 Compiler Project for CPSC423  
 Fall 2000

The programming assignments are based on a language called “Rat00F” which is defined as follows.

Rat00F

1) Lexical Conversion:
The Lexical units of a program are identifiers, keywords, integers, operators and other separators. Blanks, tabs and new lines (collectively, “white space”) as described below are ignored except as they serve to separate tokens. A white space is required to separate otherwise adjacent identifiers, keywords and integers.

Identifier is a sequence of letters, digits and “_”. The first character must be a letter.
The last character must not be a “_”. Upper and lower cases are different. Some identifiers are reserved for use as keywords and may not be used otherwise. Examples of keywords are: function, int, while, repeat etc.

Integer is a sequence of digits.
All Comments are enclosed in <*      *>.

2) Syntax for Rat00F
The following BNF grammar is the reference grammar for Rat00F.

<Function Definition> ::= function <Identifiers> ( <Opt Parameter List> ) <Body>
<Opt Parameter List> ::= <IDs> | ε
<Body> ::= <Opt Declaration List> % <Statement List> %
<Opt Declaration List> ::= <Declaration> | ε
<Declaration> ::= int <IDs> ; | boolean <IDs> ;
<IDs> ::= <Identifier> | <Identifier> ; <IDs>
<Statement List> ::= <Statement> | <Statement> <Statement List>
<Statement> ::= <Compound> | <Assign> | <If> | <While> | <Repeat> | <Return> | <Print> | <Read>
<Compound> ::= { <Statement List> }
<Assign> ::= <Identifier> = <Expression>;
<If> ::= if ( <Condition> ) <Statement> endif | if ( <Condition> ) <Statement> else <Statement> endif
<While> ::= while ( <Condition> ) <Statement> endwhile
<Repeat> ::= repeat <Statement> until <Condition> endrepeat
<Return> ::= return; | return <Expression> ;
<Print> ::= printf ( <Expression> );
<Read> ::= scanf ( <IDs>);
<Condition> ::= <Expression> <Relop> <Expression>
<Relop> ::= == | != | > | < | <= | >=


3) Semantics

Rat00F is a conventional imperative programming language (A small subset of C). Some of the semantics are as follows. Further semantics will be given later.
   a. All variables must be declared before use.
   b. Arithmetic expressions have their conventional meanings. Integer division ignores any remainder. Division by zero is an error.
   c. There is a parameter-less function called “main()” at the end of the program.

4) A Sample Rat00F Program

```c
function convert(lower; upper)

   int step; fahr; celsius;

   %
   scanf(step);
   fahr = lower;
   while (fahr < upper) {
      celsius = 5*(fahr – 32)/9;
      printf (fahr);
      printf (celsius);
      fahr = fahr + step;
   }
endwhile
%
```
VERY, VERY IMPORTANT

For each programming assignment, you should turn in the followings:

1) The cover page.
2) A hardcopy of:
   a) About 2 pages of documentation including:
      i) Problem Statement
      ii) How to use your program
      iii) Discuss the design of your project and choice of the algorithms
      iv) Describe the limitations of your program (e.g. maximum # of lines etc.)
      and any shortcomings.
   b) Source code listing with proper comments for each procedure, section if necessary.
   c) Test cases: Find at least 3 test cases and the results of the testing.
3) A diskette containing all above features. And an executable file of your program under DOS and/or Windows.

Notes: 1. I must be able to run your program in order to give you a grade.
2. I will accept late project, however, there will be some deductions, i.e., 0.5 deduction for the first day and 0.1 each day you are late (from max of 10).
3. If you turn in a program that cannot be run or if your diskette does not contain the executable, same deduction will apply unless there is a compelling reason. Make sure that your program runs in any of the open labs.
4. Not turning in the documentation will case deduction of 1.0