COP5555 Spring 2012
Project 3

Assigned Feb 14
Due: March 1 at 3pm

Modify your parser from project 2 to either return an AST or, if the input program is not syntactically valid, throw a SyntaxException.   Your parser should be callable with the sequence:

```
TokenStream stream = …
Parser parser = new Parser(stream);
AST ast = parser.parse();
```

Your class should match the following:

```
package edu.ufl.cise.cop5555.sp12;
import …
public class Parser{
             public Parser(TokenStream stream){
               …
             }
             public AST parse() throws SyntaxException {
                    …
             }
…
}
```

The AST should represent the attached abstract grammar.

A jar file containing all of the classes needed to represent the AST is provided. You will need to look at the sources of these classes--they are provided to save you time since creating them is straightforward and tedious. Providing these classes also makes grading easier. In later projects, you may want to add fields and methods to the classes, for this project, you should not modify them.

The package also contains a class ToStringVisitor which visits the AST and creates a String representation of it.   The static method printAST(AST ast) will print your AST.

Turn in a jar file called P3.jar containing all of the sources needed to run Parser, including those from previous projects.   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.