**Design Patterns**

**Homework 2**

* I want you to implement a parallel program structure to my ArrayStack example from the lecture notes, but the entire thing should be based around a LinkedStack. You may use my start on LinkedStack below.
* (30 points) Write the StackMemento for LinkedStack.
* (30 points) Write factory Methods for creating mementos and for creating iterators on your LinkedStack.
* (40 points) Write StackIterator and operator << for Stack which uses the StackIterator.  
   class CharLinkedListPair  
  {  
  public:  
   char info;  
   CharLinkedList next;  
   CharLinkedListPair( char newInfo, CharLinkedList newNext )  
   : info( newInfo ), next( newNext )  
   {  
   }  
  };  
  typedef class CharLinkedListPair \* CharLinkedList;  
  class LinkedStack  
  {  
  private:  
   CharLinkedList head;  
  public:  
   LinkedStack()  
   : head( 0 )  
   {  
   }  
   void push( char c )  
   {  
   head = new CharLinkedListPair( c, head );  
   }  
   char pop()  
   {  
   char c = head->info;  
   CharLinkedList p = head;  
   head = head->next;  
   delete p;  
   return c;  
   }  
   void print( ostream & out )  
   {  
   for ( CharLinkedList p = head; p != 0; p = p->next )  
   out << p->info << ' ';  
   }  
   ~LinkedStack()  
   {  
   CharLinkedList temp;  
   for ( CharLinkedList p = head; p != 0; )  
   {  
   temp = p;  
   p = p->next;  
   delete temp;  
   }  
   }  
  };