Answers to review questions from Chapter 1

1. When you write a C program, do you prepare a source file or an object file?

   A source file

2. What characters are used to mark comments in a C++ program?

   `/* ... */`

3. In an **`#include`** line, the name of the library header file can be enclosed in either angle brackets or double quotation marks. What is the difference between the two forms of punctuation?

   The angle brackets indicate a library inclusion; quotations are used for user-defined headers.

4. How would you define a constant called **`CENTIMETERS_PER_INCH`** with the value 2.54?

   ```
   const double CENTIMETERS_PER_INCH = 2.54;
   ```

5. What is the name of the function that must be defined in every C++ program? What statement typically appears at the end of that function?

   All C++ programs contain a function called `main`, which typically ends with the statement

   ```
   return 0;
   ```

6. What is the purpose of **`endl`** when you are writing to the **`cout`** output stream?

   The `endl` manipulator signifies the end of the output line and causes the next values written to `cout` to appear on the next line.

7. Define each of the following terms for variables: *name, type, value,* and *scope.*

   The *name* is the identifier you use in the program to refer to that variable. The *type* refers to the type of values that variable may contain. The *value* is the current contents of the variable, which will be of the specified type. The *scope* is the region of the program in which that variable is defined.

8. Indicate which of the following are legal variable names in C++:

a.	**`x`**	**Legal**
b.	**`formula1`**	**Legal**
c.	**`average_rainfall`**	**Legal**
d.	**`%correct`**	**Illegal (begins with `%`)**
e.	**`short`**	**Illegal (reserved word)**
f.	**`tiny`**	**Legal**
g.	**`total output`**	**Illegal (contains a space)**
h.	**`aVeryLongVariableName`**	**Legal**
i.	**`12MonthTotal`**	**Illegal (begins with a digit)**
j.	**`marginal-cost`**	**Illegal (contains a hyphen)**
k.	**`b4hand`**	**Legal (but perhaps too clever)**
l.	**`_stk_depth`**	**Legal**

9. What are the two attributes that define a data type?

   The *domain* is the set of values that are legal for that type; the *set of operations* defines the behavior of that type.

10. How do the types `short`, `int`, and `long` differ?

    The types `short`, `int`, and `long` represent integers of different preferred sizes. Typical values for these types are 16, 32, and 64 bits, but C++ does not guarantee a particular size. The only requirement is that the sizes of these types cannot decrease as you move through the list. Thus, a `long` must not contain fewer bits that an `int`, although the two types can be the same size.

11. What does ASCII stand for?

    American Standard Code for Information Interchange

12. List all possible values of type `bool`.

    `true` and `false`

13. What statements would you include in a program to read a value from the user and store it in the variable `x`, which is declared as a `double`?

```
double x;
cin >> x;
```

14. Suppose that a function contains the following declarations:

```
int i;
double d;
char c;
string s;
```

    Write a statement that displays the values of each of these variables on the screen along with the name of the variable, so you can tell the values apart.

```
cout << "i = " << i << "; d = " << d <<
       "c = " << c << "; s = " << s;
```

15. Indicate the values and types of the following expressions:

    a. `2 + 3` **5, `int`**
    b. `19 / 5` **3, `int`**
    c. `19.0 / 5` **3.8, `double`**
    d. `3 * 6.0` **18.0, `int`**
    e. `19 % 5` **4, `int`**
    f. `2 % 7` **2, `int`**

16. What is the difference between the unary minus and the subtraction operator?

    The unary minus operator is written before a single operand and denotes negation, as in `-x`. The subtraction operator is written between two operands as in `x - 2`.

17. What does the term *truncation* mean?

    ***Truncation* converts a floating-point number to an integer by throwing away any digits after the decimal point. Thus, the floating-point number 3.9999 truncates to the integer 3.**

18. What is a *type cast* and how do you indicate one in C++?

    A *type cast* is used to indicate conversion of one type to another. In C++, type casts are written by using the type name as a function, as in the expression `int(x)`, which truncates `x` to an integer.

19. Calculate the result of each of the following expressions:

    a. `6 + 5 / 4 - 3` **4**
    b. `2 + 2 * (2 * 2 - 2) % 2 / 2` **2**
    c. `10 + 9 * ((8 + 7) % 6) + 5 * 4 % 3 * 2 + 1` **42**
    d. `1 + 2 + (3 + 4) * ((5 * 6 % 7 * 8) - 9) - 10` **42**

20. How do you specify a shorthand assignment operation?

    Shorthand assignment operators are written by writing the operator in front of the equal sign used for assignment, as in

    ```
    balance += deposit;
    ```

21. What is the difference between the expressions `++x` and `x++`?

    Both expressions *increment* (add one to) the value of `x`. The difference is the value of the expression. The expression `++x` returns the incremented value; the expression `x++` returns the original value before the increment.

22. What is meant by *short-circuit evaluation?*

    ***Short-circuit evaluation* evaluates only as much of an expression as is necessary to determine the result. C++ uses short-circuit evaluation to evaluate the logical operators `&&` and `||`. With `&&`, for example, there is no need to evaluate the right operand if the left operand is `true`.**

23. Write out the general syntactic form for each of the following control statements: `if`, `switch`, `while`, and `for`.

    `if (`*condition*`)` *statement*

    `if (`*condition*`)` *statement* `else` *statement*

    ```
    switch (e) {
     case c1:
       statements
       break;
     case c2:
       statements
       break;
     . . . more case clauses . . .
     default:
       statements
       break;
    }
    ```

    ```
    while (conditional-expression) {
       statements
    }
    ```

    ```
    for (init; test; step) {
       statements
    }
    ```

24. Describe in English the operation of the `switch` statement, including the role of the `break` statement at the end of each `case` clause.

When the program executes a `switch` statement, it evaluates the control expression and compares it against the values c_1, c_2, and so forth, each of which must be a constant. If one of the constants matches the value of the control expression, the statements in the associated `case` clause are executed. When the program reaches the `break` statement at the end of the clause, the operations specified by that clause are complete, and the program continues with the statement that follows the entire `switch` statement. If none of the expression match the value of the control expression, the program executes the statements in the `default` clause, if any.

25. What is a *sentinel?*

A *sentinel* is a value used to indicate some kind of special processing and often signals the end of a list of input values.

26. What `for` loop control line would you use in each of the following situations?

    a. Counting from 1 to 100

    ```
    for (int i = 1; i <= 100; i++)
    ```

    b. Counting by sevens starting at 0 until the number has more than two digits

    ```
    for (int i = 0; i < 100; i += 7)
    ```

    c. Counting backward by twos from 100 to 0

    ```
    for (int i = 100; i >= 0; i -= 2)
    ```