COP5555 Spring 2015
Assignment 3

Due: Monday, March. 9 at 10am.

The goal of this assignment is to modify your parser to create an AST, and optionally for extra credit, to do some simple error recovery.

AST creation:

Use the classes in the cop5555sp15.ast package that are provided in the attached jar file. The Parser should be in a class called Parser.java with a public method parse() that returns a Program object if the input is correct, or null otherwise. This method should catch SyntaxExceptions. The SyntaxException should be added to a List<SyntaxException> that can be retrieved with the getExceptionList method. If you do not do error recovery, then your parser catch exceptions in the parse method and return after an exception is encountered. Here is an example parse method:

```
List<SyntaxException> exceptionList = new ArrayList<SyntaxException>();

public Program parse() {
	Program p = null;
	try {
		p = Program();
		if (p != null)
			match(EOF);
	} catch (SyntaxException e) {
		exceptionList.add(e);
	}
	if (exceptionList.isEmpty())
		return p;
	else
		return null;
}
```

The Program method returns a cop5555sp15.ast.Program object. This class extends ASTNode, which contains a field firstToken. The purpose of this field is to record the location in the input. As you proceed to do this assignment, look at the attached AST specification to see which AST objects should be created in what part of the grammar.

Error handling

This section is for extra credit.

If an error occurs while parsing a list of statements, the catch block should try to skip tokens until we find a ; or end of the construct and then continue parsing. For example, consider

<Closure> ::= { <FormalArgList> -> (<Statement> ;)* }

If an error occurs while parsing the list of statements, we want to catch the exception, throw away tokens until we get to a SEMICOLON, save the exception in the exceptionList, consume the SEMICOLON, then continue parsing statements.

If an RCURLY occurs, we save the exception and return. If an EOF occurs, the catch block should rethrow the exception. Always allow the block that handles the exception (i.e. does not rethrow it) to insert it in the exceptionList. Errors in Declarations and Statements in Blocks can be handled similarly.

Turn in

cop5555.jar with containing ALL of the the Java source files required to execute TestParser.java and TestParserErrorHandling.java. **Make sure that your jar file contains your sources**.

> *To ensure a successful submission, double check that your files were actually submitted. Also, double check the contents of your jar file. If you generate the jar file using the eclipse export function, be aware that by default, it only includes class files in the jar; you will need to explicitly tell it to include the source files. To grade, we will unjar, compile, and execute your code with your test cases, and with ours.* ***Make sure that you do not break the grading script.***Hints: