# CSCI 3155 – Exam I – Take-home portion (part 1)

This exam is due on Thursday, March 19th in class. Keep your answers short and concise. Use your own words and examples – do not simply copy from the text or Internet. What you turn into me must by typed, not hand-written, save for those occasions when a drawing or formula would take too much time to attempt in Word or other word-processing program.

There will be a second take-home part that will be based on this material that will be short problems. I should have that up this weekend.

1. What is one example of a lack of orthogonality in the design of C?
2. What primitive control statement is used to build more complicated control statements in languages that lack them?
3. What is the name of the category of programming languages whose structure is dictated by the von Neumann computer architecture?
4. What is an example of two language design criteria that are in direct conflict with each other?
5. What role does the symbol table play in a compiler?
6. Why is the von Neumann bottleneck important?
7. Why were linguists interested in artificial intelligence in the late 1950s?
8. What language was designed to describe the syntax of ALGOL 60?
9. What is a nonprocedural language? Provide an example.
10. What populates the Smalltalk world? Why is Smalltalk historically so important?
11. LISP began as a pure functional language but gradually acquired more and more imperative features. Why?
12. Describe the operation of a general language generator.
13. Describe the operation of a general language recognizer.
14. What is the difference between a sentence and a sentential form?
15. Distinguish between static and dynamic semantics.
16. What purpose do predicates serve in an attribute grammar?
17. What is the difference between a synthesized and an inherited attribute? Provide an example.
18. How is the order of evaluation of attributes determined for the trees of a given attribute grammar?
19. What is the primary use of attribute grammars?

20. Explain the primary uses of a methodology and notation for describing the semantics of programming languages.

21. Why can machine languages not be used to define statements in operational semantics?

22. Describe the two levels of uses of operational semantics.

23. Provide three reasons why syntax analyzers are based on grammars?

24. Why is lexical analysis is separated from syntax analysis.

25. Define lexeme and token and when, where and how are they used in the compilation process?

26. What are the primary tasks of a lexical analyzer?

27. What is a state transition diagram and how are they used in the compilation process?

28. Why are character classes used, rather than individual characters, for the letter and digit transitions of a state diagram for a lexical analyzer?

29. What are the two distinct goals of syntax analysis?

30. Describe the differences between top-down and bottom-up parsers. Make sure you describe when and where each is used.

31. Define static, stack-dynamic, explicit heap-dynamic, and implicit heap dynamic variables. What are their advantages and disadvantages?

32. Define lifetime, scope, static scope, and dynamic scope and give examples of each.

**Flex-Bison Problem**

Starting with the simple Flex/Bison program I did in class, convert the calculator from infix to postfix. By way of example, my program produced the following output:

```
>  5 * 4.0;
>  20.0
```

The postfix version should work as follows:

```
>  4   9  +
>  13
```

Control-d should terminate the program.

You simply need to send me your modified versions of the ex1.l and ex1.ypp files that I used in my example. Of course, you should make sure they work first. I will use g++ to test them. Please call them lastname_firstname_postfix.l and lastname_firstname_postfix.ypp.