

Introduction to Java Applications



Objectives

In this appendix you'll learn:

- To write simple Java applications.
- To use input and output statements.
- Java's primitive types.
- Basic memory concepts.
- To use arithmetic operators.
- The precedence of arithmetic operators.
- To write decision-making statements.
- To use relational and equality operators.



Self-Review Exercises

A.1 Fill in the blanks in each of the following statements:

- a) A(n) _____ begins the body of every method, and a(n) _____ ends the body of every method.

ANS: left brace ({}), right brace ({}).

- b) You can use the _____ statement to make decisions.

ANS: if.

- c) _____ begins an end-of-line comment.

ANS: //.

- d) _____, _____ and _____ are called white space.

ANS: Space characters, newlines and tabs.

- e) _____ are reserved for use by Java.

ANS: Keywords.

- f) Java applications begin execution at method _____.

ANS: main.

- g) Methods _____, _____ and _____ display information in a command window.

ANS: System.out.print, System.out.println and System.out.printf.

A.2 State whether each of the following is *true* or *false*. If *false*, explain why.

- a) Comments cause the computer to print the text after the // on the screen when the program executes.

ANS: False. Comments do not cause any action to be performed when the program executes. They're used to document programs and improve their readability.

- b) All variables must be given a type when they're declared.

ANS: True.

- c) Java considers the variables `number` and `NumBEr` to be identical.

ANS: False. Java is case sensitive, so these variables are distinct.

- d) The remainder operator (%) can be used only with integer operands.

ANS: False. The remainder operator can also be used with noninteger operands in Java.

- e) The arithmetic operators *, /, %, + and - all have the same level of precedence.

ANS: False. The operators *, / and % are higher precedence than operators + and -.

A.3 Write statements to accomplish each of the following tasks:

- a) Declare variables `c`, `thisIsAVariable`, `q76354` and `number` to be of type `int`.

ANS: `int c, thisIsAVariable, q76354, number;`

or

`int c;`

`int thisIsAVariable;`

`int q76354;`

`int number;`

- b) Prompt the user to enter an integer.

ANS: `System.out.print("Enter an integer: ");`

- c) Input an integer and assign the result to `int` variable `value`. Assume `Scanner` variable `input` can be used to read a value from the keyboard.

ANS: `value = input.nextInt();`

- d) Print "This is a Java program" on one line in the command window. Use method `System.out.println`.

ANS: `System.out.println("This is a Java program");`

- e) Print "This is a Java program" on two lines in the command window. The first line should end with Java. Use method `System.out.println`.

ANS: `System.out.println("This is a Java\nprogram");`

- f) Print "This is a Java program" on two lines in the command window. The first line should end with Java. Use method `System.out.printf` and two `%s` format specifiers.

ANS: `System.out.printf("%s\n%s\n", "This is a Java", "program");`

- g) If the variable `number` is not equal to 7, display "The variable number is not equal to 7".

ANS: `if (number != 7)`

`System.out.println("The variable number is not equal to 7");`

A.4 Identify and correct the errors in each of the following statements:

- a) `if (c < 7);`

`System.out.println("c is less than 7");`

ANS: Error: Semicolon after the right parenthesis of the condition (`c < 7`) in the `if`.

Correction: Remove the semicolon after the right parenthesis. [Note: As a result, the output statement will execute regardless of whether the condition in the `if` is true.]

- b) `if (c == 7)`

`System.out.println("c is equal to or greater than 7");`

ANS: Error: The relational operator `==` is incorrect. Correction: Change `==` to `>=`.

A.5 Write declarations, statements or comments that accomplish each of the following tasks:

- a) State that a program will calculate the product of three integers.

ANS: `// Calculate the product of three integers`

- b) Create a Scanner called `input` that reads values from the standard input.

ANS: `Scanner input = new Scanner(System.in);`

- c) Declare the variables `x`, `y`, `z` and `result` to be of type `int`.

ANS: `int x, y, z, result;`

or

`int x;`

`int y;`

`int z;`

`int result;`

- d) Prompt the user to enter the first integer.

ANS: `System.out.print("Enter first integer: ");`

- e) Read the first integer from the user and store it in the variable `x`.

