

## Revision Exercise 1

(1) Fill in the blanks in each of the following.

- a. The \_\_\_\_\_ begins the body of every method and the ends the body of every method.
- b. Every statement ends with a \_\_\_\_\_.
- c. The \_\_\_\_\_ structure is used to make decisions.
- d. \_\_\_\_\_ begins a single-line comment.
- e. \_\_\_\_\_, \_\_\_\_\_, \_\_\_\_\_, and \_\_\_\_\_ are known as white-space.
- f. \_\_\_\_\_ are reserved for use by Java.
- g. Java applications begin execution at method \_\_\_\_\_.
- h. Methods \_\_\_\_\_ and \_\_\_\_\_ display information in the command window.
- i. A \_\_\_\_\_ method is always called using its class name followed by a dot (.) and its method name.
- j. \_\_\_\_\_ are used to document a program and improve its readability.
- k. A Java statement that makes a decision is \_\_\_\_\_.
- l. Calculations are normally performed by \_\_\_\_\_ statements.

(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 is executed.
- b. All variables must be given a type when they are declared.
- c. Java considers the variables **number** and **NumBEr** to be identical.
- d. The modulus operator (%) can be used only with integer operands.
- e. The arithmetic operators `*`, `/`, `%`, `+` and `-` all have the same level of precedence.
- f. Method **Integer.parseInt** converts an integer to a **String**.

(3) Write Java statements to accomplish each of the following:

- a. Declare variables **c**, **thisIsAVariable**, **q76354** and **number** to be of type **int**.
- b. Prompt the user to enter an integer. Hint: Declare a **Label** reference, create a **Label** object, and specify how to place the **Label** on the applet.
- c. Convert a **String** to an integer and store the converted value in integer variable **age**. Assume that the **String** is stored in **stringValue**.
- d. If the variable **number** is not equal to 7, display **"The variable number**

**is not equal to 7**" in a message dialog.

- e. Print the message **"This is a Java program"** on one line from the **paint** method. Assume **Graphics** object **g** as the argument to the **paint** method and choose your own coordinates.
- f. Print the message **"This is a Java program"** on two lines on two lines in the command window where the first line ends with **Java**. Use only one statement.
- g. Read an integer from the keyboard and store the value entered in integer variable **age**. Assume that the **TextField** called **input** exists and that the **ActionEvent** argument **e** in method **actionPerformed** contains the information about the event that occurred.

(4) Identify and correct the errors in each of the following statements:

- a. `if ( c < 7 );`  
    `g.drawString( "c is less than 7" , 25 , 25);`
- b. `if ( c => 7 )`  
    `g.drawString( "c is equal to or greater than 7" , 25 ,`  
    `25);`

(5) Write four different Java statements that each add 1 to integer variable **x**.

(6) Write a statement to accomplish each of the following:

- a. Declare the variables **x**, **y**, **z** and **result** to be of type **int**.
- b. Declare the variables **xVal**, **yVal** and **zVal** to be of type **String**.
- c. Convert **xVal** to an **int** and store the result in the variable **x**.
- d. Compute the product of the three integers contained in variables **x**, **y** and **z**, and assign the result to the variable **result**.
- e. Assign the sum of **x** and **y** to **z** and increment the value of **x** by 1 after the calculation.
- f. Test if the value of the variable **count** is greater than 10. If it is , print **"Count is greater than 10"**.
- g. Decrement the variable **x** by 1 then subtract it from the variable **total**.
- h. Calculate the remainder after **q** is divided by divisor and assign the result to **q**. Write this statements two different ways.

