The goal of this project is to implement type checking and JVM code generation for the Java programming constructs that were introduced in *j*-- as part of Project 3 (Parsing). To compile the *j*-- compiler, run the following command:

```
$ ant clean compile jar
```

To compile a *j*-- program `P.java`, run the following command:

```
$ $j/j--/bin/j-- P.java
```

Run the following command to run the *j*-- program `P.class`:

```
$ java P
```

Problem 1. (*Double Basic Type*) Add support for the `double` basic type.

```
basicType ::= boolean | char | int | double
```

```
$ $j/j--/bin/j-- Double.java
$ java Double 1.0
3.141592653589793
```

Problem 2. (*Operators*) Add support for the logical or operator `||`, the assignment operators `-=`, `*=`, `/=`, `%=`, the prefix operator `--`, and the postfix operator `++`.

```
assignmentExpression ::= conditionalOrExpression // must be a valid lhs
                           [ ( =
                             | +=
                             | -=
                             | *=
                             | /=
                             | %=
                             ) assignmentExpression
                           ]

conditionalOrExpression ::= conditionalAndExpression { || conditionalAndExpression }

unaryExpression ::= ++ unaryExpression
                  | -- unaryExpression
                  | - unaryExpression
                  | + unaryExpression
                  | simpleUnaryExpression

postfixExpression ::= primary { selector } { -- | ++ }
```

```
$ $j/j--/bin/j-- Operators.java
$ java Operators 23 5
5
true
```

Problem 3. (*Blocks*) Add support for static and instance blocks.

```
classBody ::= { { static block // static block
                | block         // instance block
                | modifiers memberDecl
                }
              }
```

```
$ $j/j--/bin/j-- Blocks.java
$ java Blocks 6
1182
```

Problem 4. (*Interface Type Declaration*) Implement support for interface declaration.

typeDeclaration ::= modifiers (classDeclaration | interfaceDeclaration)

classDeclaration ::= class <identifier> [extends qualifiedIdentifier]
[implements qualifiedIdentifier { , qualifiedIdentifier }]
classBody

interfaceDeclaration ::= interface <identifier> // can't be final
[extends qualifiedIdentifier { , qualifiedIdentifier }]
interfaceBody

interfaceBody ::= { { modifiers interfaceMemberDecl } }

interfaceMemberDecl ::= (void | type) <identifier> // method
formalParameters ;
| type variableDeclarators ; // fields; must have inits

```
$ $j/j--/bin/j-- Interface.java
$ java Interface 5
25
125
```

Problem 5. (*Conditional Expression*) Add support for conditional expression (e1 ? e2 : e3).

conditionalExpression ::= conditionalOrExpression
[? assignmentExpression : conditionalExpression]

```
$ $j/j--/bin/j-- ConditionalExpression.java
$ java ConditionalExpression 50 42
50 is greater than 42
```

Problem 6. (*For Statements*) Add support for a for-statement, both the basic for-statement and the enhanced for-statement.

statement ::= block
| <identifier> : statement
| if parExpression statement [else statement]
| for ([forInit] ; [expression] ; [forUpdate]) statement
| for (type <identifier> : expression) statement
| while parExpression statement
| return [expression] ;
| ;
| statementExpression ;

forInit ::= statementExpression { , statementExpression }
| [final] type variableDeclarators

forUpdate ::= statementExpression { , statementExpression }

```
$ $j/j--/bin/j-- ForStatements.java
$ java ForStatements
55
55
```

Problem 7. (*Exception Handlers*) Add support for exception handling, which involves supporting the `try`, `catch`, `finally`, `throw`, and `throws` clauses.

```
statement ::= block
            | <identifier> : statement
            | if parExpression statement [else statement]
            | while parExpression statement
            | try block
                { catch ( formalParameter ) block }
                  [ finally block ] // must be present if no catches
            | return [expression] ;
            | throw expression ;
            | ;
            | statementExpression ;
```

```
memberDecl ::= <identifier> // constructor
                 formalParameters
                   [ throws qualifiedIdentifier { , qualifiedIdentifier } ] block
             | ( void | type) <identifier> // method
                     formalParameters
                       [ throws qualifiedIdentifier { , qualifiedIdentifier } ] ;
             | type variableDeclarators ; // fields
```

```
interfaceMemberDecl ::= ( void | type) <identifier> // method
                          formalParameters
                            [ throws qualifiedIdentifier { , qualifiedIdentifier } ] ;
                      | type variableDeclarators ; // fields; must have inits
```

```
$ $j/j--/bin/j-- ExceptionHandlers.java
$ java ExceptionHandlers 42
42: The answer to life, the universe and everything!
Done!
$ java ExceptionHandlers 43
43
Done!
```

Files to Submit

1. `j--.zip` (*j*-- source tree as a single zip file)
2. `report.txt` (project report)

Before you submit:

- Make sure you create the zip file `j--.zip` such that it only includes the source files and not the binaries, which can be done on the terminal as follows:

  ```
  $ cd $j/j--
  $ ant clean
  $ cd ..
  $ tar -cvf j--.tar j--/*
  $ gzip j--.tar
  ```

- Make sure the files `$j/j--/lexicalgrammar` and `$j/j--/grammar` are updated with the syntactic changes you have made to the *j--* language.

- Make sure your report isn't too verbose, doesn't contain lines that exceed 80 characters, and doesn't contain spelling/grammatical mistakes