STANDARD PROJECT REQUIREMENTS

This project involves four main phases: scanning, parsing, semantic analysis and code generation (optionally).

Two extra credit points if the project contains the last phase (code generation).

1. Scanning

One lexical analyser for the proposed language has to be constructed. The parser must detect the tokens of the language and report some lexical errors.

For a project having the type B or C, a scanner generator can be used (lex of Flex for C/C++ and JLex or JFlex for Java).

2. Parsing

A parser for the proposed language has to be constructed. The parser has two main goals:

a) to check whether the input is a syntactically correct program, and

b) to generate an abstract syntax tree (AST) that records all important information about the program (the intermediate representation); to construct AST, some actions must to be added to the parser.

3. Semantic analysis

A two-pass static-semantic analyzer for the programs represented as abstract-syntax trees has to be constructed:

a) the name analysis method, and

b) type checking method.

The semantic analyzer will traverse the tree constructed in parsing phase to build a table of the symbols contained in the programs and to decorate AST with semantic information. Name analysis perform several tasks, like: building symbol table, finding bad declarations, multiply declared names and uses of undeclared names, etc. They are used to generate useful error messages in this phase.

The job of the type checker is to determine the type of every expression in the abstract syntax tree and verify that the type of each expression is appropriate for its context.

4. Code generation

Finally, the abstract syntax tree will be traversed again to perform final code generation. The code will be generated for a specified assembly language. The target language can be ASM (the assembly language for Intel processors) or Jasmin assembly language (for Java virtual machines).

Remark. Sun has not defined an assembler format for Java byte code. Jasmin is an assembly language for Java byte code. Jasmin is an assembler that reads as input a Jasmin assembly program and produces as output a Java class file ready to be run on a JVM.