Comp 333 Programming Project # 1 Part B ( 15 pts)

Due: Sept 26

**Modify the Recursive Descent Parser for the Assignment Language**

Download the recursive descent parser for the Assignment Language from the COMP 333 Moodle site. Run the parser on a set of “good” strings and “bad” strings to get a feel for how it works. Then modify the parse so that it prints helpful error messages. It should print out a helpful message for each error found. For example if a ‘;’ is missing, it could print “Missing semicolon at location x “. [Note: The parser only finds the first error it encounters and then quits. That is fine.]

The easiest way to modify the parser is to modify the error method so that it has a string parameter with the error message. When the error method is called, the callee can insert an error message.

**Test Cases: ( All input strings end with a $ to indicate end of string)**

x=6;y=5$

x=78$

y = 0; z = 8$ There are lots of spaces here

y=z=4$

x=5;y=2

x =8;$

cat =8;y=7$

b=6$

x=8$

x=5 $; These is a space between the 5 and the $

**Hand in:**

1. Submit source code (one java file called “AssignLangParser.java” with your name and date embedded in file) to Moodle. Instructor will test your code with random test cases.
2. Hard copy of source code with required test cases runs and results to be turned in class. If your program is only partially working, you must also submit a typed explanation of which errors your program identifies and which errors is does not.

**Assignment Language**

**<prog> 🡪 <stmtList>**

**<stmtList> 🡪 <stmt> | <stmt> ;<stmtList>**

**<stmt> 🡪 <id> = <num>**

**<id> 🡪 x | y | z**

**<num> 🡪 0|1|2|3|4|5|6|7|8|9**

**Token List: { x,y,z,0,1,2,3,4,5,6,7,8,9, ; , =}**