

1. Modify the definition of class Money from Display 8.5 by

a) adding overloaded operators <, <=, > and >= for Money objects (Hint: Self-Test exercise 8)

b) adding a *member* function with this declaration (to be placed in the class definition and implemented)

```
const Money percent(int percentFigure) const;
//Returns percentFigure percent of Money.
//example: If percentFigure is 10, a Money object is returned that
//represents 0.1 of the amount of money of the calling object
//If purse is a Money object representing $100.20, purse.percent(10)
//is a Money object representing $10.02.
```

2. 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 of 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.

3. Programming Hw3 asked you to write a `BoxOfProduce` class that stored three bundles of fruits or vegetables (stored in an array 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.

4. 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 5 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.

5. 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 that piece is first, then zero or more spaces then a dash (-) character, then one or 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 names (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 – F.H. Hummel and J.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 Marc – G.F. Handel
39. Do They Think of Me At Home Chas. W. Glover
40. The Dearest Spot – W.T. 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. Te 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 composer's names are moved to the front, but obviously the code will be different.

- f. Move the composer's 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.