Oct. 4, 2003 **ICS 313-02 (031)** Dr. Mamdouh Najjar

Fundamentals of Programming Languages

Programming Assignment # 1
Interpreter for a Simple Language (*SIM v.1.0*)

Due Nov. 1, 2003

A. **Given *SIM v.1.0 s*yntax described in BNF rules as follows:**

```
<program> → Start_Program <program_name>  < blocks>
                End_Program <program_name>;
<program_name> → <id>
<blocks> → <datablock> <statblock>
<datablock> → Begin_Define <datatypes> End_Define;
<statblock> → Begin_Block  <stmt_list> End_Block ;
<stmt_list> → <stmt> | <stmt> ; <stmt_list>
<stmt> → <assign> | <read> | <write>
<assign> → <var> := <expr>
<read> → Read  ( <varlist> )
<write> → Write  ( <varlist> )
<expr> → <expr> + <term> | <expr>  - <term> | <term>
<term> → <term> * <factor> | <term> / <factor> | <factor>
<factor> → ( <expr> ) | <var> | <value>
<var> → <id>
<id> → <char> | <char> <id>
<char> → A | B | … | Z | a | b | c | … | z | _
<varlist>  → <id> | <id> , <varlist>
<datatypes>  → <datatype> | <datatype> <datatypes>
<datatype>  → <vartype> | <consttype>
<vartype>  → Var <varlist> : <varkind> ;
<varkind> → Integer | Real
<consttype> →      Constant <id> = <value>;
<value>  → <sign_integer> | <real>
<real>  → <sign_integer> . <integer>
<sign_integer> → <sign> <integer>
<sign> → + | - | NULL
<integer> → <digit> | <digit> <integer>
<digit> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

B. **Your Program**

Your program should read a program written in ***SIM*** and executes its statements assuming all statements are syntacticaly correct. This grammer will allow the generation of programs like:

```
Start_Program Salary_Calculation

Begin_Define
Var  Basic_Salary , Grade      :   Integer ;
Var  Adj_Salary                :   Real ;
Var  Ret_Fund_Amount           :   Real ;
Var  Comp_A_Amount             :   Real ;
Var  Net_Salary                :   Real ;

Constant     Year_Allowance = 450 ;
Constant     Ret_Fund = 0.09 ;
Constant     Comp_Allowance = 0.25 ;
Constant     Trans_Allowance = 600 ;
End_Define;

Begin_Block
Read ( Basic_Salary , Grade ) ;
Adj_Salary :=  ( Grade * Year_Allowance ) + Basic_Salary ;
Ret_Fund_Amount := Adj_Salary * Ret_Fund ;
Comp_A_Amount  := Basic_Salary * Comp_Allowance ;
Net_Salary := Adj_Salary – Ret_Fund_Amount + Comp_A_Amount +
Trans_Allowance ;
Write ( Net_Salary )
End_Block ;
End_Program Salary_Calculation ;
```

Important notes:

1- Your interpreter must read the above ***SIM*** program from a file named ***SC1.sim***.
2- Your ***SIM*** program reads the input data from ***SIM.in***
3- Your ***SIM*** program writes program results into an output file named ***SIM.out***.
4- The interpreter source file ***SIM.ext*** must include the following:

- your name, ID number
- Course title, number, and section number
- the statement of the problem (the Grammar, etc.)
- a brief summary of your interpreter implementation method
- a dictionary of all global variables (name, type, and usage)
- for each method or procedure or function must include the following:
  - Usage of the procedure or function
  - Meanings of the input variables
  - Meanings of the output variables

5- No assignment will be accepted after the due date.
6- Hand out a print out and a diskette contain all files or electronically submit it through the WebCT.