Chapter 3:
Top-Down Design with Functions

Problem Solving & Program Design in C

Seventh Edition

By Jeri R. Hanly &
Elliot B. Koffman

Addison Wesley
is an imprint of
PEARSON

Calculate Kelvin from Fahrenheit

$$k = 5/9 * (f - 32) + 273.15;$$

Outline

- Functions as modular programming
- Standard library functions in C
- Structure charts
- Functions without input or output
- Functions with parameters
  Function with single return values
- Functions with multiple return values

Building a program

- Problem
- Analysis
- Data requirements
  - Constants, Input, output and Formulas
- Design
  - Initial algorithm
  - Refine algorithm
- Implementation (Coding)
- Testing
- Maintenance

Functions

- int main(int argc, char* argv[]){}
- printf();
- scanf();
- pow();

C Library Functions

- #include <header.h>
- stdio.h
- math.h

Parts if a C program

- Preprocessor statements
- Function prototypes
- Main function
- Other functions

Top Down Design & Structure Charts

- Map function calls
- Hierarchal diagram

Functions

- Advantages
  - Procedural abstraction
  - Reuse of code
- Prototypes
- No return and no parameters
- No return with parameters (single & mult)
- Return with parameters

Documentation

- Paragraph describing function
- Precondition and postcondition
- Meaningful variable names
- Comments
- Indentation

Figure 3.1 **Edited Data Requirements and Algorithm for Conversion Program**

```
1.  /*
2.   * Converts distance in miles to kilometers.
3.   */
4.
5.  #include <stdio.h>                 /* printf, scanf definitions */
6.  #define KMS_PER_MILE 1.609         /* conversion constant */
7.
8.  int
9.  main(void)
10. {
11.       double miles;   /* input - distance in miles.         */
12.       double kms;     /* output - distance in kilometers    */
13.
14.       /* Get the distance in miles.                          */
15.
16.       /* Convert the distance to kilometers.                 */
17.          /* Distance in kilometers is
18.                1.609 * distance in miles.                    */
19.
20.       /* Display the distance in kilometers.                 */
21.
22.       return (0);
23. }
```


Calculating the area and circumference of a circle

$$\text{area} = \pi r^2$$

$$\text{circumference} = 2\pi r$$

Figure 3.2 **Outline of Program Circle**

```
1.  /*
2.   * Calculates and displays the area and circumference of a circle
3.   */
4.
5.  #include <stdio.h> /* printf, scanf definitions */
6.  #define PI 3.14159
7.
8.  int
9.  main(void)
10. {
11.       double radius;    /* input - radius of a circle  */
12.       double area;      /* output - area of a circle   */
13.       double circum;    /* output - circumference      */
14.
15.       /* Get the circle radius */
16.
17.       /* Calculate the area */
18.          /* Assign PI * radius * radius to area. */
19.
20.       /* Calculate the circumference */
21.          /* Assign 2 * PI * radius to circum */
22.
23.       /* Display the area and circumference */
24.
25.       return (0);
26. }
```


Figure 3.3 **Calculating the Area and the Circumference of a Circle (cont'd)**

```
1.  /*
2.   * Calculates and displays the area and circumference of a circle
3.   */
4.
5.  #include <stdio.h> /* printf, scanf definitions */
6.  #define PI 3.14159
7.
8.  int
9.  main(void)
                                                              (continued)
10. {
11.       double radius; /* input - radius of a circle */
12.       double area;   /* output - area of a circle  */
13.       double circum; /* output - circumference     */
14.
15.       /* Get the circle radius */
16.       printf("Enter radius> ");
17.       scanf("%lf", &radius);
18.
19.       /* Calculate the area */
20.       area = PI * radius * radius;
21.
22.       /* Calculate the circumference */
23.       circum = 2 * PI * radius;
24.
25.       /* Display the area and circumference */
26.       printf("The area is %.4f\n", area);
27.       printf("The circumference is %.4f\n", circum);
28.
29.       return (0);
30. }

Enter radius> 5.0
The area is 78.5397
The circumference is 31.4159
```

