CH8

1. Modify the definition of the class ***Money*** shown in Display 8.5 so that the following are added:
   a. The operators <, <=, > and >= have each been overloaded to apply to the type ***Money***. (Hint: see Self-Test exercise 8)
   b. The following member function has been added to the class definition. (We show the function declaration as it should appear in the class definition. The definition of the function itself will include the qualifier ***Money ::*** .)

      const Money percent (int percentFigure) const;

      // Returns a percentage of the money amount in the calling object. For example, if percentFigure is 10, then the value returned is 10% of the amount of money represented by the calling object.

      For example, if purse is an object of type ***Money*** whose value represents the amount $100.10, then the call

      purse. percent (10);

      returns 10% of $100.10; that is, it returns a value of type ***Money*** that represents the amount $10.01.

※此題須另外加上+ - ==的 operators (詳細請看 sample)

4. Cumulatively modify the example from Display 8.7 as follows.
   a. In Display 8.7, replace the private ***char*** members ***first*** and ***second*** with an array of ***char*** size 100 and a private data member named ***size***.

      Provide a default constructor that initializes size to 10 and sets the first 10 of the ***char*** positions to '#'. (This only uses 10 of the possible 100 slots.)

      Provide an accessor function that returns the value of the private member ***size***.

      Test.

   b. Add an ***operator [ ]*** member that returns a ***char&*** that allows the user to access or to set any member of the private data array using a non-negative index that is less than ***size***.

      Test.

   c. Add a constructor that takes an ***int*** argument, sz, that sets the first sz members of the char array to '#'.

      Test.

d. Add a constructor that takes an ***int*** argument, sz, and an array of ***char*** of size sz. The constructor should set the first sz members of the private data array to the sz members of the argument array of char.

   Test.

NOTES: When you test, you should test with known good values, with data on boundaries and with deliberately bad value. You are not required for index out of bounds errors in your code, but that would be a nice touch. Error handling alternatives: Issue an error message then die (that is, call ***exit (1)***) or give the user another chance to make a correct entry.

5. Write the definition for a class named ***Vector2D*** that stores information about a two-dimensional vector. The class should have function to get and set the *x* and *y* component, where *x* and *y* are integers.

   Next, overload the * operator so that it returns the dot product of two vectors. The dot product of two-dimensional vectors **A** and **B** is equal to

$$(A_x \times B_x) + (A_y \times B_y)$$

   Finally, write a main subroutine that tests the * operation.

8. Do Programming Project 6.10, the definition of a ***Temperature*** class, except overload = =, << and >> as member operators. The = = operator should return true if the two temperature values are identical, while << should output the temperature in Fahrenheit and >> should input the temperature in Fahrenheit. Create appropriate tests for the overloaded operators. (此題請以 kelvin 氏溫標轉為華氏和攝氏溫標，詳細請看 sample)

9. Programming Project 6.12 asked you to write a ***BoxOfProduce*** class that stored three bundles of fruits or vegetables (stored in an array of string of size 3) to ship to a customer. Rewrite this class to use a vector instead of an array and add appropriate constructors, mutators, and accessors. The class should have a function to add additional fruits or vegetables to the box so there could be more than three bundles in one box. The ***output*** function should output all items in the box. Overload the + operator to return a new ***BoxOfProduce*** object that combines the vector contents of both operand BoxOfProduce objects. Test your functions and + operator from the main function. You don't have to implement the rest of Programming Project 6.12 for this Programming Project, only the changes to the ***BoxOfProduce*** class.

CH9

2. An intelligence agency sends secure information to recipients in an encrypted form. The recipient then decrypts the message using a decryption algorithm , the rules of which are as follows.
   a. Every alphabet should be replaced with the third letter after it in the alphabet.
   b. Every number should be added with 5, and the remainder of the sum after division by 10 should be used to get a single digit.
   c. Every space and newline character should be omitted.
   d. Every special character should be replaced by a space.

   Write a program that reads encrypted text and displays the decrypted message by applying the algorithm on it.

4. Write a program that reads in a line of text and replaces all four-letter words with the word " love ". For example, the input string

   I hate you, you dodo!

   should produce the following output

   I love you, you love!

