Tutorial Programs

1. Write a program that requests the user to enter two integers.The program should then calculate and report the sum of all the integers between and including the two integers. At this point,assume that the smaller integer is entered first. For example, if the user enters 2 and 9, the program should report that the sum of all the integers from 2 through 9 is 44.
2. Write a program that opens a text file, reads it character-by-character to the end of the file,and reports the number of characters in the file.
3. Write a program that reads up to 10 donation values into an array of double. (Or, if you prefer, use an array template object.) The program should terminate input on non-numeric input. It should report the average of the numbers and also report how many numbers in the array are larger than the average.
4. Here is a structure declaration:

```
struct box
{
        char maker[40];
        float height;
        float width;
        float length;
        float volume;
};
```

    a. Write a function that passes a box structure by value and that displays the value of each member.
    b. Write a function that passes the address of a box structure and that sets the volume member to the product of the other three dimensions.
    c. Write a simple program that uses these two functions.
    d. Write a function that has a reference to a box structure as its formal argument and displays the value of each member (Separate program point d & e).
    e. Write a function that has a reference to a box structure as its formal argument and sets the volume member to the product of the other three dimensions.
5. Write a program that uses the following functions:
   Fill_array() takes as arguments the name of an array of double values and an array size. It prompts the user to enter double values to be entered in the array. It ceases taking input when the array is full or when the user enters non-numeric input,and it returns the actual number of entries.
   Show_array() takes as arguments the name of an array of double values and an array size and displays the contents of the array.
   Reverse_array() takes as arguments the name of an array of double values and an array size and reverses the order of the values stored in the array.

The program should use these functions to fill an array, show the array, reverse the array, show the array, reverse all but the first and last elements of the array,and then show the array

6. Suppose the song() function has this prototype:
   void song(const char * name, int times);
   a. How would you modify the prototype so that the default value for times is 1?
   b. What changes would you make in the function definition?
   c. Can you provide a default value of "O, My Papa" for the name?
7. Which of the following initializations are legal? Explain why.
   (a) int i = -1, &r = 0; (b) int *const p2 = &i2;
   (c) const int i = -1, &r = 0; (d) const int *const p3 = &i2;
   (e) const int *p1 = &i2; (f) const int &const r2;
   (g) const int i2 = i, &r = i;
8. Explain the following definitions. Identify any that are illegal.
   (a) int i, *const cp; (b) int *p1, *const p2;
   (c) const int ic, &r = ic; (d) const int *const p3;
   (e) const int *p;
9. Using the variables in the previous exercise, which of the following assignments are legal? Explain why.
   (a) i = ic; (b) p1 = p3;
   (c) p1 = ⁣ (d) p3 = ⁣
   (e) p2 = p1; (f) ic = *p3;
10. Assuming txt_size is a function that takes no arguments and returns an int value, which of the following definitions are illegal? Explain why.
    unsigned buf_size = 1024;
    (a) int ia[buf_size]; (b) int ia[4 * 7 - 14];
    (c) int ia[txt_size()]; (d) char st[11] = "fundamental";
11. Correct the errors in each of the following code fragments:
    (a) if (ival1 != ival2)
            ival1 = ival2
        else ival1 = ival2 = 0;
    (b) if (ival < minval)
            minval = ival;
            occurs = 1;
    (c) if (int ival = get_value())
            cout << "ival = " << ival << endl;
        if (!ival)
            cout << "ival = 0\n";
    (d) if (ival = 0)
        ival = get_value();
12. Explain each of the following loops. Correct any problems you detect.
    (a) do
            int v1, v2;
            cout << "Please enter two numbers to sum:" ;

```
        if (cin >> v1 >> v2)
        cout << "Sum is: " << v1 + v2 << endl;
  while (cin);
```

(b)
```
do {
        // ...
  } while (int ival = get_response());
```

(c)
```
do {
        int ival = get_response();
  } while (ival);
```

13. Indicate which of the following functions are in error and why. Suggest how you might correct the problems.

(a)
```
int f() {
        string s;
        // ...
        return s;
  }
```

(b)
```
f2(int i) { /* ... */ }
```

(c)
```
int calc(int v1, int v1) /* ... */ }
```

(d)
```
double square(double x) return x * x;
```

14. Define a pair of classes X and Y, in which X has a pointer to Y, and Y has an object of type X.
15. Which, if any, of the following statements are untrue? Why?

    (a) A class must provide at least one constructor.

    (b) A default constructor is a constructor with an empty parameter list.

    (c) If there are no meaningful default values for a class, the class should not provide a default constructor.

    (d) If a class does not define a default constructor, the compiler generates one that initializes each data member to the default value of its associated type.
16. 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.
17. Write a program that uses an array of char and a loop to read one word at a time until the word done is entered.The program should then report the number of words entered (not counting done). A sample run could look like this:

    Enter words (to stop, type the word done):

    anteater birthday category dumpster

    envy finagle geometry done for sure

    You entered a total of 7 words.

    You should include the cstring header file and use the strcmp() function to make the comparison test.

