1. Language of choice: Java  
     
   The number of built-in functions in Java is determined when the language is designed. That is why they are called “built-in” functions; they are a part of the language.  
     
   Variable declarations are bound at compile time. An integer officially becomes an integer at compile time because it is bound to the integer keyword.  
     
   The maximum length of a String is determined when the language is being implemented.   
     
   The referencing environment for a subroutine (if possible in Java) would be bound at compiles time. The compiler needs to know where the code to execute exists in the stack.  
     
   The address of a particular routine is also bound at compile time for the same reasons as above.  
     
   The total amount of space occupied the program cannot be determined until run time. When the program is run, java allocates a certain amount of memory.
2. In C a static variable can be declared locally in a function but persist past the lifetime of the function. In C a global variable can have a local shadow variable, in this case the global variable is live but not in scope. In C++ non-public member variables of an object of a class are live but not in scope when the execution is not inside a member of that class.
3. A) Every time Brad inserts a new item into the list, a new pointer to a list is created. Therefore, every time a new item is added, essentially a new list is added. Then when it is deleted, we are only deleting the last list we created, leaving all other lists behind.  
     
   B) In this example, Brad is deallocating the memory for the list before he is reassigning the reference. This means the reference no longer points to what he thinks it does.
4. Static Output: 2, 1, 1  
   Dynamic Output: 2, 1, 0 (or whatever empty int is initialized to).
5. Shallow Binding: 1, 1, 2, 2, 3  
   Deep Binding: 1, 1, 0, 3, 0, 0
6. Static Scope: 3  
   Deep Binding: 3  
   Shallow Binding: 5