   Of course, the output Will not always make sense .For example, the input string

   John will run home.

   should produce the following output:

   Love love run love.

   If the four-letter word starts with a capital letter, it should be replaced by "Love‘", not by "love". You need not check capitalization, except for the first letter of a word. A word is any string consisting of the letters of the alphabet and delimited at each end by a blank, the end of the line, or any other character that is not a letter. Your program should repeat this action until the user say to quit.

6. There is a CD available for purchase that contains .jpeg and .gif images of music that Is in the public domain. The CD includes a file consisting of lines containing the names , then composers of that title, one per line. The name of the piece is first , then zero or more space then a dash (-) character ,then one more spaces ,then the composer’s name. The composer name may be only the last name , an initial and one name, two names (first and last), or three name (first ,middle and Last). There are a few tunes with “no author listed” as author. In the subsequent processing, ”no author listed” should not be rearranged. Here is a very abbreviated list of the titles and authors.

   1. Adagio “MoonLight” Sonata - Ludwig Van Beethoven

2. An Alexis ~ FH. Hummel and].N. Hummel
3. A La Bien Aimee - Ben Schutt
4. At Sunset - E. MacDowell
5. Angelus - J. Massenet
6. Anitra's Dance - Edward Grieg
7. Ase's Death - Edward Grieg
8. Au Matin- Benj. – Godard
…
37. The Dying Poet - L. Gottschalk
38. Dead March - G.F. Handel
39. Do They Think of Me At Home - Chas. W. Glover
40. The Dearest Spot ~ WT. Wrighton
1. Evening ~ L. Van Beethoven
2. Embarrassment: - Franz Abt
3. Erin is my Home - no author listed
4. Ellen Bayne - Stephen C. Foster
…
9. Alla Mazurka -A. Nemerowsky
….
1.The Dying Volunteer - A.E. Muse
2. Dolly Day - Stephen C. Foster
3. Dolcy Jones - Stephen C. Foster
4. Dickory, Dickory, Dock - no author listed
5. The Dear Little Shamrock - no author listed
6. Dutch Warbler - no author listed
…

The ultimate task is to produce an alphabetized list of composers followed by a list of pieces by them alphabetized on the title within composer. This exercise is easier if it is broken into pieces:

Write code to do the following:

a. Remove the lead numbers, any periods, and any spaces so that the first word of the title is the first word of the line.

b. Replace any multiple spaces with a single space.

c. A few titles may have several - characters, for example.

20. Ba- Be- Bi- Bo- Bu - no author listed

Replace all dash - characters on any line before the end of the line by a space except the last one.

d. The last word in the title may have the - character with no space between it and the= character. Put the space in.

e. When alphabetizing the title, you do not want to consider an initial “A”, “An”, or “The” in the title. Write code to move such initial words to just before the -character. A comma after the last word in the title is not required, but that would be a nice touch. ”This can be done after the composers’ names are moved to the front, but obviously the code will be different.

f. Move the composers’ names to the beginning of the line, followed by the character, followed by the composition title.

g. Move any first initial, or first and second names of the composer to after the composer’s last name. If the composer is “no author listed” this should not be rearranged, so test for this combination.

h. Alphabetize by composer using any sort routine you know. You may ignore any duplicate composer’s last name, such as CPE Bach and JS Bach. but sorting by composer’s second name would be a nice touch. You may use the insertion sort, or selection sort, or bubble sort, or other sorting algorithm.

i. If you have not already done so, move “A”, “An”, or “The“ that may begin a title to the end of the title. Then alphabetize within each composer by composition title.

j. Keep a copy of your design and your code. You will be asked to do this over using the STL vector container.

10. Write a simple trivia quiz game. Start by creating a Trivia class that contains information about a single trivia question. The class should contain a String for the question, a string for the answer to the question, and an integer representing the dollar amount the question is worth (harder questions should be worth more). Add appropriate constructor and accessor functions. In your main function create either an array or a vector of type Trivia and hard-code at least five trivia questions of your choice. Your program should then ask each question to the player, input the player’s answer, and check if the player‘s answer matches the actual answer. If so, award the player the dollar amount for that question. If the player enters the wrong answer your program should display the correct answer. When all questions have been asked display the total amount that the player has won.