18. Carefully consider the following program:

```
#include <iostream>
using namespace std;
int main()
{
        char ch;
        int ct1, ct2;
        ct1 = ct2 = 0;
        while ((ch = cin.get()) != '$')
        {
                cout << ch;
                Ct1++;
                if (ch = '$')
                        Ct2++;
                cout << ch;
        }
        cout <<"ct1 = " << ct1 << ", ct2 = " << ct2 << "\n";
        return 0;
}
```

Suppose you provide the following input, pressing the Enter key at the end of each line:

Hi!

Send $10 or $20 now!

What is the output?

19. Consider the following skeletal C program:

```
void fun1(void); /* prototype */
void fun2(void); /* prototype */
void fun3(void); /* prototype */
void main() {
  int a, b, c;
  . . . }
void fun1(void) {
 int b, c, d;
 . . . }
void fun2(void) {
 int c, d, e;
 . . . }
void fun3(void) {
 int d, e, f;
 . . . }
```

Given the following calling sequences and assuming that dynamic scoping is used, what variables are visible during execution of the last function called? Include with each visible variable the name of the function in which it was defined.

a. main calls fun1; fun1 calls fun2; fun2 calls fun3.
b. main calls fun1; fun1 calls fun3.
c. main calls fun2; fun2 calls fun3; fun3 calls fun1.
d. main calls fun3; fun3 calls fun1.
e. main calls fun1; fun1 calls fun3; fun3 calls fun2.
f. main calls fun3; fun3 calls fun2; fun2 calls fun1.

20. Assume the following JavaScript program was interpreted using static-scoping rules. What value of x is displayed in function sub1? Under dynamic-scoping rules, what value of x is displayed in function sub1?

```
var x;
function sub1()
{
        document.write("x = " + x + "<br />");
}
function sub2()
{
        var x;
        x = 10;
        sub1();
}
x = 5;
sub2();
```

21. Given the following classes, explain each print function:

```
class base {
public:
        string name() { return basename; }
        virtual void print(ostream &os) { os << basename; }
private:
        string basename;
};
class derived : public base {
public:
        void print(ostream &os) { print(os); os << " " << i; }
private:
        int i;
};
```

If there is a problem in this code, how would you fix it?

22. 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 uses a function that takes as arguments a reference to CandyBar,a pointer-to-char, a double,and an int and uses the last three values to set the corresponding members of the structure.The last three arguments should have default values of “Millennium Munch,” 2.85,and 350. Also the program should use a function that takes a reference to a CandyBar as an argument and displays the contents of the structure. Use const where appropriate.

23. Show the stack with all activation record instances, including static and dynamic chains, when execution reaches position 1 in the following skeletal program. Assume Bigsub is at level 1.

```
function bigsub() {
  function a() {
     function b() {
        ... <----------------------------1
     } // end of b
     function  c() {
       ...
       b();
       ...
     } // end of c
     ...
     c();
     ...
  } // end of a
  ...
  a();
  ...
} // end of bigsub
```

24. Show the stack with all activation record instances, including static and dynamic chains, when execution reaches position 1 in the following skeletal program. Assume Bigsub is at level 1.

```
function bigsub() {
var mysum;
function a() {
 var x;
  function b(sum) {
   var y, z;
   ...
   c(z);
   ...
  } // end of b
  ...
  b(x);
  ...
 } // end of a
 function c(plums) {
  ... <----------------------------1
 } // end of c
 var  l;
...
a();
...
} // end of bigsub
```

25. Although local variables in the methods of C-based languages are allocated at the beginning of each activation, under what circumstances would the value of a local variable in a particular activation retain the value of the previous activation?
26. Show the stack with all activation record instances, including dynamic chains, when execution reaches position 1 in the following skeletal program. This program uses the deep-access method to implement dynamic scoping.

```
void fun1() {
  float a;
  . . .
}

void fun2() {
  int b, c;
  . . . <--------- 1
}

void fun3() {
  float d;
  . . .
}

void main() {
  char e, f, g;
   . . .
}
```

The calling sequence for this program for execution to reach fun3 is

main calls fun1

fun1 calls fun3

fun3 calls fun2

27. It is stated in this chapter that when nonlocal variables are accessed in a dynamic-scoped language using the dynamic chain, variable names must be stored in the activation records with the values. If this were actually done, every nonlocal access would require a sequence of costly string comparisons on names. Design an alternative to these string comparisons that would be faster.
28. Show the stack with all activation record instances, including static and dynamic chains, when execution reaches position 1 in the following skeletal program. Assume bigsub is at level 1.

```
function bigsub() {
function a(flag) {
  function b() {
    ... a(false);
  ...
 } // end of b
 ...
  if (flag)
   b();
  else c();
 ...
} // end of a
 function c() {
  function d() {
   ... <------------------------1
 } // end of d
 ...
 d();


    ...
  } // end of c
  ...
  a(true);
  ...
 } // end of bigsub
```

The calling sequence for this program for execution to reach d is

bigsub calls a
a calls b
b calls a
a calls c
c calls d