ECE114 Final Exam Sample (C++)

Your full name is _____

Honor Pledge: (Required by the University of Rochester on all exams)
I affirm that I will not give or receive any unauthorized help on this exam, and that all work will be my own.
Your signature: _____

Complete a C++ program for each of the following three questions**.** This is a closed-book exam; however, you can bring a letter/A4-sized cheat sheet (two sides) with notes for your reference. During the exam, you **CANNOT** access the internet, communicate with your classmates, or get any help from anyone (e.g., no phone calls, emails, messages, etc.). **Any violation of academic honesty will get 0 from this exam and be reported to the university.** You can have **1 hour and 40 minutes** to complete them. **DO NOT DELETE ANY FILES, COPY ANY FILES, MOVE ANY FILES, OR CHANGE FILE NAMES ON YOUR USB DRIVE.** After you finish the exam, please stay at your seat first. You can raise your hand. **A TA will help check your files and collect your question sheets & USB drive.** Then you can leave.
Please remember to save your source files frequently during the exam. You also must write short but efficient comments to explain your variables and methods. The total score is 100 points. You will get 40% from your best program, 30% from the second-best, and 30% from the third. If you get 100, 90, and 80 points from the three programs, your score is 100 * 40% + 90 *30% + 80 * 30% = 91. *Good luck on your final exam!!*

1. (LastName_FirstName_Q1.cpp) Write a C++ program to ask the user to enter book titles until q is entered. Then sort book titles in the alphabetical order (an uppercase letter is prior than a lowercase letter) and output them in the new order. Your program should be able to have an output exactly the same as the following sample run:

```
Enter book titles for sorting.
Enter a title (q to exit): War and Peace
Enter a title (q to exit): Harry Potter and the Sorcerer's Stone
Enter a title (q to exit): Harry Potter and the Half-Blood Prince
Enter a title (q to exit): Don Quixote
Enter a title (q to exit): Ulysses
Enter a title (q to exit): q
After sorting in the alphabetic order, the book titles are:
Don Quixote
Harry Potter and the Half-Blood Prince
Harry Potter and the Sorcerer's Stone
Ulysses
War and Peace
```

Hint:
I recommend you to create an empty vector of string objects and use its *push_back* member function to add elements. The general form of defining a vector is:

```
std::vector<typeName> vt(n_elem);    // You need to include <vector>
```

The *push_back* member function takes a new element as an argument.
You may use any sorting algorithm with the string vector's overloaded operators (>, <, =). You also need to use nested *for* loops.

(*Continued on next page*)

2. (LastName_FirstName_Q2.cpp)
Write a C++ program to compare the volumes of four gift boxes. You have to use the following struct declaration:

```
struct Box{
   std::string name; // receiver's full name
   double length;   // inch
   double width;    // inch
   double height;   // inch
   double volume;   // cubic inches
};
```

In the main function, you need to create four Box structs (or an array of four Box structs) using the information from the following table.

	Box 1	Box 2	Box 3	Box 4
name	Spider-Man	Superman	Batman	Captain America
length	3.5	50.5	12	20
width	6.5	60.0	18	20
height	5	72.5	16	5

Write a function that passes a Box <u>by reference</u> and <u>sets the volume member to the product of the three dimensions</u>. In the main function, your program should compare the volumes of the four boxes and **find the maximum one.**
A sample output should be as follows:

```
Box information:
Spider-Man's box: 113.75 cubic inches
Superman's box: 219675 cubic inches
Batman's box: 3456 cubic inches
Captain America's box: 2000 cubic inches

The receiver of the maximum box is Superman.
The volume of the maximum box is 219675 cubic inches.
```

(*Continued on next page*)

3. (Burger.h, Burger.cpp, and LastName_FirstName_Q3.cpp)
   Write a C++ program for Burger Queen's online order. You have to define a class called **Burger**.

   In the private access area, it includes the following data members: the name of the burger (**std::string name**), the price of the burger (**float price**), the tax (**float tax**), the number ordered (**int number**) and the subtotal cost (**float subtotal**). It also has a function called **cost** that calculates the subtotal of a particular type of burgers.

   In the public access area, the constructer should initialize all the data members by taking **the type**, **the price**, **the number** and **the tax, as well as calculating the subtotal using the cost function**. The default arguments are "unknown burger" for name, 0.0 for price, 0.0 for tax, and 0 for number. The Burger class needs to create a set function and a get function for name, price, number, and tax. It also needs a public function called **getSubtotal()** to get the subtotal. It also has a public member function (called **show**) to show the order of a particular type of burgers (see the sample output).

   The template contains three files. You have to write a class declaration in the header file (**Burger.h**) and define the member functions for the class in the file **Burger.cpp**. You don't need to type and change the main function in **LastName_FirstName_Q3.cpp; however, you have to include Burger.h**.

The main function is as follows:

```
int main() {
     using namespace std;
     Burger special;
    special.show();

    special.setName("ECE114 Premium Burger");
    special.setPrice(8.99);
    special.setNumber(58);
    special.setTax(0.08);
    special.show();

    Burger chicken("Chicken Burger", 3.99, 0.08);
    chicken.setNumber(15);
    chicken.show();

    Burger cheeseburger("Cheeseburger", 5.49, 0.08, 45);
    cheeseburger.show();

    cout << fixed;
    cout.precision(2);
    std::cout << "Total cost (plus tax): " << special.getSubtotal()
            + chicken.getSubtotal() + cheeseburger.getSubtotal() << endl;

      return 0;
}
```

(*Continued on next page*)

Your output should be as follows:

```
Burger name: unknown burger
Order number: 0
Unit price: 0.00
Subtotal (plus tax): 0.00

Burger name: ECE114 Premium Burger
Order number: 58
Unit price: 8.99
Subtotal (plus tax): 563.13

Burger name: Chicken Burger
Order number: 15
Unit price: 3.99
Subtotal (plus tax): 64.64

Burger name: Cheeseburger
Order number: 45
Unit price: 5.49
Subtotal (plus tax): 266.81

Total cost (plus tax): 894.59
```