ANS: `x = input.nextInt();`

- f) Prompt the user to enter the second integer.

ANS: `System.out.print("Enter second integer: ");`

- g) Read the second integer from the user and store it in the variable `y`.

ANS: `y = input.nextInt();`

- h) Prompt the user to enter the third integer.

ANS: `System.out.print("Enter third integer: ");`

- i) Read the third integer from the user and store it in the variable `z`.

ANS: `z = input.nextInt();`

- j) Compute the product of the three integers contained in variables `x`, `y` and `z`, and assign the result to the variable `result`.

ANS: `result = x * y * z;`

- k) Display the message "Product is" followed by the value of the variable `result`.

ANS: `System.out.printf("Product is %d\n", result);`

4 Chapter A Introduction to Java Applications

A.6 Using the statements you wrote in Exercise A.5, write a complete program that calculates and prints the product of three integers.

```
1 // Ex. 2.6: Product.java
2 // Calculate the product of three integers.
3 import java.util.Scanner; // program uses Scanner
4
5 public class Product
6 {
7     public static void main( String[] args )
8     {
9         // create Scanner to obtain input from command window
10        Scanner input = new Scanner( System.in );
11
12        int x; // first number input by user
13        int y; // second number input by user
14        int z; // third number input by user
15        int result; // product of numbers
16
17        System.out.print( "Enter first integer: " ); // prompt for input
18        x = input.nextInt(); // read first integer
19
20        System.out.print( "Enter second integer: " ); // prompt for input
21        y = input.nextInt(); // read second integer
22
23        System.out.print( "Enter third integer: " ); // prompt for input
24        z = input.nextInt(); // read third integer
25
26        result = x * y * z; // calculate product of numbers
27
28        System.out.printf( "Product is %d\n", result );
29    } // end method main
30 } // end class Product
```

```
Enter first integer: 10
Enter second integer: 20
Enter third integer: 30
Product is 6000
```

Exercises

NOTE: Solutions to the programming exercises are located in the AppAsolutions folder. Each exercise has its own folder named exA_## where ## is a two-digit number representing the exercise number. For example, exercise A.11's solution is located in the folder exA_11.

A.7 Fill in the blanks in each of the following statements:

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

ANS: Comments.

b) A decision can be made in a Java program with a(n) _____.

ANS: if statement.

c) Calculations are normally performed by _____ statements.

ANS: assignment statements.

d) The arithmetic operators with the same precedence as multiplication are _____ and _____.

ANS: division (/), remainder (%)

- e) When parentheses in an arithmetic expression are nested, the _____ set of parentheses is evaluated first.

ANS: innermost.

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

ANS: variable.

A.8 Write Java statements that accomplish each of the following tasks:

- a) Display the message "Enter an integer: ", leaving the cursor on the same line.

ANS: `System.out.print("Enter an integer: ");`

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

ANS: `a = b * c;`

- c) Use a comment to state that a program performs a sample payroll calculation.

ANS: `// This program performs a simple payroll calculation.`

A.9 State whether each of the following is *true* or *false*. If *false*, explain why.

- a) Java operators are evaluated from left to right.

ANS: False. Some operators (e.g., assignment, =) evaluate from right to left.

- b) The following are all valid variable names: `_under_bar_`, `m928134`, `t5`, `j7`, `her_sales$`, `his_$account_total`, `a`, `b$`, `c`, `z` and `z2`.

ANS: True.

- c) A valid Java arithmetic expression with no parentheses is evaluated from left to right.

ANS: False. The expression is evaluated according to operator precedence.

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

ANS: False. Identifier `h22` is a valid variable name.

A.10 Assuming that `x = 2` and `y = 3`, what does each of the following statements display?

- a) `System.out.printf("x = %d\n", x);`

ANS: `x = 2`

- b) `System.out.printf("Value of %d + %d is %d\n", x, x, (x + x));`

ANS: Value of 2 + 2 is 4

- c) `System.out.printf("x =");`

ANS: `x =`

- d) `System.out.printf("%d = %d\n", (x + y), (y + x));`

ANS: `5 = 5`

A.12 What does the following code print?

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

ANS:

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


