# CS 4000/5353 Compilers – Extra Credit

## Spring 2010

```
P       →  program D S end
D       → IL D1 | begin
D1      → array[CL D | integer D
IL      → id IL1
IL1     → , IL | :
CL      → cons CL1
CL1     → , CL | ]

ID      → id ID1
ID1     → [EL | ε                    {:=+-*/ ),<=>] and or ; end fi od esac do then else}
EL      → E EL1
EL1     → , EL | ]
IDL     → ID IDL1
IDL1    → , IDL | )

E       → T E1
E1      → + T E1 | - T E1 | ε        {),<=>] and or ; end fi od esac do then else }
T       → F T1
T1      → * F T1 | / F T1 | ε        {+ - ),<=>] and or ; end fi od esac do then else }
F       → ID | cons | exp(E,E) | (E)

C       → X C1
C1      → or X C1 | ε   {do then )}
X       → Y X1
X1      → and Y X1 | ε                {or do then )}
Y       → E Y1 | not(C) | [C]
Y1      → < E | > E | = E

S       → ID := E S1 | read(IDL S1 | write(IDL S1 | readln(IDL S1 | writeln(IDL S |
              case C do S  M S1| while C do S od S1 | if C then S S2|
              foreach id in id do S  od S1 | with D S end S1
S1      → ; S | ε                     {end : od fi esac }
S       → fi S1 | else S fi S1

M       → : C do S M | esac
```

Format:

1. Source lines are 80 characters in length and tokens may appear anywhere in the line.
2. A token may not be broken across line boundaries.
3. A comment is any sequence of characters beginning with /* and ending with */
4. Tokens for reserved words must be separated from one another by at least one blank.

5.  Upper and lowercase characters are equivalent.

6.  Spaces, all operators, and all special characters are delimiters.

To do:

1.  Generate Selection Sets for the grammar.

2.  You are to write a phased implementation of a compiler to translate a program written in the language described above into an object language to be defined at a later date.  Your compiler will consist of a lexical analyzer (or scanner), syntax analyzer, semantic analyzer, and code generator.

3.  Provide lexical errors (e.g., encountering a symbol that is not a valid token of the language), syntax errors (we will discuss in detail when we talk about syntax analysis), and semantic errors (e.g., referencing an undeclared variable, improper use of subscripts, etc.).