Practice Questions

Question 1:

In ocean navigation, locations are measured in degrees and minutes of latitude and longitude. Thus if you're lying off the mouth of Papeete Harbor in Tahiti, your location is 149 degrees 34.8 minutes west longitude, and 17 degrees 31.5 minutes south latitude. This is written as 149°34.8' W, 17°31.5' S. There are 60 minutes in a degree. (An older system also divided a minute into 60 seconds, but the modern approach is to use decimal minutes instead.) Longitude is measured from 0 to 180 degrees, east or west from Greenwich, England, to the international dateline in the Pacific. Latitude is measured from 0 to 90 degrees, north or south from the equator to the poles.

Create a class angle that includes three member variables: an int for degrees, a float for minutes, and a char for the direction letter (N, S, E, or W). This class can hold either a latitude variable or a longitude variable. Write one member function to obtain an angle value (in degrees and minutes) and a direction from the user, and a second to display the angle value in 179°59.9' E format. Also write a three-argument constructor.
Write a main() program that displays an angle initialized with the constructor, and then, within a loop, allows the user to input any angle value, and then displays the value. You can use the hex character constant '\xF8', which usually prints a degree (°) symbol.

Question 2:

Start with the date structure in Exercise 3 in the previous practice questions and transform it into a date class. Its member data should consist of three ints: month, day, and year. It should also have two member functions: getdate(), which allows the user to enter a date in 12/31/02 format, and showdate(), which displays the date.

Question 3:

Create an employee class, basing it on Exercise 4 of previous practice questions. The member data should comprise an int for storing the employee number and a float for storing the employee's compensation. Member functions should allow the user to enter this data and display it. Write a main() that allows the user to enter data for three employees and display it.

Question 4:

Extend the employee class of Exercise 3 to include a date class (see Exercise 2) and an etype enum. An object of the date class should be used to hold the date of first employment; that is, the date when the employee was hired. The etype variable should hold the employee's type: laborer, secretary, manager, and so on.

These two items will be private member data in the employee definition, just like the employee number and salary. You'll need to extend the getemploy() and putemploy() functions to obtain this new information from the user and display it. These functions will probably need switch statements to handle the etype variable. Write a main() program that allows the user to enter data for three employee variables and then displays this data.

Question 5:

Write a program that converts a number entered in Roman numerals to decimal. Your program should consist of a class, say, romanType. An object of type romanType should do the following:

a. Store the number as a Roman numeral.
b. Convert and store the number into decimal form.
c. Print the number as a Roman numeral or decimal number as requested by the user.

The decimal values of the Roman numerals are:

M	1000
D	500
C	100
L	50
X	10
V	5
I	1

d. Test your program using the following Roman numerals: MCXIV, CCCLIX, MDCLXVI.

Question 6:

a. Some of the characteristics of a book are the title, author(s), publisher, ISBN, price, and year of publication. Design a class bookType that defines the book as an ADT.

i. Each object of the class bookType can hold the following information about a book: title, up to four authors, publisher, ISBN, price, and number of copies in stock. To keep track of the number of authors, add another member variable.

Include the member functions to perform the various operations on objects of type bookType. For example, the usual operations that can be performed on the title are to show the title, set the title, and check whether a title is the same as the actual title of the book. Similarly, the typical operations that can be performed on the number of copies in stock are to show the number of copies in stock, set the number of copies in stock, update the number of copies in stock, and return the number of copies in stock. Add similar operations for the publisher, ISBN, book price, and authors. Add the appropriate constructors and a destructor (if one is needed).

b. Write the definitions of the member functions of the class bookType.
c. Write a program that uses the class bookType and tests various operations on the objects of the class bookType. Declare an array of 100 components of type bookType. Some of the operations that you should perform are to search for a book by its title, search by ISBN, and update the number of copies of a book.

Question 7:

In this exercise, you will design a class memberType.

a. Each object of memberType can hold the name of a person, member ID, number of books bought, and amount spent.

b. Include the member functions to perform the various operations on the objects of memberType—for example, modify, set, and show a person's name. Similarly, update, modify, and show the number of books bought and the amount spent.

c. Add the appropriate constructors.
d. Write the definitions of the member functions of memberType.
e. Write a program to test various operations of your class memberType.

Question 8:

Using the classes designed in Programming Exercises 6 and 7, write a program to simulate a bookstore. The bookstore has two types of customers: those who are members of the bookstore and those who buy books from the bookstore only occasionally. Each member has to pay a $10 yearly membership fee and receives a 5% discount on each book purchased. For each member, the bookstore keeps track of the number of books purchased and the total amount spent. For every eleventh book that a member buys, the bookstore takes the average of the total amount of the last 10 books purchased, applies this amount as a discount, and then resets the total amount spent to 0.

Write a program that can process up to 1000 book titles and 500 members. Your program should contain a menu that gives the user different choices to effectively run the program; in other words, your program should be user driven.