CSC435/535: Assignment 2
(Due: 18 October 2012)

Introduction

This assignment asks you to attach tree building actions to the rules of the Cb grammar so that an abstract syntax tree (AST) is created for any Cb program.

Assignment Description

The introduction said it all. The AST must be implemented using the classes and node tags which have been provided in the `CbAST.cs` file (see below).

You may need to adjust the grammar rules in order to simplify the task of building the AST. This is permitted provided that the language accepted by the parser is not changed.

The Provided Materials

- You are provided with everything needed to start this assignment. If you wish, you may discard your solution to Assignment 1 and start over with the files provided on the conneX website.
- Note: the lexer supports a slightly different set of keywords from before. If you keep your lexer from assignment 1, the set of keywords obnviously needs be modified. (The supplied lexer does not support nested comments.)
- Note: the grammar rules in the `CbParser.y` file have been adjusted in a few ways. One of these changes was to add methods with non-void result types to the language. (This was an accidental omission from the language described in the Cb Language document.) Other changes were:
  To use left-recursive rules for a list of 'using' clauses, because left-recursion made it easier to attach semantic actions for building the AST.
  To introduce a new rule Identifier → Ident which simplifies creating a leaf node for the identifier (the code needs to appear in only one place in the grammar instead of many places).
- The supplied source code files are listed in the table below.

File	Description
`cbc.cs`	The main program which invokes everything else.
`CbLexer.lex`	The specification file to be processed by gplex
`CbParser.y`	The grammar file to be processed by gppg
`CbAST.c`	The classes used for building the AST
`CbVisitor.cs`	The parent class for the Visitor pattern
`CbPrVisitor.cs`	A visitor for printing the AST
`CbType.cs`	Classes used for describing Cb datatypes
`AST-DataStructure.pdf`	Explanation of the AST structure

- For Windows users, three batch command files are provided:

`rungplex.bat`	Runs gplex on `CbLexer.lex`
`rungppg.bat`	Runs gppg on `CbParser.y`
`build.bat`	Builds `cbc.exe` from all the C# source files

Linux or Mac OS X users can copy the commands into a Makefile or into shell scripts.

Submission Requirements

1. You must provide exactly one file. It must be a zip file or gzipped tar file which contains the source code files `cbc.cs CbLexer.lex CbParser.y CbAST.c CbVisitor.cs CbPrVisitor.cs CbType.cs`. If you added more C# files to your project, include those too.
2. Also include a file named README.txt which identifies the team members. If you have any comments you want to share about problems with the assignment, this is an appropriate place to supply the comments.
3. Important: do *not* include any files generated by gplex or gppg in your submission.
4. Again, the project is to be completed in teams of either 2 or 3 persons. The ideal size is 2 people. The teams do not have to contain the same members as for Assignment 1.