

Spring 2025 CIS 524 Project (Group Project)

(Due ~~Feb-23~~ March 2)

In this project you are asked to write an interpreter which uses the top-down recursive-descent method to parse and evaluate a very simple programming language. The tiny 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> } |
                  if <cond> then <expr> else <expr>
<term>       ::= <factor> { * <factor> | / <factor> }
<factor>     ::= ( <expr> ) | id | number | <type> ( id )
<cond>       ::= <oprnd> <oprnd> |
                  <oprnd> <= <oprnd> |
                  <oprnd> > <oprnd> |
                  <oprnd> >= <oprnd> |
                  <oprnd> == <oprnd> |
                  <oprnd> <> <oprnd>
<oprnd>      ::= id | intnum
```

The interpreter (parser_XXXXXXX.py where XXXXXXX is your CSU ID) should be written in Python with similar structures to the c++ version in the Blackboard. It takes one input file (sample.tiny) which is a text file consisting of the statements of the above grammar. The input file name sample.tiny is given from the command line. The interpreter parser_XXXXXXX.py reads the program file sample.tiny, checks the syntax and outputs the result if the program is legitimate; otherwise, the interpreter prints "Error".

Below is a test 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
```

For another 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 ;

let a : int = 3 ;
b : real = 0.5 ;
c : real = b * b ;
in
    real ( if a > 5 then b + 1.1 else c )
end ;

```

Your program should display

```

30
314.16
0.25

```

Submission

Option 1:

1. Upload the following files to Blackboard [No Zip files]:
 - parser_XXXXXXX(csuid).py
 - A screenshot showing the output of your code
 - A brief explanation of your code

Option 2:

1. Upload your assignment to GitHub and share the repository URL on Blackboard.
2. Your GitHub repository must include:
 - parser_XXXXXXX(csuid).py
 - A screenshot showing the output of your code
 - A README file explaining your code