- (7) Write a Java statement to accomplish each of the following tasks.
- Declare variables **sum** and **x** to be of type **int**.
  - Assign 1 to variable **x**.
  - Assign 0 to variable **sum**.
  - Add variable **x** to variable **sum** and assign the result to variable **sum**.
  - Print "**The sum is:** " followed by the value of variable **sum**.
- (8) Combine the statements that you wrote in Exercise 7 into Java application that calculates and prints the sum of the integers from 1 to 10. Use **while** structure to loop through the calculation and increment statements. The loop should terminate when the value of **x** becomes 11.
- (9) Determine the values of each variable after the calculation is performed. Assume that when each statement begins executing all variables have the integer value 5.
- product \*=x++;**
  - quotient /= ++x;**
- (10) Write an application that asks the user to enter two integers, obtains the numbers from the user and displays the larger number followed by the words "**is larger**". If the numbers are equal, print the message "**These numbers are equal**".
- (11) Write an application that inputs from the user the radius of a circle and prints the circle's diameter, circumference and area. Use the constant value 3.14159 for  $\pi$ . [Note: You may also use the predefined constant **Math.PI** for the value of  $\pi$ . This constant is more precise than the value 3.14159. Class **Math** is defined in the **java.lang** package, so you do not need to **import** it.] Use the following formulas ( $r$  is the radius):  $diameter = 2r$ ,  $circumference = 2\pi r$ ,  $area = \pi r^2$ .
- (12) Write an application that reads an integer and determines and prints whether it is odd or even. (Hint: Use the modulus operator. An even number is a multiple of two. Any multiple of two leaves a remainder of zero when divided by 2.)
- (13) You learned about integers and the data type **int**. Java can also represent uppercase letters, lowercase letters and a considerable variety of special symbols. Every character has a corresponding integer representation. The set of characters a computer uses and the corresponding integer representations for those characters is called that computer's character set. You can indicate a character value in a program by simply enclosing that character in single quotes as with '**A**'.

You can determine the integer equivalent of a character by preceding that character with **(int)**—this is called a cast .

```
(int) 'A'
```

The following statement would output a character and its integer equivalent

```
System.out.println(  
    "The character " + 'A' + " has the value " + (int)  
    'A' );
```

When the preceding statement executes, it displays the character **A** and the value **65** (from the so- called Unicode character set) as part of the string.

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

### Answers

(1) a) Left brace (**{**), right brace (**}**). b) Semicolon (**;**). c) **if**. d) **//**. e) Blank lines, space characters, newline characters and tab characters. f) Keywords. g) **main**. hi) **System.out.print** and **System.out.println**. i) **static**. j) Comments. k) **if**. l) assignment.

(2)

- a. False. Comments do not cause any action to be performed when the program is executed. They are not used to document programs and improve their readability.
- b. True.
- c. False. Java is case sensitive, so these variables are distinct.
- d. False. The modulus operator can also be used with non-integer operands in Java.
- e. False. The operators **\***, **/** and **%** are on the same level of precedence and the operators **+** and **-** are on a lower level of precedence.
- f. False. **Integer.parseInt** method converts a **String** to an integer (**int**) value.

(3)

- a. **int c, thisIsAVariable, q76354, number;**

b. `Label prompt;`

```
prompt = new Label( "Enter an integer" );
```

```
add(prompt);
```

c. `age = Integer.parseInt( stringValue );`

d. `if ( number != 7 )`

```
    g.drawString("The variable number is not equal to 7" ,  
    10 , 10);
```

e. `g.drawString( "This is a Java program",10,10);`

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

g. `Age = Integer.parseInt(e.getActionCommand() );`

(4)

a. Error: Semicolon after the right parenthesis of the condition in the **if** statement.

Correction: Remove the semicolon after the right parenthesis. Note: The result of this error is that the output statement will be executed whether or not the condition in the **if** statement is true. The semicolon after the right parenthesis is considered an empty statement—a statement that does nothing. We will learn more about the empty statement in the next chapter.

b. Error: The relational operator `=>` is incorrect.

Correction: Change `=>` to `>=`.

