**Chapter Review**

1. What are the modules of C++ programs called?

Functions.

2. What does the following preprocessor directive do?

***#include <iostream>***

This directive causes the preprocessor to add the contents of the iostream file to your program.

3. What does the following statement do?

***using namespace std;***

To make the definitions in iostream available to your program.

4. What statement would you use to print the phrase “Hello, world” and then start a new line?

***cout << “Hello, world” << endl;***

5. What statement would you use to create an integer variable with the name ***cheeses***?

***int cheeses;***

6. What statement would you use to assign the value 32 to the variable ***cheeses***?

***cheeses = 32;***

7. What statement would you use to read a value from keyboard input into the variable ***cheeses***?

***cin >> cheeses;***

8. What statement would you use to print “We have X varieties of cheese,” where the current value of the ***cheeses*** variable replaces ***X***?

***cout << “We have ” << cheeses << “ varieties of cheese,”;***

9. What do the following function prototypes tell you about the functions?

***int froop(double t);*** froop() function with double-type argument and return integer value

***void rattle(int n);*** rattle() function with integer-type argument and return no value.

***int prune(void);*** prune() function with no arguments and return integer value

10. When do you not have to use the keyword ***return*** when you define a function?

When the function prototype is void.

11. Suppose your ***main()*** function has the following line:

***cout << “Please enter your PIN: “;***

And suppose the compiler complains that ***cout*** is an unknown identifier. What is the likely cause of this complaint, and what are three ways to fix the problem?

Does not include ***#include<iostream>*** / ***using namespace std;***

* Place ***using namespace std;*** above the function definitions in a file.
* Place ***using namespace std;*** in a specific function definition.
* Use ***std::*** prefix whenever you use elements from the std namespace.

**Programming Exercises**

1. Write a C++ program that displays your name and address (or if you value your privacy, a fictitious name and address).

#include<iostream>

int main()

{

    using namespace std;

    cout << "Name: GAO QI QIANG" << endl << "Address: Jing Hai" << endl;

    return 0;

}

2. Write a C++ program that asks for a distance in furlongs and converts it to yards. (One furlong is 220 yards.)

#include<iostream>

int yards2furlongs(int furlongs);

int main()

{

    using namespace std;

    int furlongs;

    cout << "Please enter the distance from GBS to MMU in furlongs: ";

    cin >> furlongs;

    int yards = yards2furlongs(furlongs);

    cout << furlongs << " furlongs is equivalent of " << yards << " yards."<< endl;

    return 0;

}

int yards2furlongs(int f)

{

    return 220 \* f;

}

3. Write a C++ program that uses three user-defined functions (counting ***main()*** as one) and produces the following output:

***Three blind mice***

***Three blind mice***

***See how they run***

***See how they run***

One function, called two times, should produce the first two lines, and the remaining function, also called twice, should produce the remaining output.

#include<iostream>

using namespace std;

void blind();

void run();

int main()

{

    blind();

    blind();

    run();

    run();

    return 0;

}

void blind()

{

    cout << "Three blind mice" << endl;

}

void run()

{

    cout << "See how they run" << endl;

}

4. Write a program that asks the user to enter his or her age. The program then should display the age in months:

***Enter your age: 29***

***Your age in months is 384.***

#include<iostream>

using namespace std;

int main()

{

    cout << "Enter your age: ";

    int age;

    cin >> age;

    int month = age \* 12;

    cout << "Your age in months is " << month << "." << endl;

    return 0;

}

5. Write a program that has ***main()*** call a user-defined function that takes a Celsius temperature value as an argument and then returns the equivalent Fahrenheit value.

The program should request the Celsius value as input from the user and display the result, as shown in the following code:

***Please enter a Celsius value: 20***

***20 degrees Celsius is 68 degrees Fahrenheit.***

For reference, here is the formula for making the conversion:

Fahrenheit = 1.8 × degrees Celsius + 32.0

#include<iostream>

using namespace std;

double c2f(double c);

int main()

{

    cout << "Please enter a Celsius value: ";

    double celsius;

    cin >> celsius;

    double fahrenheit = c2f(celsius);

    cout << celsius << " degrees Celsius is " << fahrenheit << " degrees Fahrenheit."<< endl;

    return 0;

}

double c2f(double c)

{

    return 1.8 \* c + 32.0;

}

6. Write a program that has ***main()*** call a user-defined function that takes a distance in light years as an argument and then returns the distance in astronomical units.

The program should request the light year value as input from the user and display the result, as shown in the following code:

***Enter the number of light years: 4.2***

***4.2 light years = 265608 astronomical units.***

An astronomical unit is the average distance from the earth to the sun (about 150,000,000 km or 93,000,000 miles), and a light year is the distance light travels in a year (about 10 trillion kilometers or 6 trillion miles). (The nearest star after the sun is about 4.2 light years away.) Use type double (as in Listing 2.4) and this conversion factor:

1 light year = 63,240 astronomical units

#include<iostream>

using namespace std;

double l2a(double l);

int main()

{

    cout << "Enter the number of light years: ";

    double lightyears;

    cin >> lightyears;

    double astronomicalunits = l2a(lightyears);

    cout << lightyears << " light years = " << astronomicalunits << " astronomical units." << endl;

    return 0;

}

double l2a(double l)

{

    return l \* 63240;

}

7. Write a program that asks the user to enter an hour value and a minute value. The ***main()*** function should then pass these two values to a type ***void*** function that displays the two values in the format shown in the following sample run:

***Enter the number of hours: 9***

***Enter the number of minutes: 28***

***Time: 9:28***

#include<iostream>

using namespace std;

void display(int hour, int minute);

int main()

{

    cout << "Enter the number of hours: ";

    int hours;

    cin >> hours;

    cout << "Enter the number of minutes: ";

    int minutes;

    cin >> minutes;

    display(hours, minutes);

    return 0;

}

void display(int hour, int minute)

{

    cout << "Time: " << hour << ":" << minute << endl;

}