CS712/CS812 Fall 2016 Final Exam

- **Place your answers on separate paper.**
- **The exam is closed book and notes.**
- **Please keep all electronic devices turned off and out of reach.**
- **This exam has 7 questions and a total of 100 points.**
- **For partial credit, show your work!**
- **When turning in your exam, please fold your papers in half, left-to-right, and write your name on the outside.**
- ***I will return your exam to your Kingsbury mailbox unless you tell me not to.***

Question 1 (10 points)

Compute the set of null-deriving nonterminals, the First sets and the Follow sets for the following grammar, in which S is the start symbol:

$S \rightarrow X\ Y\ Z$
$X \rightarrow$
$X \rightarrow c\ Z$
$Y \rightarrow$
$Y \rightarrow a\ Y\ b$
$Z \rightarrow$
$Z \rightarrow c\ Z$

Question 2 (10 points)

Compute the LALR(1) configuration sets for the grammar of Question 1.

Question 3 (40 points)

Is the grammar of Question 1 LR(0)? Is it SLR(1)? Is it LALR(1)? Is it LR(1)? **Carefully state and justify each of your answers**.

Question 4 (10 points)

Produce a regular expression to describe the language generated by the grammar of Question 1. If this cannot be done, *carefully* explain why not.

Question 5 (10 points)

Prove or disprove the following statement: All SLR(1) grammars generate regular languages.

Question 6 (15 points)
Consider the following Go program:

```
package main;

type S1 struct { x, y int32 }

var x *S1

var a int32;

func main(){
  a = 2;
  x = new(S1);
  x.y = 18 + 12 * a;  // <-- trace compilation of this statement
  println(x.y);
}
```

Trace your compiler's processing of the third assignment statement in the function. By *trace* I mean: show the AST that would be produced by the parser, the AST after semantic analysis, and the code that would be generated. Be sure to explicitly show DEREF nodes and to label the types of expression nodes. You may discuss the generated code in terms of pseudo-code for LLVM.

Question 7 (5 points)
In our Go compiler,

- Explain the purpose of the classes derived from the abstract class Type.
- Explain the purpose of the classes derived from the abstract class TypeTree.
- Why did we need both of these sets of classes?