(5) `x = x + 1; Or x += 1; Or ++x; Or x++;`

(6)

a. `int x, y, z, result;`

b. `String xVal, yVal, zVal;`

c. `x = Integer.parseInt( xVal );`

d. `result = x * y * z;`

e. `z = x++ + y;`

f. `if (count > 10)`

```
    System.out.println( "Count is greater than 10" );
```

```
g. total -= --x;
h. q %= divisor; or q = q % divisor;
```

(7)

```
a. int sum , x;
b. x = 1;
c. sum = 0;
d. sum += x; or sum = sum + x;
e. System.out.println("The sum is: " + sum);
```

(8)

//Calculate the sum of the integers from 1 to 10

```
public class Calculate {
    public static void main(String args[])
    {
        int sum , x;
        x = 1;
        sum = 0;
        while (x <= 10) {
            sum += x;
            ++x;
        }
        System.out.println("The sum is: " + sum);
    }
}
```

(9) a) **product = 25 , x = 6;**      b) **quotient = 0 , x = 6;**

(10)

// Larger.java

// Program determines the larger of two numbers

```
import javabook.*;
```

```
class Larger
```

```
{
    public static void main (String args[])
    {
        MainWindow mainWindow = new MainWindow("Which larger?");
        InputBox inputBox = new InputBox(mainWindow);
        OutputBox outputBox = new OutputBox(mainWindow);
        mainWindow.show();
        outputBox.show();
    }
}
```

```

int      number1,
        number2;

//get input
number1      = inputBox.getInteger("Enter first number:");
number2      = inputBox.getInteger("Enter second number:");

//display the result
if ( number1 > number2 )
    outputBox.println(number1 + " is larger.");
else if ( number1 < number2 )
    outputBox.println(number2 + " is larger.");
else
    outputBox.println("These numbers are equal.");
}
}
(11)
// Circle.java
// Program calculate the area, circumference, and diameter for a circle
import javabook.*;
class Circle
{
    public static void main (String args[])
    {
        MainWindow mainWindow = new MainWindow("Circle program");
        InputBox  inputBox      = new InputBox(mainWindow);
        OutputBox outputBox      = new OutputBox(mainWindow);
        mainWindow.show();
        outputBox.show();
        int      radius;
        //get input
        radius      = inputBox.getInteger("Enter radius:");
        //display the result
        outputBox.println( " Radius = " + Format.leftAlign(6 , 2 , 2*radius));
        outputBox.println(" Area = " + Format.leftAlign(6 , 2 , Math.PI *
radius * radius));
        outputBox.println("Circumference = " + Format.leftAlign(6 , 2 , 2*
Math.PI * radius));
    }
}

```

```

}
(12)
// OddEven.java
// Program determines if a number is odd or even
import javabook.*;
class OddEven
{
    public static void main (String args[])
    {
        MainWindow mainWindow = new MainWindow("Circle program");
        InputBox  inputBox      = new InputBox(mainWindow);
        OutputBox outputBox      = new OutputBox(mainWindow);
        mainWindow.show();
        outputBox.show();
        int        number;
        String      result;
        //get input
        number      = inputBox.getInteger("Enter an integer:");
        if ( number % 2 == 0 )
            result = "Number is even.";
        else
            result = "Number is odd.";
        //display the result
        outputBox.println( result);
    }
}

```

```

(13)
// Display.java
// Program prints a unicode character
// and its integer equivalent
public class Display {
    public static void main( String args[] )
    {
        System.out.println( "The character " + 'A' +
            " has the value " + ( int ) 'A' );
        System.out.println( "The character " + 'B' +
            " has the value " + ( int ) 'B' );
        System.out.println( "The character " + 'C' +

```



```

        " has the value " + ( int ) 'C' );
System.out.println( "The character " + 'a' +
        " has the value " + ( int ) 'a' );
System.out.println( "The character " + 'b' +
        " has the value " + ( int ) 'b' );
System.out.println( "The character " + 'c' +
        " has the value " + ( int ) 'c' );
System.out.println( "The character " + '0' +
        " has the value " + ( int ) '0' );
System.out.println( "The character " + '1' +
        " has the value " + ( int ) '1' );
System.out.println( "The character " + '2' +
        " has the value " + ( int ) '2' );
System.out.println( "The character " + '$' +
        " has the value " + ( int ) '$' );
System.out.println( "The character " + '*' +
        " has the value " + ( int ) '*' );
System.out.println( "The character " + '/' +
        " has the value " + ( int ) '/' );
System.out.println( "The character " + ' ' +
        " has the value " + ( int ) ' ' );
System.out.println( "The character " + ',' +
        " has the value " + ( int ) ',' );
    }
}

```