Programming Exercises

1. Write a short program that asks for your height in integer inches and then converts your height to feet and inches. Have the program use the underscore character to indicate where to type the response. Also use a `const` symbolic constant to represent the conversion factor.

2. Write a short program that asks for your height in feet and inches and your weight in pounds. (Use three variables to store the information.) Have the program report your body mass index (BMI). To calculate the BMI, first convert your height in feet and inches to your height in inches (1 foot = 12 inches). Then convert your height in inches to your height in meters by multiplying by 0.0254. Then convert your weight in pounds into your mass in kilograms by dividing by 2.2. Finally, compute your BMI by dividing your mass in kilograms by the square of your height in meters. Use symbolic constants to represent the various conversion factors.

3. Write a program that asks the user to enter a latitude in degrees, minutes, and seconds and that then displays the latitude in decimal format. There are 60 seconds of arc to a minute and 60 minutes of arc to a degree; represent these values with symbolic constants. You should use a separate variable for each input value. A sample run should look like this:

```
Enter a latitude in degrees, minutes, and seconds:
First, enter the degrees: 37
Next, enter the minutes of arc: 51
Finally, enter the seconds of arc: 19
37 degrees, 51 minutes, 19 seconds = 37.8553 degrees
```

4. Write a program that asks the user to enter the number of seconds as an integer value (use type `long`, or, if available, `long long`) and that then displays the equivalent time in days, hours, minutes, and seconds. Use symbolic constants to represent the number

of hours in the day, the number of minutes in an hour, and the number of seconds in a minute. The output should look like this:

```
Enter the number of seconds: 31600000
31600000 seconds = 365 days, 17 hours, 46 minutes, 40 seconds
```

5. Write a program that requests the user to enter the current world population and the current population of the U.S. (or of some other nation of your choice). Store the information in variables of type `long long`. Have the program display the percent that the U.S. (or other nation's) population is of the world's population. The output should look something like this:

   ```
   Enter the world's population: 6898758899
   Enter the population of the US: 310783781
   The population of the US is 4.50492% of the world population.
   ```

   You can use the Internet to get more recent figures.

6. Write a program that asks how many miles you have driven and how many gallons of gasoline you have used and then reports the miles per gallon your car has gotten. Or, if you prefer, the program can request distance in kilometers and petrol in liters and then report the result European style, in liters per 100 kilometers.

7. Write a program that asks you to enter an automobile gasoline consumption figure in the European style (liters per 100 kilometers) and converts to the U.S. style of miles per gallon. Note that in addition to using different units of measurement, the U.S. approach (distance / fuel) is the inverse of the European approach (fuel / distance). Note that 100 kilometers is 62.14 miles, and 1 gallon is 3.875 liters. Thus, 19 mpg is about 12.4 l/100 km, and 27 mpg is about 8.7 l/100 km.

8. Write a C++ program that requests and displays information as shown in the following example of output:

   ```
   What is your first name? Betty Sue
   What is your last name? Yewe
   What letter grade do you deserve? B
   What is your age? 22
   Name: Yewe, Betty Sue
   Grade: C
   Age: 22
   ```

   Note that the program should be able to accept first names that comprise more than one word. Also note that the program adjusts the grade downward—that is, up one letter. Assume that the user requests an A, a B, or a C so that you don't have to worry about the gap between a D and an F.

9. Rewrite Listing 2.18, using the C++ `string` class instead of `char` arrays.

10. Write a program that asks the user to enter his or her first name and then last name, and that then constructs, stores, and displays a third string, consisting of the user's last

name followed by a comma, a space, and first name. Use char arrays and functions from the cstring header file. A sample run could look like this:

```
Enter your first name: Flip
Enter your last name: Fleming
Here's the information in a single string: Fleming, Flip
```

11. Write a program that asks the user to enter his or her first name and then last name, and that then constructs, stores, and displays a third string consisting of the user's last name followed by a comma, a space, and first name. Use string objects and methods from the string header file. A sample run could look like this:

```
Enter.your first name: Flip
Enter your last name: Fleming
Here's the information in a single string: Fleming, Flip
```

12. The CandyBar structure contains three members. The first member holds the brand name of a candy bar. The second member holds the weight (which may have a fractional part) of the candy bar, and the third member holds the number of calories (an integer value) in the candy bar. Write a program that declares such a structure and creates a CandyBar variable called snack, initializing its members to "Mocha Munch", 2.3, and 350, respectively. The initialization should be part of the declaration for snack. Finally, the program should display the contents of the snack variable.

13. The CandyBar structure contains three members, as described in Programming Exercise 5. Write a program that creates an array of three CandyBar structures, initializes them to values of your choice, and then displays the contents of each structure.

14. William Wingate runs a pizza-analysis service. For each pizza, he needs to record the following information:

    - The name of the pizza company, which can consist of more than one word
    - The diameter of the pizza
    - The weight of the pizza

    Devise a structure that can hold this information and write a program that uses a structure variable of that type. The program should ask the user to enter each of the preceding items of information, and then the program should display that information. Use cin (or its methods) and cout.

15. Do Programming Exercise 7 but use new to allocate a structure instead of declaring a structure variable. Also have the program request the pizza diameter before it requests the pizza company name.

16. Do Programming Exercise 6, but instead of declaring an array of three CandyBar structures, use new to allocate the array dynamically.

17. Write a program that requests the user to enter three times for the 40-yd dash (or 40-meter, if you prefer) and then displays the times and the average. Use an array object to hold the data. (Use a built-in array if array is not available.)