Homework 4 Due March 21, 2016

1. Indicate the binding time for each of the following decisions in C/C++.

* + The number of built-in types. **The number of built-in types is typically bound at the language design time, though it may be increased by certain implementations.**
  + The variable declaration that corresponds to a particular variable reference. **The variable declaration that corresponds to a particular variable reference is bound at compile time, as C uses static scope.**
  + The maximum length allowed for a constant character string. **The maximum length allowed for a constant character string is bound at language implementation time because C does not have nested subroutines.**
  + The address of a particular library routine. **The address of a particular library routine is bound at link time in most systems, though it may not be known until load time or even run time in systems that perform dynamic linking.**
  + The total amount of space occupied by program code and data. **The total amount of space occupied by program code and data is bound at run time.**

2. In Fortran 77, local variables are typically allocated statically. In C++, they are typically allocated in the stack. In some other languages such as lisp, they are at least partially in the heap. What accounts for these differences? Give an example C++ routine that would not work correctly if local variables are allocated statically. **Fortran does not allow recursion, which allows local variables to be allocated statically. C++ allows recursion, which requires local variables to be allocated in the stack. Other languages have dynamically sized array constructs, which requires heap to be used for local variables.**

**An example C++ routine that would not work correctly if local variables are allocated statically would be the following:**

**int fib(int n)**

**{**

**if ( n == 0 )**

**return 0;**

**else if ( n == 1 )**

**return 1;**

**else**

**return (fib(n-1) + fib(n-2) );**

**}**

3. For the following C++ program, indicate where the variables a, b, c, d, f, and g allocated (stack, heap or static data?).

**a, b: Static Data**

**c, f: Stack**

**d, g: Heap**

4. Consider the following pseudocode. What does the program print if the language uses static scoping? What does it print with dynamic scoping?

**Static Scoping: 10 10 3.5 4.5 10**

**Dynamic Scoping: 10 10 3.5 3.5 3.5**

5. Consider the following pseudocode, which uses dynamic scoping. What does the program print if the language uses shallow binding? What does it print with deep binding?

**Shallow Binding: 5 14**

**Deep Binding: 10 7**