<http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html>

## Grammar

|  |  |  |
| --- | --- | --- |
| Program | ::= | [MainClass](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod2) ( [ClassDeclaration](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod3) )\* <EOF> |
| MainClass | ::= | "class" [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) "{" "public" "static" "void" "main" "(" "String" "[" "]" [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) ")" "{" [Statement](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod5) "}" "}" |
| ClassDeclaration | ::= | "class" [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) ( "extends" [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) )? "{" ( [VarDeclaration](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod6) )\* ( [MethodDeclaration](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod7) )\* "}" |
| VarDeclaration | ::= | [Type](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod8) [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) ";" |
| MethodDeclaration | ::= | "public" [Type](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod8) [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) "(" ( [Type](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod8) [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) ( "," [Type](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod8) [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) )\* )? ")" "{" ( [VarDeclaration](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod6) )\* ( [Statement](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod5) )\* "return" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) ";" "}" |
| Type | ::= | "int" "[" "]" |
|  | | | "boolean" |
|  | | | "int" |
|  | | | [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) |
| Statement | ::= | "{" ( [Statement](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod5) )\* "}" |
|  | | | "if" "(" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) ")" [Statement](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod5) "else" [Statement](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod5) |
|  | | | "while" "(" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) ")" [Statement](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod5) |
|  | | | "System.out.println" "(" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) ")" ";" |
|  | | | [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) "=" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) ";" |
|  | | | [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) "[" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) "]" "=" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) ";" |
| Expression | ::= | [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) ( "&&" | "<" | "+" | "-" | "\*" ) [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) |
|  | | | [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) "[" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) "]" |
|  | | | [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) "." "length" |
|  | | | [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) "." [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) "(" ( [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) ( "," [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) )\* )? ")" |
|  | | | <INTEGER\_LITERAL> |
|  | | | "true" |
|  | | | "false" |
|  | | | [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) |
|  | | | "this" |
|  | | | "new" "int" "[" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) "]" |
|  | | | "new" [Identifier](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod4) "(" ")" |
|  | | | "!" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) |
|  | | | "(" [Expression](http://www.cs.tufts.edu/~sguyer/classes/comp181-2006/minijava.html#prod9) ")" |
| Identifier | ::= | <IDENTIFIER> |

## Lexical Issues

**Identifiers:**

An *identifier* is a sequence of letters, digits, and underscores, starting with a letter. Uppercase letters are distinguished from lowercase. In this reference manual the symbol *id* stands for an identifier.

**Integer literals:**

A sequence of decimal digits is an *integer constant* that denotes the corresponding integer value. In this specification the symbol *INTEGER\_LITERAL* stands for an integer constant.

**Binary operators:**

A *binary operator* is one of

&&     <     +     -     \*

In this appendix the symbol *op* stands for a binary operator.

**Comments:**

A comment may appear between any two tokens. There are two forms of comments: one starts with */\*, ends with \*/, and may be nested; another begins with // and goes to the end of the line.*

## Sample Program

class Factorial{

public static void main(String[] a){

System.out.println(new Fac().ComputeFac(10));

}

}

class Fac {

public int ComputeFac(int num){

int num\_aux ;

if (num < 1)

num\_aux = 1 ;

else

num\_aux = num \* (this.ComputeFac(num-1)) ;

return num\_aux ;

}

}