CS 4700 Written Homework 2 (10 points)

Note, these exercises may be done in groups of one, two, or three. Working with someone else is strongly recommended. If more than one person is involved, list all the names on ONE set of answers. Groups may change throughout the term.

Working in groups is a BIG plus for you. Take advantage of it. If you work in groups, you must work in the group for the ENTIRE assignment. It is considered cheating if you work with someone else for some of the answers, but turn in an individual copy of the answers. It is an all or nothing situation. You can’t work together on some questions and alone on some. Sometimes I see an individual whose name is listed in two groups. This is strictly forbidden and is considered cheating. You cannot work in two groups. Assignments are due at the beginning of class and should be typed.

Topic 2

1. Language developments often occur as a result of (or in step with) new hardware or machine developments. For the developments below, discuss their relationship to programming language development.
   - personal computers
   - fast random access memory
   - multiple processors
   - the internet

2. Define the term abstraction, as it is used in computer science. List some of the more common examples.

3. What do we mean by the term backwards compatible?

4. Many computer scientists believe that languages with strong compile-time type checking are better than those that are typeless, dynamically typed, or are weakly type checked. Discuss the reasons for this view.

5. If strictly-checked data types are seen as good, discuss whether augmenting a language with many more primitive data types is better. Consider, in particular, the possibility of incorporating into a language such as Java many new numerical types such as packed decimal of various precisions, scaled arithmetic, and new types to hold values representing distance, mass and time. How would these additions affect the readability and reliability of programs?

6. What is the defining feature of early assembly languages, in contrast to compiled languages?

7. Object-oriented languages do not lead to the most efficient programs at run time. So why are object oriented languages so widely used?

8. Name the language that was developed by Niklaus Wirth which distilled the ideas in ALGOL into a small programming language that was used for teaching programming?

9. Name the person who led the original design team for FORTRAN.

10. What is the main reason that the languages PL/I, ALGOL68, and ADA had limited popularity and were not widely adopted by programmers?
Topic 3

1. When a language feature is independent of the context in which it is used, we term it
   a. portable
   b. orthogonal
   c. readable
   d. unified

2. Some languages distinguish between uppercase and lowercase letters in identifiers. What are the pros and cons of this decision?

3. Some programming languages such as C++, allow for operator overloading. Others, such as Java, do not. Evaluate this decision in regards to each of the following. Make sure your answer reflects that you understand the design feature.
   a. Readability
   b. Writability
   c. Orthogonality
   d. Simplicity

4. C++ requires variable names to be declared. Consider the following change. Variable names do not need to be declared unless they are a structured type (records, arrays) or their type cannot be determined from context. Notice, we are still assuming static typing. Discuss this change from the point of view of
   - Readability
   - Writability
   - Orthogonality
   - Efficiency
   - Reliability

5. In C++, integers can be assigned to float variables, but not vice versa. What design principle does this violate? Would it be a good idea to allow floats to be assignable to integer variables? Why do you think the rule is the way it is?

6. One of the major problems in programming language design is the lack of understanding of many human factors that affect language use, from psychology to human-machine interaction ("ergonomics"). One example of this is that programming errors made by humans are far from random, and language design should attempt to prevent the most common errors. List three common errors users make in C++ and state how they could have been prevented by language design. Hint: Try googling "common programmer errors C++".

7. Given your experience with Ruby, how would you rate the learnability of Ruby?

8. How would you rate Ruby in terms of regularity? Give examples to explain.
9. Do you agree with the statement, "There really are only a few good ideas in language design."? Explain.