CIS 424 Project 2 Fall 2021

(Due: Oct. 21)

In this project you are asked to write an interpreter which uses the top-down recursive-descent method and inherited/synthesized attributes to parse and evaluate a very simple programming language. The tiny strong-typed language's grammar is given below.

```
<prog>       ::=  <let-in-end> { <let-in-end> }
<let-in-end> ::=  let <decl-list> in <type> ( <expr> ) end ;
<decl-list>  ::=  <decl> { <decl> }
<decl>       ::=  id : <type> = <expr>  ;
<type>       ::=  int | real
<expr>       ::=  <term> { + <term> | - <term> }
<term>       ::=  <factor> { * <factor> | / <factor> }
<factor>     ::=  <base> ^ <factor> | <base>
<base>       ::=  ( <expr> ) | id | number | <type> ( id )
```

The interpreter (i.e. `let.py`) should be written in Python3. It takes one input file which contains the program to be executed. The input file name is given from the command line. For example,

```
spirit % ./let.py sample.tiny
```

Note that your program must run using above format on our Linux workstations. Otherwise, no credit will be given.

The interpreter `let.py` reads the program file `sample.tiny`, checks the syntax and outputs the result for each let-in-end if the program is legitimate; otherwise, the interpreter prints "Error". Below is a test example:

```
let x : int = 5 ;
in
    int ( x + x * x )
end ;

let r : real = 10.0 ;
    pi : real = 3.1416 ;
in
    real ( pi * r * r )
end ;
```

Your program should display

```
30
314.6
```

For another example,

```
let x : int = 7 ;
    y : real = 3.0 ;
```

```
in
    real ( ( real ( x ) + y ) * ( real ( x ) - y ) )
end;

let x = 8 ; in ( x + y ) end ;
```

Your program will output

```
40.0
Error
```

Turnin

Each group (two students) needs to hand in a typed document which includes the description of your code, experiences in debugging and testing, etc. The cover page should contain your photos, names, 7-digit CSU IDs, and your computer login-id your group used to turnin the files. You also have to submit your program and your photos electronically by using the following turnin command on grail:

```
turnin -c cis424s -p proj2 let.py photo1_lastname1.jpg photo2_lastname2.jpg
```

Start on time and good luck. If you have any questions, send e-mail to `j.sang@csuohio.edu`.