CSCI 331
Compilers

So what have we done this semester?

I. **Scanning**. You should know what tokens are. If I give you the grammar for a language you should be able to find the tokens in it. I will not ask you to write a scanner or part of one on the midterm. **See Feb. 3**

II. **Parsing in general**. You should know what a parse tree is. If I give you a grammar, you should be able to say whether a specific string is accepted by that grammar, and you should be able to draw a parse tree for that string. **See Feb. 5, Feb. 8**

III. **Recursive descent parsing**. You should know what the recursive descent descent technique is. If I give you a grammar you should be able to write (in pseudocode or your favorite language) the recursive descent parse method for one of the grammar symbols of that language. **See Feb. 10, 12**

IV. **Table-driven parsers**. We talked about many different parsing techniques. The most important of those is the LR-parser. If I give you an LR parse table you should be able to use it and a stack to parse a given string. I will not ask you to create a parse table, but you should be able to find teh First and Follow sets for a grammar. **Feb. 15, 22, 24**

V. **Type checking**. If I give you a parse tree for a program in a language like BPL, you should be able to add links to it that would be added in the type checking state of a compiler. **Feb. 26**

VI. **Attribute grammars**/ If I give you an attribute grammar and an expression you should be able to draw the parse tree for the expression and evaluate the attributes in the tree. **March 7**