj`

Compile generated java code: `javac Parser/*.java`

Run the test input: `java Parser.MiniJavaParser SampleInput.txt`

The output of your code:

You are not required to output anything. If your program parses an input file cleanly, the main function call will indicate this. If a parse error occurs, the built-in javacc tools will report it. Finally if any other exception occurs, main function will report it.

How your code should be organized:

All your work should be done in the `MiniJavaParser.jj` file.

What to submit:

Only your “`MiniJavaParser.jj`” file

How to submit your work:

To submit: `turnin -c cs352 -p p2 MiniJavaParser.jj`

To verify: `turnin -v -c cs352`

Any doubts concerning the project requirements should be raised by sending email to lib@purdue.edu. The TA may then discuss things with the instructor if necessary.

BNF for MiniJava

NON-TERMINALS

Goal ::= MainClass (ClassDeclaration)* <EOF>

MainClass ::= **class** Identifier "{" **public static void main** "(" **String** "[" "]" Identifier ")" "{"
Statement "}" "}"

ClassDeclaration ::= **class** Identifier (**extends** Identifier)? "{" (VarDeclaration | MethodDeclaration)*
"}"

VarDeclaration ::= Type Identifier ("," Identifier)* ";"

MethodDeclaration ::= **public** (**void** | Type) Identifier "(" (Type Identifier ("," Type Identifier)*)? ")" "{"
(VarDeclaration | Statement)* **return** Expression ";" "}"

Type ::= NameType
| Type "[" "]"

NameType ::= **int**
| **boolean**
| Identifier

Statement ::= "{" (Statement)* "}"
| **if** "(" Expression ")" Statement (**else** Statement)?
| **while** "(" Expression ")" Statement
| **System.out.println** "(" Expression ")" ";"
| Identifier "=" Expression ";"
| Identifier "[" Expression "]" "=" Expression ";"

Expression ::= Expression ("&&" | "||" | "==" | "!=" | "<" | ">" | "<=" | ">=" | "+" | "-" |
"*" | "/") Expression
| Expression "[" Expression "]"
| Expression "." "length"
| Expression "." Identifier "(" (Expression ("," Expression)*)? ")"
| <INTEGER_LITERAL>
| **true**
| **false**
| Identifier
| **this**
| **null**
| **new** NameType "[" Expression "]" ("[" "]")*
| **new** Identifier "(" ")"
| **!** Expression
| "(" Expression ")"

Identifier ::= <IDENTIFIER>