

Intro to C Programming

2

Objectives

In this chapter, you'll:

- Write simple C programs.
- Use simple input and output statements.
- Use the fundamental data types.
- Learn computer memory concepts.
- Use arithmetic operators.
- Learn the precedence of arithmetic operators.
- Write simple decision-making statements.
- Begin focusing on secure C programming practices.



2 Chapter 2 Intro to C Programming

Instructor Note: This file contains only the solutions to the short answer questions. For programming exercises that require complete programs, the source code for those programs is located in the `ch02solutions` folder.

Exercises

2.7 Identify and correct the errors in each of the following statements. (*Note:* There may be more than one error per statement.)

a) `scanf("d", value);`

Answer: `scanf("%d", &value);`

b) `printf("The product of %d and %d is %d\n", x, y);`

Answer: `printf("The product of %d and %d is %d\n", x, y, x * y);`

c) `firstNumber + secondNumber = sumOfNumbers`

Answer: `sumOfNumbers = firstNumber + secondNumber`

d) `if (number => largest) {
 largest == number;
}`

Answer:

```
if (number >= largest) {  
    largest == number;  
}
```

e) `/* Program to determine the largest of three integers */`

Answer: `/* Program to determine the largest of three integers */`

f) `Scanf("%d", anInteger);`

Answer: `scanf("%d", &anInteger);`

g) `printf("Remainder of %d divided by %d is\n", x, y, x % y);`

Answer: `printf("Remainder of %d divided by %d is %d\n", x, y, x % y);`

h) `if (x = y); {
 printf("%d is equal to %d\n", x, y);
}`

Answer: `if (x == y); {
 printf("%d is equal to %d\n", x, y);
}`

i) `print("The sum is %d\n", x + y);`

Answer: `print("The sum is %d\n", x + y);`

j) `Printf("The value you entered is: %d\n", &value);`

Answer: `printf("The value you entered is: %d\n", value);`

2.8 Fill-In the blanks in each of the following:

a) _____ are used to document a program and improve its readability.

Answer: comments.

b) The function used to display information on the screen is _____.

Answer: `printf`.

c) A C statement that makes a decision is _____.

Answer: `if`.

d) Calculations are normally performed by _____ statements.

Answer: assignment.

e) The _____ function inputs values from the keyboard.

Answer: `scanf`.

2.9 Write a single C statement or line that accomplishes each of the following:

a) Display the message "Enter two numbers."

Answer: `puts("Enter two numbers");`

b) Assign the product of variables `b` and `c` to variable `a`.

Answer: `a = b * c;`

c) State that a program performs a sample payroll calculation (i.e., use text that helps to document a program).

Answer: `// Sample payroll calculation program`

d) Input three integer values and place them in `int` variables `a`, `b` and `c`.

Answer: `scanf("%d%d%d", &a, &b, &c);`

2.10 State which of the following are *true* and which are *false*. If *false*, explain why.

a) C operators evaluate left-to-right.

Answer: False. Some operators are evaluated left to right and others are evaluated from right to left depending on their grouping.

b) Each of the following is a valid variable name: `_under_bar_`, `m928134`, `t5`, `j7`, `her_sales`, `his_account_total`, `a`, `b`, `c`, `z`, `z2`.

Answer: True.

c) The statement `printf("a = 5;");` is an example of an assignment statement.

Answer: False. The statement prints `a = 5;` on the screen.

d) An arithmetic expression containing no parentheses evaluates left-to-right.

Answer: False. Multiplication, division, and modulus are all evaluated first from left to right, then addition and subtraction are evaluated from left to right.

e) The following are all invalid variable names: `3g`, `87`, `67h2`, `h22`, `2h`.

Answer: False. Only those beginning with a number are invalid.

2.11 Fill-In the blanks in each of the following:

a) What arithmetic operations are on the same level of precedence as multiplication? _____.

Answer: division, modulus.

b) When parentheses are nested, which set of parentheses is evaluated first in an arithmetic expression? _____.

Answer: The innermost pair of parentheses.

c) A location in the computer's memory that may contain different values at various times throughout the execution of a program is called a _____.

Answer: variable.

2.12 What, if anything, displays when each of the following statements is performed? If nothing displays, then answer "Nothing." Assume `x = 2` and `y = 3`.

4 Chapter 2 Intro to C Programming

a) `printf("%d", x);`

Answer: 2.

b) `printf("%d", x + x);`

Answer: 4.

c) `printf("%s", "x=");`

Answer: x=.

d) `printf("x=%d", x);`

Answer: x=2.

e) `printf("%d = %d", x + y, y + x);`

Answer: 5 = 5.

f) `z = x + y;`

Answer: Nothing. Value of $x + y$ is assigned to z .

g) `scanf("%d%d", &x, &y);`

Answer: Nothing. Two integer values are read into the variables x and y .

h) `// printf("x + y = %d", x + y);`

Answer: Nothing. This is a comment.

i) `printf("%s", "\n");`

Answer: This prints a newline character, positioning the cursor at the beginning of the next line on the screen.

2.13 Which of the following C statements contain variables whose values are replaced?

a) `scanf("%d%d%d%d%d", &b, &c, &d, &e, &f);`

b) `p = i + j + k + 7;`

c) `printf("%s", "Values are replaced");`

d) `printf("%s", "a = 5");`

Answer: (a) and (b).

2.14 Given the equation $y = ax^3 + 7$, which of the following, if any, are correct C statements for this equation?

a) `y = a * x * x * x + 7;`

b) `y = a * x * x * x * (x + 7);`

c) `y = (a * x) * x * (x + 7);`

d) `y = (a * x) * x * x + 7;`

e) `y = a * (x * x * x) + 7;`

f) `y = a * x * (x * x + 7);`

Answer: (a), (d), and (e).

