**Definition of SEL**

(Simple Expression Language)

1. **Overview.**

SEL is a simple expression language for integer and real (floating point) numbers. Integers are promoted (coerced) to reals in all expressions. Variables are implicitly declared to have type double. The interpreter performs no error recovery.  
  
An expression can be an assignment expression or a numeric expression. The value of an assignment expression is simply the value being assigned. Chained assignments are allowed; i.e., you can write “A = B = 5”. A single variable is an expression; its value is the value of the previously assigned expression or 0 if it has never previously appeared on the left side of an assignment expression.

1. **Lexical Considerations.**

no comments

lexical grammar:

identifier = letter ( letter | digit )\*

numericLiteral = digit+ ("." digit+)?

= + - \* / ( ) // one character tokens

1. **Grammar.**

program = ( expression )\* .

expression = ( assignExpr | numericExpr ) EOL .

assignExpr = identifier "=" expression .

numericExpr = ( addOp )? term ( addOp term )\* .

addOp = "+" | "-" .

term = factor ( multOp factor )\* .

multOp = "\*" | "/" .

factor = numericLiteral | identifier | "(" expression ")" .

1. **Output.**

There are no I/O statements in SEL. The interpreter computes the value of every statement. When run interactively, the interpreter writes out the value of every line. When the interpreter runs a program, the interpreter writes out the value of the last line.

1. **Sample Programs.**
2. For the following single expression:  
    -5 + 10\*9\*8  
   the output from would be  
    715
3. For the following list of expressions:  
    Pi = 3.14159265359  
    r = 4  
    A = Pi\*r\*r  
   the output would be  
    50.26548245744