COP5555 Spring 2012
Project 2

Assigned Jan 31
Due: Feb 14 at 3pm

This project is to implement a recursive descent parser for the attached grammar. Use your scanner from Project 1 to recognize tokens. This means that before proceeding, you should correct any errors found in Project 1 Errors detected while parsing should throw a SyntaxException, code given below. Your parse method should catch that exception and return it as an argument. Thus, for the purposes of this assignment, the parse method should return null for legal input. In the case of an error, it should return the SyntaxException object containing the offending token for input with an error. (In Project 3, we will revise the parser so that it generates an abstract syntax tree.)

The TokenStream class should be initialized with char array containing the input characters. Your scanner from Project scans the input.

For this project, your parser's parse method will simply return null if the input is legal, and return the thrown SyntaxException object if there is an error.

The way these classes would be used in a parser is as follows

```
TokenStream stream = new TokenStream(inputChars);
Scanner scanner = new Scanner(stream);
scanner.scan();
SimpleParser parser = new SimpleParser(stream);
Exception e = parser.parse();
System.out.println(e.getMessage()); //or whatever you want
}
```

Turn in a jar file called P2.jar containing all of the sources needed to run SimpleParser, including those from Project 1. We will recompile and call the public methods from our test program. Do not change the signatures of these methods or the package declarations or you will break the test script.