```
//-----------------------------------------
// Recognizer for simple Java Class grammar
// Written by Jagan Chidella 9/3/2017
//
// to run on Athena (linux) -
//    save as:  JClassRecognizer.java
//    compile:  javac JClassRecognizer.java
//    execute:  java JClassRecognizer
//
// EBNF Grammar is -
// Java Class             <jClass > ::=  <className>  B  < varlist> {<method>}   E
// Class Name             <className>  ::= C|D
// Variable List          <varlist>  ::=  <vardef> {, <vardef>};
// Variable Declaration   <vardef>  ::=<type> <var>
// Type                   <type> ::= I | S
// Variable Names         <var> ::= V|Z
// Class Method           <method> ::= P <type> <mname> (<vardef> {,<vardef>}) B
//                                          <stmnt> <returnstmnt> E
// Method Name            <mname> ::=  M|N
// Statement              <stmnt> ::= <ifstmnt>|<assignstmnt>|<whilestmnt>
// If statement           <ifstmnt> ::= F <cond> T B {<stmnt>} E [L B { <stmnt> } E]
// Assignment Statement   <assignstmnt> ::= <var> = <digit>;
// While Statement        <whilestmnt> ::= W <cond> T B <stmnt> {<stmnt>} E
// Condition              <cond> ::= (<var> == <digit>)
// Digit                  <digit> ::= 0|1|2|3|4|5|6|7|8|9
// Return Statement       <returnstmnt> ::= R <var>;
//
// KEY:      C and D are names of two Java classes
//           I stands for Integer
//           S stands for String
//           B stands for { brace
//           E stands for } brace
//           V and Z are names of two variables
//           P stands for the word: public
//           F stands for If
//           T stands for Then
//           L stands for Else
//           W stands for While
//           R stands for Return
// Valid (Legal)strings to ENTER for testing class method containing following
// statements: ENTER the corresponding string.
// Response 'Legal' validates string. Response 'Errors Found' invalidates string.
// IF statement:                      CBIV,SZ;PIM(IV)BF(V==9)TBV=8;ERV;EE$
// IF THEN ELSE statement:            CBIV,SZ;PSM(SZ)BF(V==9)TBV=8;ELBV=0;ERV;EE$
// WHILE statement:                   CBIV,SZ;PIM(IV)BW(V==0)TBV=9;Z=2;ERV;EE$
// Assignment:                        CBIV,SZ;PIM(IV)BV=8;RV;EE$
// Invalid Strings: CBIV,SZ;PIM(IV)BF(V==9)TBV=8;E;RV;E$
// Invalid Strings: CBIV,SZ;PIM(IV)BF(V==9)TBV=8;E;RVEE$
//-------------------------------------------------------------

// Create FIRST and FOLLOW table 70% points by 10/4. Show to me, get it corrected.

// Then Write the Parser 30% points by 10/9. Show to me, get it corrected.

// Recursively ☺ improve both for full points. See page 2,3 for snippet of program.
```

```
public class JClassParser {

                static String inputString;
                static int index = 0;
                static int errorflag = 0;

                private char token()
                { return(inputString.charAt(index)); }

                private void advancePtr()
                { if (index < (inputString.length()-1)) index++; }

                private void match(char T)
                { if (T == token()) advancePtr(); else error(); }

                private void error()
                {
                  System.out.println("error at position: " + index);
                  errorflag = 1;
                  advancePtr();
                }
              //----------------------
                private void jClass()
                {

                      className();

                      match('B');
                      varlist();
                      while(token() == 'P') {
                            method();
                      }
                      match('E');

                }


// WRITE YOUR REST OF THE PARSER HERE

//----------------------
                private void start()
                {
                  jClass();
                  match('$');

                  if (errorflag == 0)
                    System.out.println("legal." + "\n");
                  else
                    System.out.println("errors found." + "\n");
                }
              //----------------------
```

```
                public static void main (String[] args) throws IOException
                {
                      JClassParser rec = new JClassParser();

                  Scanner input = new Scanner(System.in);

                  System.out.print("\n" + "enter an expression: ");
                  inputString = input.nextLine();


                  rec.start();
                }
}
```