Homework 09

Due Date: Monday 14 April 2014 11:59 PM MST
Note: If you submit after the due date (but before the hard deadline), your submission score will be penalized by 20%.

Hard Deadline: Wednesday 16 April 2014 11:59 PM MST
Note: If you submit any time after the hard deadline, you will not receive credit.

Problem 01 (15 points)

Imagine a publishing company that markets both book and audiocassette versions of its works. A class called **publication** that stores the title (a string) and price (type float) of a publication is created. Based on the given program **Homework09P01.cpp**,

- From this class derive two classes: **book**, which adds a page count (type int); and **tape**, which adds a playing time in minutes (type float). Each of the three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display the data.
- Write a main() program that creates an array of pointers to publication. This is similar to the VIRTPERS example in the textbook on page 513. In a loop, ask the user for data about a particular book or tape, and use new to create an object of type book or tape to hold the data. Put the pointer to the object in the array. When the user has finished entering the data for all books and tapes, display the resulting data for all the books and tapes entered, using a for loop and a single statement such as

  pubarr[j]->putdata();

  to display the data from each object in the array.

- Add a member function of type bool called **isOversize()** to the **book** and **tape** classes. Let's say that a book with more than 800 pages, or a tape with a playing time longer than 90 minutes (which would require two cassettes), is considered oversize. You can access this function from **main()** and display the string "Oversize" for oversize books and tapes when you display their other data. If book and tape objects are to be accessed using pointers to them that are stored in an array of type publication, what do you need to add to the publication base class? Can you instantiate members of this base class?

Problem 02 (15 points)

Using an abstract class with only pure virtual functions, you can specify similar behaviors for possibly disparate classes. Governments and companies worldwide are becoming increasingly

concerned with carbon footprints (annual releases of carbon dioxide into the atmosphere) from buildings burning various types of fuels for heat, vehicles burning fuels for power, and the like. Many scientists blame these greenhouse gases for the phenomenon called global warming. Based on the given program **Homework09P02.cpp**:

- Create three small classes unrelated by inheritance—classes **Building**, **Car** and **Bicycle**. Give each class some unique appropriate attributes and behaviors that it does not have in common with other classes.
- Write an abstract class **CarbonFootprint** with only a pure virtual **getCarbonFootprint** method. Have each of your classes inherit from that abstract class and implement the getCarbonFootprint method to calculate an appropriate carbon footprint for that class (check out a few websites that explain how to calculate carbon footprints).
- The **main()** function in the given program creates objects of each of the three classes, places pointers to those objects in a vector of CarbonFootprint pointers. You need to iterate through the vector, polymorphically invoking each object's getCarbonFootprint method. At last, you need to release elements in the list using **delete** method.