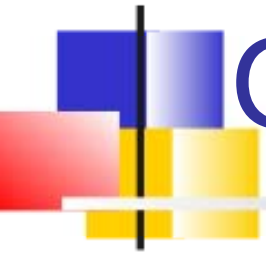

CSE 401 – Compilers

MiniJava Scanner

Automatic Scanner Generation ForMiniJava

- We use the jflex tool to automatically create a scanner from a specification file, Scanner/minijava.jflex
- We use the CUP tool to automatically create a parser from a specification file, Parser/minijava.cup
- Token classes are shared by jflex and CUP. CUP generates code for the token classes specified by the Symbol class
- The MiniJava ant build.xml file automatically rebuilds the scanner (or parser) whenever its specification file changes

Symbol Class

Tokens are represented as instances of class Symbol

```
class Symbol {
Int sym; // which token class?
Object value; // any extra data for this lexeme
...
}
```

A different integer constant is defined for each token class in the sym helper class

```
class sym {
static int CLASS = 1;
static int IDENTIFIER = 2;
static int COMMA = 3;
...
}
```

Can use this in printing code for Symbols; see symbolToString in minijava.jflex

Token Declarations in CUP

- Declare new token classes in Parser/minijava.cup, using terminal declarations
  - include Java type if Symbol stores extra data
- Examples

```
/* reserved words: */
terminal CLASS, PUBLIC, STATIC, EXTENDS;
...
/* operators: */
terminal PLUS, MINUS, STAR, SLASH, EXCLAIM;
...
/* delimiters: */
terminal OPEN_PAREN, CLOSE_PAREN;
terminal EQUALS, SEMICOLON, COMMA, PERIOD;
...
/* tokens with values: */
terminal String IDENTIFIER;
terminal Integer INT_LITERAL;
```

jflex Token Specifications

- Helper definitions for character classes and re's

  letter = [a-z A-Z]

  eol = [\r\n]

- Simple token definitions are of the form:

regexp { *Java stmt* }

regexp can be (at least):

- a string literal in double-quotes, e.g. "class", "<="
- a reference to a named helper, in braces, e.g. {letter}
- a character list or range,in square brackets ,e.g. [a-z A-Z]
- a negated character list or range, e.g. [^\r\n]
- **.** (which matches any single character)
- *regexp regexp*,*regexp*|*regexp*,*regexp**,*regexp*+, *regexp*?, (*regexp*)

jflex Specifications (cont.)

- *Java stmt* (the accept action) in a token specification is typically:
  - return symbol(sym.CLASS); for a simple token
  - return symbol(sym.CLASS,yytext()); for a token with extra data based on the lexeme stringyytext()
  - empty for whitespace