

## SDSU CS530 Spring 2017 Assignment #3

### Design, develop, test, and deliver a parser for assignments and expressions

**Project Deadline:** 1730, 5 May 2017

You shall work on this project on a team of 2 people.

#### Specification:

You shall develop a grammar and implement a parser which recognizes valid statements. Your program shall read input from a file named in.txt, scan the input, and determine if the statement(s) is/are valid. Your program shall print out the statement, and either pass/fail into a file named out.txt. If it failed, you shall print out why (in the same out.txt file).

**Note:** **id** ::= identifier, **exp** ::= expression, **op** ::= operator, **char** ::= character

Your compiler will recognize the following as valid statements:

- **assignment**
- **expression**

**assignment** shall have the form:

**id** = **exp** ;

**expression** shall have the form:

**id op id {op id}\***

**note** - each **id**, **op**, = (equal symbol), and ; (semi-colon symbol) will have a space as it's precedent and antecedent.

An **id** shall be made up of any combination of **digits**, **chars**, and \_ (the underscore symbol). The first position of the identifier must contain a **char** or an underscore.

A **digit** is one of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 0

A **char** is one of a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z

An **op** is one of:

+, -, \*, /, %

The file ex.txt also posted in the assignment folder in the course's Blackboard contains examples of both good and bad statements

For Extra Credit: Provide for the use of parenthesis to group any **expression**, including nested parenthesis. E.g., the following are all valid **expressions**:

- **id op (id op id) op id**
- **id op id op (id op id)**
- **id op (id op (id op id) )**
- **id op (id op (id op (id op id) ) )**
- **id op id op (id op id) op ( (id op id) op id)**

**note** - The open parenthesis will have a space as it's precedent but it may or may not have a space immediately following it. The converse is true for the closing parenthesis (guaranteed a space as it's antecedent but not necessarily preceding it).

### **Turning In Your Work:**

When ready to turn this project in, use the turnin procedure posted on the course Blackboard

### **Development Environment:**

You may develop your code using **C, C++, Python, lex/yacc, or Lisp**, or any combination of these...provided that you are using tools with the same versions found on edoras.

In addition to the normal requirements for your README file (see notes on Blackboard), your README file shall also contain:

- The grammar (use BNF) describing what a valid statement is
- Program inputs and program outputs

All other rules/instructions apply (name in every file, no cheating, etc).