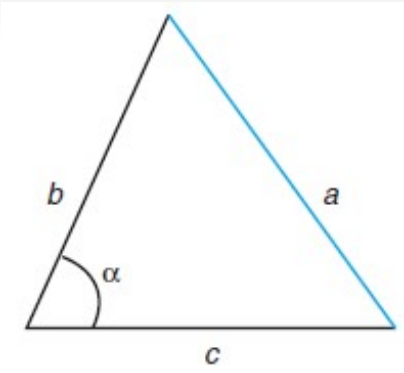
Calculate the hypotenuse of a triangle

$$a^2 = b^2 + c^2$$

Calculate the distance an object falls in a vacuum

$$y = 1/2Gt^2$$

Figure 3.8 **Triangle with Unknown Side *a***



$$a^2 = b^2 + c^2 - 2bc \cos \alpha$$

Figure 3.6 **Function sqrt as a "Black Box"**

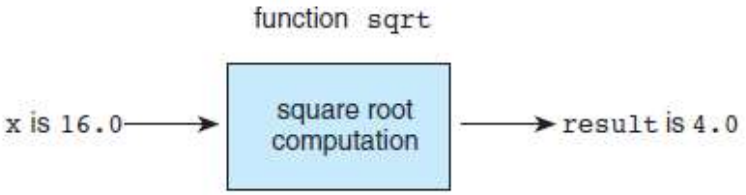


Figure 3.7 **Square Root Program**

```
1.  /*
2.   * Performs three square root computations
3.   */
4.
5.  #include <stdio.h> /* definitions of printf, scanf */
6.  #include <math.h>  /* definition of sqrt           */
7.
8.  int
9.  main(void)
10. {
11.       double first, second,    /* input - two data values        */
12.              first_sqrt,       /* output - square root of first  */
13.              second_sqrt,      /* output - square root of second */
14.              sum_sqrt;         /* output - square root of sum    */
15.
16.       /* Get first number and display its square root. */
17.       printf("Enter the first number> ");
18.       scanf("%lf", &first);
19.       first_sqrt = sqrt(first);
20.       printf("The square root of the first number is %.2f\n", first_sqrt);
```

(continued)

Figure 3.7 **Square Root Program (cont'd)**

```
21.       /* Get second number and display its square root. */
22.       printf("Enter the second number> ");
23.       scanf("%lf", &second);
24.       second_sqrt = sqrt(second);
25.       printf("The square root of the second number is %.2f\n", second_sqrt);
26.
27.       /* Display the square root of the sum of the two numbers. */
28.       sum_sqrt = sqrt(first + second);
29.       printf("The square root of the sum of the two numbers is %.2f\n",
30.              sum_sqrt);
31.
32.       return (0);
33. }

Enter the first number> 9.0
The square root of the first number is 3.00
Enter the second number> 16.0
The square root of the second number is 4.00
The square root of the sum of the two numbers is 5.00
```

Figure 3.4 **Computing the Rim Area of a Flat Washer**

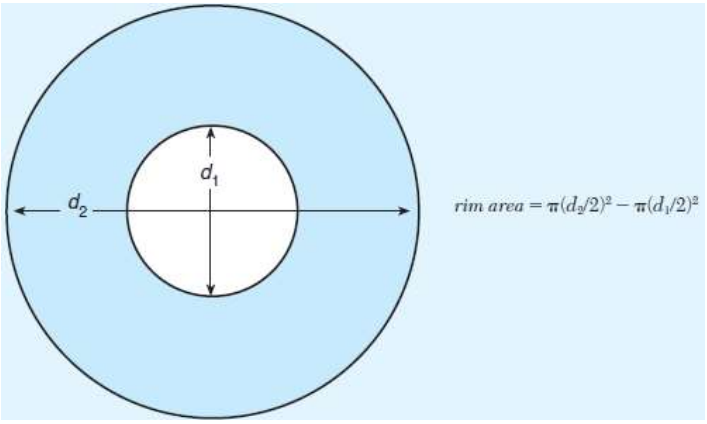


Figure 3.5 **Flat Washer Program**

```
1.  /*
2.   * Computes the weight of a batch of flat washers.
3.   */
4.
5.  #include <stdio.h> /* printf, scanf definitions */
6.  #define PI 3.14159
7.
8.  int
9.  main(void)
10. {
11.       double hole_diameter; /* input - diameter of hole         */
12.       double edge_diameter; /* input - diameter of outer edge   */
13.       double thickness;     /* input - thickness of washer      */
14.       double density;       /* input - density of material used */
15.       double quantity;      /* input - number of washers made   */
16.       double weight;        /* output - weight of washer batch  */
17.       double hole_radius;   /* radius of hole                   */
18.       double edge_radius;   /* radius of outer edge             */
19.       double rim_area;      /* area of rim                      */
20.       double unit_weight;   /* weight of 1 washer               */
21.
22.       /* Get the inner diameter, outer diameter, and thickness.*/
23.       printf("Inner diameter in centimeters> ");
24.       scanf("%lf", &hole_diameter);
25.       printf("Outer diameter in centimeters> ");
26.       scanf("%lf", &edge_diameter);
27.       printf("Thickness in centimeters> ");
28.       scanf("%lf", &thickness);
29.
30.       /* Get the material density and quantity manufactured. */
31.       printf("Material density in grams per cubic centimeter> ");
32.       scanf("%lf", &density);
33.       printf("Quantity in batch> ");
34.       scanf("%lf", &quantity);
35.
36.       /* Compute the rim area. */
37.       hole_radius = hole_diameter / 2.0;
38.       edge_radius = edge_diameter / 2.0;
```

(continued)

Figure 3.5 **Flat Washer Program (cont'd)**

```
39.       rim_area = PI * edge_radius * edge_radius -
40.                  PI * hole_radius * hole_radius;
41.
42.       /* Compute the weight of a flat washer. */
43.       unit_weight = rim_area * thickness * density;
44.       /* Compute the weight of the batch of washers. */
45.       weight = unit_weight * quantity;
46.
47.       /* Display the weight of the batch of washers. */
48.       printf("\nThe expected weight of the batch is %.2f", weight);
49.       printf(" grams.\n");
50.
51.       return (0);
52. }

Inner diameter in centimeters> 1.2
Outer diameter in centimeters> 2.4
Thickness in centimeters> 0.1
Material density in grams per cubic centimeter> 7.87
Quantity in batch> 1000

The expected weight of the batch is 2670.23 grams.
```