2.15 State the order of evaluation of the operators in each of the following C statements and show the value of x after each statement is performed.

a) `x = 7 + 3 * 6 / 2 - 1;`

Answer: $*$ is first, $/$ is second, $+$ is third, $-$ is fourth and $=$ is last. Value of x is 15.

b) `x = 2 % 2 + 2 * 2 - 2 / 2;`

Answer: $\%$ is first, $*$ is second, $/$ is third, $+$ is fourth, $-$ is fifth and $=$ is last. Value of x is 3.

c) `x = (3 * 9 * (3 + (9 * 3 / (3))));`

Answer:

`x = (3 * 9 * (3 + (9 * 3 / (3))));`

5 6 4 2 3 1. The = evaluates last. Value of x is 324.

2.16 (Arithmetic) Write a program that reads two integers from the user then displays their sum, product, difference, quotient and remainder.

2.17 (Displaying Values with printf) Write a program that displays the numbers 1 to 4 on the same line. Write the program using the following methods.

a) Using one printf statement with no conversion specifications.

Answer: `printf("1 2 3 4\n\n");`

b) Using one printf statement with four conversion specifications.

Answer: `printf("%d %d %d %d\n\n", 1, 2, 3, 4);`

c) Using four printf statements.

Answer:

`printf("1 ");`

`printf("2 ");`

`printf("3 ");`

`printf("4\n");`

2.18 (Comparing Integers) Write a program that reads two integers from the user then displays the larger number followed by the words “is larger.” If the numbers are equal, display the message “These numbers are equal.” Use only the single-selection form of the if statement you learned in this chapter.

2.19 (Arithmetic, Largest Value and Smallest Value) Write a program that inputs three different integers from the keyboard, then displays the sum, the average, the product, the smallest and the largest of these numbers. Use only the single-selection form of the if statement you learned in this chapter. The screen dialogue should appear as follows:

```
Enter three different integers: 13 27 14
Sum is 54
Average is 18
Product is 4914
Smallest is 13
Largest is 27
```

2.20 (Circle Area, Diameter and Circumference) For a circle of radius 2, display the diameter, circumference and area. Use the value 3.14159 for π . Use the following formulas (r is the radius): $diameter = 2r$, $circumference = 2\pi r$ and $area = \pi r^2$. Perform each of these calculations inside the printf statement(s) and use the conversion specification %f. This chapter discussed only integer constants and variables. Chapter 3 will discuss floating-point numbers—that is, values that can have decimal points.

2.21 What does the following code display?

`printf("%s", "\n**\n***\n****\n*****\n");`

Answer:

```
*
**
***
****
*****
```

2.22 (Odd or Even) Write a program that reads an integer and determines and displays whether it's odd or even. Use the remainder operator. An even number is a multiple of two. Any multiple of two leaves a remainder of zero when divided by 2.

2.23 (Multiples) Write a program that reads two integers and determines and displays whether the first is a multiple of the second. Use the remainder operator.

2.24 Distinguish between the terms fatal error and nonfatal error. Why might you prefer to experience a fatal error rather than a nonfatal error?

Answer: A fatal error causes the program to terminate prematurely. A nonfatal error occurs when the logic of the program is incorrect, and the program does not work properly. A fatal error is preferred for debugging purposes. A fatal error immediately lets you know there is a problem with the program, whereas a nonfatal error can be subtle and possibly go undetected.

2.25 (Integer Value of a Character) Here's a peek ahead. In this chapter, you learned about integers and the type `int`. C can also represent uppercase letters, lowercase letters and a considerable variety of special symbols. C uses small integers internally to represent each different character. The set of characters a computer uses together with the corresponding integer representations for those characters is called that computer's character set. You can display the integer equivalent of uppercase A, for example, by executing the statement

```
printf("%d", 'A');
```

Write a C program that displays the integer equivalents of some uppercase letters, lowercase letters, digits and special symbols. At a minimum, determine the integer equivalents of the following: A B C a b c 0 1 2 \$ * + / and the space character.

2.26 (Separating Digits in an Integer) Write a program that inputs one five-digit number, separates the number into its individual digits and displays the digits separated from one another by three spaces each. [*Hint:* Use combinations of integer division and the remainder operation.] For example, if the user types in 42139, the program should display

```
4   2   1   3   9
```

2.27 (*Table of Squares and Cubes*) Using only the techniques you learned in this chapter, write a program that calculates the squares and cubes of the numbers from 0 to 10 and uses tabs to display the following table of values:

number	square	cube
0	0	0
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000

2.28 (*Target Heart-Rate Calculator*) While exercising, you can use a heart-rate monitor to see that your heart rate stays within a safe range suggested by your doctors and trainers. According to the American Heart Association (AHA) (<http://bit.ly/AHATargetHeartRates>), the formula for calculating your maximum heart rate in beats per minute is 220 minus your age in years. Your target heart rate is 50–85% of your maximum heart rate. Write a program that prompts for and inputs the user's age and calculates and displays the user's maximum heart rate and the range of the user's target heart rate. [These formulas are estimates provided by the AHA; maximum and target heart rates may vary based on the health, fitness and gender of the individual. Always consult a physician or qualified healthcare professional before beginning or modifying an exercise program.]

2.29 (*Sort in Ascending Order*) Write a program that inputs three different numbers from the user. Display the numbers in increasing order. Recall that an `if` statement's body can contain more than one statement. Prove that your script works by running it on all six possible orderings of the numbers. Does your script work with duplicate numbers? [This is challenging. In later chapters you'll do this more conveniently and with many more numbers.]