

Programming Assignment 3, Parser Project

CSC 413 Fall 2014

Due by Oct. 10, 2014 23:55pm

Modify the Parser in the compiler package given for assignment 2 according to the notes below:

Parse the new rules specified in the modified grammar and build the indicated AST's

PROGRAM -> 'program' BLOCK ==> program

BLOCK -> '{ D* S* }' ==> block

D -> TYPE NAME ==> decl

-> TYPE NAME FUNHEAD BLOCK ==> functionDecl

TYPE -> 'int'

-> 'boolean'

-> 'float' <-----**NEW RULE**

-> 'char'

FUNHEAD -> '(' (D list ',')? ')' ==> formals

S -> 'if' E 'then' BLOCK 'else' BLOCK ==> if

-> 'if' E 'then' BLOCK ==> if <-----**NEW RULE**

-> 'while' E BLOCK ==> while

-> 'return' E ==> return

-> BLOCK

-> NAME '=' E ==> assign

-> 'do' BLOCK 'while' E ==> repeat ---**NEW RULE WITH NEW TOKEN AND AST NODE**

E -> SE

-> SE '==' SE ==> =

-> SE '!=' SE ==> !=

-> SE '<' SE ==> <

-> SE '<=' SE ==> <=

SE -> T

-> SE '+' T ==> +

-> SE '-' T ==> -

-> SE '|' T ==> or

T -> F

-> T '*' F ==> *

-> T '/' F ==> /

-> T '&' F ==> and

F -> '(' E ')'

-> NAME

-> <int>

-> NAME '(' (E list ',')? ')' ==> call

-> <float> <-----NEW RULE

-> <char> <-----NEW RULE

-> <scientificN> <-----NEW RULE

-> '!' E <----- NEW RULE

-> '-' E <----- NEW RULE

NAME -> <id>

Following demonstrates a test case (note that the token printout is no longer included):

```
program {  
    do {  
        float f  
        f = 1.2  
    } while b  
}
```

-----AST-----

```
Program  
  Block  
    do  
      Block  
        Decl  
          FloatType  
          Id: f  
        Assign  
          Id: f  
          float: 1.2  
      Id: b
```

Since you will not be using constrainer or codegen you must:

- Comment out the code, as indicated in the Compiler.java class.
- Comment out the constrainer/codegen import statements in Compiler.java
- When you build a project DO NOT include constrainer/codegen files

Submission

REMOVE ALL DEBUG statements that are not required.

SUBMIT THE src DIRECTORY, along with the .jar file, as explained in the Reader/posted at ilearn for earlier assignment

YOU NEED to SUBMIT A DOCUMENTATION WITH CLASS DIAGRAM AND DESCRIPTION (will be discussed in the class)

More notes:

Q. What kind of test cases should be used to test the Parser?

You should try to test for every conceivable possibility within the language. Of course, you cannot test for every possible X language program, but try to make sure you test for as many scenarios as you can think of, including things that you are sure will not work, and things that you may not be sure about. There is probably a reason why this lab is worth more points than the previous one, given that it probably took you less time to make the modifications required.

Q. Do any additional Lexer modifications need to be made?

Yes. You will have to make some changes to the lexer to handle the new reserved words.

Q. What main method should be run?

Use the main method in Compiler.java to test and produce output for this project. Follow the instructions in the main method of Compiler.java and comment out (delete) the appropriate parts so that it will run. If you don't comment the right things out, the other components called by the compiler (constrainer, for example) will fail because they have not been modified to handle the changes that we have made to the language.