CSCD211 In-class / Lab – Selection Sort for Linked List
5/22/2014

Name: Aaron Jones

Consider the following code sorting an array of Comparables:

```
//  Selection sort for an array of Comparable objects
   public static void selectionSort(Comparable[] array)
   {
      int curPos;
      int indexSmallest;
      int start;
      Comparable temp;

      for (start = 0; start < array.length - 1; start++)
      {
         indexSmallest = start;
         for (curPos = start + 1; curPos < array.length; curPos++)
         {
            if (array[indexSmallest].compareTo(array[curPos]) > 0)
            {
               indexSmallest = curPos;
            }
         } // end for

         temp = array[start];
         array[start] = array[indexSmallest];
         array[indexSmallest] = temp;
      } // end for
   }
```

1. Write a non-static 'sort()' method that will sort a Linked List using Selection Sort.

   public void sort(LinkedList linkList)

{

Node smallest;

Node curr;

Node start;

for(start = head.next; start.next != null; start = start.next;)

{

smallest = start;

```
        for(curr = start; curr != null; curr = curr.next)
        {
          if(curr < smallest)
          {
            smallest = curr;
          }
        }

        int temp = start.value;
        start.value = smallest.value;
        smallest.value = temp.value;
    }

}
```

CSCD211 In-class / Lab – Selection Sort for Linked List
5/22/2014

2. Key-in your solution and run in debug and test for:
- Empty linked list
- List with only one node
- List with several nodes

Print your listing and submit on paper in class – see Canvas for due date.