Exercises

18.4 Fill in the blanks in each of the following:

a. Class string member functions ___________ and ___________ convert strings to Cstyle strings.
b. Class string member function ___________ is used for assignment.
c. ___________ is the return type of function rbegin.
d. Class string member function ___________ is used to retrieve a substring.

18.5 State which of the following statements are true and which are false. If a statement is false, explain why.

a. strings are always null terminated.
b. Class string member function max_size returns the maximum size for a string.
c. Class string member function at can throw an out_of_range exception.
d. Class string member function begin returns an iterator.

18.6 Find any errors in the following and explain how to correct them:

a. std::cout << s.data() << std::endl; // s is "hello"
b. erase( s.rfind( "x" ), 1 ); // s is "xenon"
c. string& foo() 
d. { 
e. string s( "Hello" );
f. ... // other statements 
g. return;
h. } // end function foo
i.

18.7 (Simple Encryption) Some information on the Internet may be encrypted with a simple algorithm known as "rot13," which rotates each character by 13 positions in the alphabet. Thus, ‘a’ corresponds to ‘n’, and ‘x’ corresponds to ‘k’. rot13 is an example of symmetric key encryption. With symmetric key encryption, both the encrypter and decrypter use the same key.

a. Write a program that encrypts a message using rot13.
b. Write a program that decrypts the scrambled message using 13 as the key.

c. After writing the programs of part (a) and part (b), briefly answer the following question: If you did not know the key for part (b), how difficult do you think it would be to break the code? What if you had access to substantial computing power (e.g., supercomputers)? In Exercise 18.26 we ask you to write a program to accomplish this.

18.8 Write a program using iterators that demonstrates the use of functions `rbegin` and `rend`.

18.9 Write a program that reads in several strings and prints only those ending in "r" or "ay". Only lowercase letters should be considered.

18.10 Write a program that demonstrates passing a string both by reference and by value.

18.11 Write a program that separately inputs a first name and a last name and concatenates the two into a new string.

18.12 Write a program that plays the game of Hangman. The program should pick a word (which is either coded directly into the program or read from a text file) and display the following:

```
Guess the word: X X X X X X
```

Each X represents a letter. The user tries to guess the letters in the word. The appropriate response yes or no should be displayed after each guess. After each incorrect guess, display the diagram with another body part filled. After seven incorrect guesses, the user should be hanged. The display should look as follows:

```
O
/|\ 
/ \
```

After each guess, display all user guesses. If the user guesses the word correctly, the program should display

```
Congratulations!!! You guessed my word. Play again? yes/no
```

18.13 Write a program that inputs a string and prints the string backward. Convert
all uppercase characters to lowercase and all lowercase characters to uppercase.

18.14 Write a program that uses the comparison capabilities introduced in this chapter to alphabetize a series of animal names. Only uppercase letters should be used for the comparisons.

18.15 Write a program that creates a cryptogram out of a string. A cryptogram is a message or word in which each letter is replaced with another letter. For example the string

    The bird was named squawk

might be scrambled to form

    cin vrjs otz ethns zxqtop

Note that spaces are not scrambled. In this particular case, 'T' was replaced with 'x', each 'a' was replaced with 'h', etc. Uppercase letters become lowercase letters in the cryptogram. Use techniques similar to those in Exercise 18.7.

18.16 Modify Exercise 18.15 to allow the user to solve the cryptogram. The user should input two characters at a time: The first character specifies a letter in the cryptogram, and the second letter specifies the replacement letter. If the replacement letter is correct, replace the letter in the cryptogram with the replacement letter in uppercase.

18.17 Write a program that inputs a sentence and counts the number of palindromes in it. A palindrome is a word that reads the same backward and forward. For example, "Tree" is not a palindrome, but "noon" is.

18.18 Write a program that counts the total number of vowels in a sentence. Output the frequency of each vowel.

18.19 Write a program that inserts the characters "*****" in the exact middle of a string.

18.20 Write a program that erases the sequences "by" and "BY" from a string.

18.21 Write a program that inputs a line of text, replaces all punctuation marks with spaces and uses the C-string library function `strtok` to tokenize the string into individual words.
18.22 Write a program that inputs a line of text and prints the text backwards. Use iterators in your solution.

18.23 Write a recursive version of Exercise 18.22.

18.24 Write a program that demonstrates the use of the erase functions that take iterator arguments.

18.25 Write a program that generates the following from the string "abcdefghijklmnopqrstuvwxyz{":

```
  a
  bcb
cdedc
defgfed
efghihgf
  fghijkjihgf
gijklmlkjihih
  ghijklmnonlkj
  ijklnopqponmlkji
  jklmnopqrsrpomnlk
  klmnopqrstsrqponmlk
  lmnopqrstuvwvutsrqponml
  mnpqrstuvwxyz{zyxwvutsrqpon
```

18.26 In Exercise 18.7, we asked you to write a simple encryption algorithm. Write a program that will attempt to decrypt a "rot13" message using simple frequency substitution. (Assume that you do not know the key.) The most frequent letters in the encrypted phrase should be replaced with the most commonly used English letters (a, e, i, o, u, s, t, r, etc.). Write the possibilities to a file. What made the code breaking easy? How can the encryption mechanism be improved?

18.27 Write a version of the selection sort routine (Fig. 8.28) that sorts strings. Use function swap in your solution.

18.28 Modify class Employee in Figs. 13.6.13.7 by adding a private utility function called isValidSocialSecurityNumber. This member function should validate the format of a social security number (e.g., ###-##-####, where # is a digit). If the format is valid, return true; otherwise return false.