CS 135
Bubblesort Handout

Bubblesort is a sorting strategy that compares adjacent values in an array, interchanging them if they are not in the desired order. Each time a complete pass is made through the list one value is "bubbled" to its proper position. If n is the number of elements in the list, n-1 passes will ensure that it has been sorted. It is not a very efficient method, but it works!

This version of bubblesort will sort a list of integers into descending order. It expects the list to be sorted and the number of elements in the list. It returns the sorted list.

```
void bubblesort(int list[ ],int count)
// Sort an array of integers into descending order.
// Parameters:
//     list: array of integers to be sorted
//     count: (integer) number of values in the array
// Value passed back: sorted list
{
   int temp;   //place holder when values are interchanged
   for (int i=0; i < count-1; i++)
       for (int j=0; j < count-(i+1); j++)
         if (list[j] < list[j+1])
         {
            temp = list[j];
            list[j] = list[j+1];
            list[j+1] = temp;
         }
}
```

Conversion to an ascending sort
To convert the bubblesort function into an ascending sort, change the relational operator in the if statement from less than (<) to greater than (>).

Modifying the function to sort an array of non-integers

- change the data type of the formal parameter - list - to reflect the data type of the array to be sorted
- change the data type of temp to match the data type of the elements in the array
- make sure the appropriate values are compared in the if statement for an ascending or descending sort
    - if an array of records (structs) is being sorted, the appropriate field (member) of the array element must be compared

Return to Handouts