Name: ____

Problem	Points	Score
1a	25	
1b	25	
1c	25	
1d	25	
Total	100	

Notes:

(1) For this exam you are allowed to open a terminal window on your computer, you are allowed to web surf with Google, but you cannot use online chat or other interactive services.

(2) The first step in this exam is to create a workspace in the following directory:

/data/courses/ece_1111/current/exams/ex_03/lastname_firstname

(3) Set the permissions using "chmod -R u+rwx,g-rwx,o-rwx <lastname_firstname>" so only you have read and write permission to this directory. Create three subdirectories within this directory: p01, p02 and p03. You will use these for parts 1, 2, 3 and 4 respectively. Put ALL your code in these directories. Do not touch your files after the exam is over.

(4) You must use a make file, a header file and a main program file named p01.cc (or p02.cc). All other code needs to go into an implementation file called p01_00.cc (or p02_00.cc).

Problem No. 1: In this extended problem, we are going to develop a fully functional doubly linked list class. Your ultimate goal will be to read this data (just an example of course):

```
ECE_1111   11  -1.000
ECE_2222   22  10.000
ECE_3333   33  5.000
```

and sort it according to the floating-point value which is the third value on the line. You are going to write four simple programs.

1a) In directory p01, create a program that reads the above data and stores it in a MyData object. The goal of this program is to provide a debug framework for your code and to demonstrate your basic data class works.

Create a class called MyData that contains the necessary elements to store one line of the information above. MyData should have protected data that consists of a character string, an integer and a floating-point value. It should have four methods: (1) default constructor; (2) constructor with three arguments; and (3) a print method that displays the contents of a MyData object. You can optionally add a fourth method called assign that assigns values to an object based on the arguments. Your constructor with arguments can call this method.

You need not store these values from each line of the file in an array. Your main program should read a line, set the object's values, and continue. You can delete the existing object or declare it inside your I/O loop.

Reading and parsing of the data must occur in your main program. Do not embed this in the class. We will reuse this code.

Your solution should have a make file, a header file, a main program file called p01.cc and an implementation file called p01_00.cc.

1b) Build on your solution to (1a) by copying your code to directory p02. Add a Node class. This class should contain MyData as protected data and manage the pointers necessary for a doubly-linked list. Create a driver program that reads the above file line by line and creates independent Node objects. They do not have to be linked yet. Your Node class should have a default constructor, a constructor that takes a MyData object as an argument, and a print method that prints its internal data.

1c) Build on your solution to (1b) by copying your code to directory p03. Add a doubly linked list class called DoublyLinkedList. Modify your driver program for (1b) so that it reads the above file line by line and adds nodes to the list. Implement a print method that displays the contents of your list. Do not worry about cleaning up memory. Focus on creating the list using an insert method and iterating over the list using a print method.

1d) Implement a sort method. Copy your code from (1c) to directory p04. Add a sort method to the class. Demonstrate that it works using the print method of (1c). Sort based on the floating-point number – the third argument. Sort in ascending order. Use your print method in (1c) to demonstrate sorting is working. Show the list before and after sorting.

NOTE: If you use cin and cout, you will get a zero on this exam ☹ You must use fopen, fclose, fprintf and fscanf for your I/O functions.