Solutions to Chapter 4

Review Questions

1. a. True

3. a. True

5. b. False

7. d. The function definition contains executable statements that perform the function's task.

9. a. Empty parameter lists are declared with the keyword *void*.

11. d. Indirection operator (`*`)

13. d. `(rand () % 21) + 30`

Exercises

15. This function either must be declared to return an integer or must not return the value of the local variable `z`.

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

19.

a. `int sun (int x, int y);`

b. `int sun (int x, int y);` (semicolon was missing)

c. `void sun (void);` (single void parameter)

d. `void sun (int x, float y);`

21.

a. 9.5

b. 2.4

c. 3.4

d. 7.0

e. 7.0

23.

a. 3.5 3.5 3.8 3.2 3.5

b. 3.5 3.45 3.76 3.23 3.46

c. 3.5 3.45 3.76 3.234 3.457

25. –2 2

27. See Figure 4-1.

Output:

5 0

12 1

10 5 25

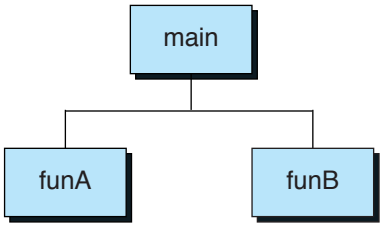

Figure 4-1 Solution for Chapter 4, Exercise 27

Problems

29. See Program 4-1.

Program 4-1 Solution to Problem 29

```
/* This program generates a random number.
   The range is 1 through 6.
      Written by:
      Date:
*/
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main (void)
{
// Local Declarations
   int a;

// Statements
   srand (time (NULL));

   // range is 1 through 6
   a = rand() % 6 + 1;

   printf ("\nThe random number is %d\n", a);

   return 0;
}  // main
```

31. "A function should do only one thing" means that the number of objects it handles must be one. Object refers to anything that exists separately from other elements of the function.

33. See Program 4-2.

Program 4-2 Solution to Problem 33

```
/* This program converts inches into centimeters.
      Written by:
      Date:
*/
#include <stdio.h>

// Function Declarations
float convertInch (float inches);

int main (void)
{
// Local Declarations
```

Program 4-2 Solution to Problem 33 (continued)

```
   float inches;
   float centis;

// Statements
   printf ("Please enter number of inches: ");
   scanf  ("%f", &inches);

   centis = convertInch (inches);

   printf ("%f inches equals %f centimeters\n",
           inches, centis);
   return 0;
}  // main

/* ============== convertInch ==================
   This function converts inches to centimeters.
      Pre  inches is the number of inches
      Post number of centimeters returned
*/
float convertInch (float inches)
{
// Statements
   return (inches * 2.54);
}  // convertInch
```

35. See Program 4-3.

Program 4-3 Solution to Problem 35

```
/* This program calculates the sum, difference,
   product, quotient, and remainder of the two numbers
   read from the keyboard.
      Written by:
      Date:
*/
#include <stdio.h>

// Function Declarations
   int  add    (int a, int b);
   int  subt   (int a, int b);
   int  mult   (int a, int b);
   void divide (int a, int b, int* quot, int* rem);

int main (void)
{
// Local Declarations
   int a;
   int b;
   int sum;
   int diff;
   int product;
   int quotient;
   int remainder;

// Statements
   // Prompt user for input and get data
   printf ("\nPlease enter two integer numbers: ");

   // Read numbers into a and b
   scanf  ("%d %d", &a, &b);

   // Make the calculations
   sum     = add  (a, b);
   diff    = subt (a, b);
   product = mult (a, b);
   divide (a, b, &quotient, &remainder);
```

Program 4-3 Solution to Problem 35 (continued)

```
   printf ("\n\n%d + %d = %d\n", a, b, sum);
   printf ("%d - %d = %d\n", a, b, diff);
   printf ("%d * %d = %d\n", a, b, product);
   printf ("%d / %d = %d with %d left over.\n",
           a, b, quotient, remainder);
   return 0;
}  // main

/* ================== add ==================
   This function adds two integers & returns their sum.
      Pre     Parameters a and b
      Post    Returns (a + b)
*/
int add (int a, int b)
{
   return (a + b);
}  // add

/* ==================== subt ====================
   This function receives two numbers and returns their
   difference.
      Pre    Parameters a and b
      Post   Returns (a - b)
*/
int subt (int a, int b)
{
   return (a - b);
}  // subt

/* ==================== mult ====================
   Receives two numbers & returns their product.
      Pre    Parameters a and b
      Post   Returns (a * b)
*/
int mult (int a, int b)
{
   return (a * b);
}  // mult

/* ==================== divide ====================
   This function receives two numbers and returns their
   quotient and modulus using pass by address.
      Pre    Parameters a and b
      Post   quotient stored in quot;
             modulus stored in rem
*/
void divide (int  a, int  b, int* quot, int* rem)
{
// Statements
   *quot = a / b;
   *rem  = a % b;

   return;
}  // divide
```

37. See Program 4-4.

Program 4-4 Solution to Problem 37

```
/* ================ roundTwo ==============
   This function rounds a floating-point number.
   to two decimal places.
      Pre  floating-point number
      Post the rounded number
*/
```

Program 4-4 Solution to Problem 37

```
long double roundTwo (long double x)
{
// Local Declarations
   long double  result;
   long double  temp1;
   long int     temp2;

// Statements
   temp1  = x * 1000 + 5;
   temp2  = temp1 / 10;
   result = temp2 / 100.00;
   return result;
}  // roundTwo
```

39. See Program 4-5.

Program 4-5 Solutions to Problem 39

```
/* ================= triangle ===================
   This function calculates the perimeter and area of a
   right triangle.
      Pre  a & b are lengths for sides of the triangle
           peri & area contain address of answer fields
      Post peri contains perimeter of triangle
           area contains the area of triangle
*/
void triangle (int a, int b, float* peri, float* area)
{
// Local Declarations
   float c;

// Statements
   *area = (a * b) * 0.5;
   c     =  pow (a, 2.0) + pow (b, 2.0);
   *peri =  a + b + pow (c, 0.5);

   return;
}  // triangle
```