Functions 4

REVIEW QUESTIONS

1. The principles of top–down design and structured programming dictate that a program should be divided into a main module and its related modules.

   a. True

3. Function calls that return *void* may not be used as a part of an expression.

   a. True

5. The process of dividing a program into functions—which in turn are divided into functions until they consist of only elementary processing that is intrinsically understood and cannot be further subdivided—is known as

   b. factoring

7. Which of the following is not a part of a function header?

   d. title

9. Which of the following statements about local variables is false?

   e. Local variables' names can be the same as the function's parameter names.

11. Which of the following statements will generate a random number in the range 30–50?

   b. `(rand () % 21) + 30`

EXERCISES

13. The return statement is invalid for the function definition. Either the function definition or the return must be changed.

```
Line 1:  int fun (int x, int y)
```

   -or-

```
Line 5:  return;
```

15. Function `sun` is defined inside function `fun`.

17. The errors are:

   a. int sun (int x, int y);// Missing type for y

   b. int sun (int x, int y); // Missing semicolon

c. void sun (void);// void parameter optional;
// if present, only one allowed

d. void sun (int x, float y); // Parameter types / IDs

19. The value of each expression is:

a. 9.5

b. 2.4

c. 3.4

d. 7.0

e. 7.0

21. For the values x = 3.5, 3.45, 3.76, 3.234, and 3.4567, the floors are:

a. 3.5, 3.5,3.8,3.2,3.5

b. 3.5, 34.5,37.6,32.3,34.6

c. 350, 334,376,323.4,345.7

23. The program output is:

```
-2      2
```

PROBLEMS

25.

```
/* Print myname in a star box.
      Written by: Ima Student
      Date:       Sep 24
*/
#include <iostream>
using namespace std;

// Prototype Statements
void printMyName ();

int main ()
{
   printMyName();
   return 0;
}  // main

/* Prints my name in a star box.
      Pre  Nothing
      Post Name printed
*/
void printMyName ()
{
   cout << "******************************\n";
   cout << "*                            *\n";
   cout << "*         Ima Student        *\n";
   cout << "*                            *\n";
   cout << "******************************\n";
   return;
}  // printMyName
```

27.

```
/* This program generates a random number.
   The range is 1, 4, 7, 10, 13, 16
   (a series of r * 3 + 1).
      Written by:
      Date:
*/
```

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main ()
{
   srand (time (NULL));

   cout << "\nThe random number is "
        << (rand () % 6) * 3 + 1 << endl;
   return 0;
}  // main
```

29. The structure chart is:

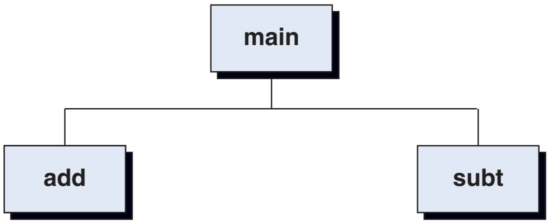

FIGURE 4-1 Solution for Problem 29

31.

```
/* This is a sample of top-down development using
   stubs.
      Written by:
      Date:
*/
#include <iostream>
using namespace std;

// Prototype Declarations
int initialize ();
int process    ();
int endOfjob   ();

int main ()
{
   cout << "\nBegin program \n\n";

   initialize ();
   process    ();
   endOfjob   ();

   return 0;
}  // main

//----------------initialize-------------------
int initialize ()
{
   cout << "\nIn initialize: \n";
   return 0;
}  // initialize

//----------------process----------------------
int process ()
{
   cout << "\nIn process: \n";
   return 0;
}  // process

//----------------endOfjob---------------------
int endOfjob ()
{
   cout << "\nIn endOfjob: \n";
   return 0;
}  // endOfjob
```

33.

```
/* This program adds the least significant three
   digits (hundreds, tens, and ones).
      Written by:
      Date:
*/
#include <iostream>
using namespace std;

// Prototype Declarations
int addThreeDigits  (int a);
int firstDigit      (int a);
int secondDigit     (int a);
int thirdDigit      (int a);

int main ()
{
   cout << "Enter number with at least 3 digits: ";
   int number;
   cin  >> number;

   int sum  = addThreeDigits (number);
   cout << "Sum of 3 digits is " << sum << endl;
   return 0;
}  // main

/* =============== addThreeDigits ===============
   Add the three least significant digits as returned
   from each subfunction.
      Pre  Given a
      Post Returns the sum of rightmost 3 digits
*/
int addThreeDigits (int a)
{
   int result = firstDigit  (a)
       + secondDigit (a)
       + thirdDigit  (a);
   return result;
}  // addThreeDigits

/* ================== firstDigit ==================
   Extracts the least signifcant digit of an integer.
      Pre  a contains an integer
      Post Returns least signifcant digit
```

```
*/
int firstDigit (int a)
{
   return (a % 10);
}  // firstDigit

/* ================ secondDigit ==================
   Extracts the second least signifcant digit of an
   integer.
      Pre  a contains an integer
      Post Returns digit in 10s position
*/
int secondDigit (int a)
{
   int result = (a / 10) % 10;
   return result;
}  // secondDigit

/* ================ thirdDigit ==================
   Extracts the third least signifcant digit of an
   integer.
      Pre  a contains an integer
      Post Returns digit in 100s position
*/
int thirdDigit (int a)
{
   int result = (a / 100) % 10;
   return result;
}  // thirdDigit
```

35.

```
/* Reads a floating-point number and print the
   ceiling, floor, and rounded value.
      Written by:
      Date:
*/
#include <iostream>
#include <iomanip>
#include <cmath>

// Prototype Declarations
long double roundTwo (long double);

int main ()
{
   cout << "Enter number with at least 2 decimals: ";
   long double num;
   cin  >> num;

   long double ceiled  = ceil     (num);
   long double floored = floor    (num);
   long double rounded = roundTwo (num);

   cout << showpoint << setprecision (9);
   cout << "Ceiled  number of " << num << " is "
        << ceiled << endl;
   cout << "Floored number of " << num << " is "
        << floored << endl;
   cout << "Rounded number of " << num << " is "
        << rounded << endl;
   return 0;
}  // main

/* ===============  roundTwo ===============
   This function receives a floating-point number and
   rounds it to two decimal places.
      Pre  floating-point number
      Post the rounded number
*/
long double roundTwo (long double x)
{
   long double temp1   = x * 1000 + 5;
   long int    temp2   = temp1 / 10;
   long double result  = temp2 / 100.00;
   return result;
}  // roundTwo
```