

JAVA™

AN INTRODUCTION TO
PROBLEM SOLVING
AND PROGRAMMING

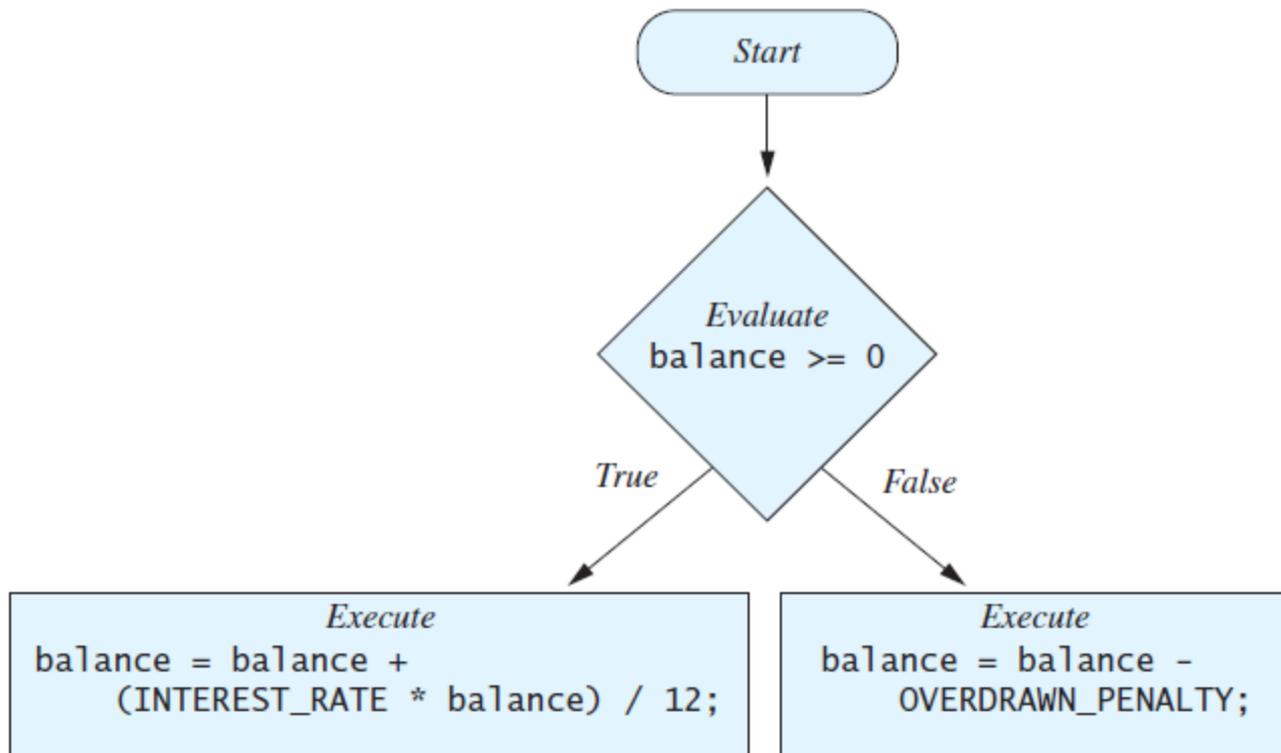
7TH EDITION

WALTER SAVITCH

Flow of Control: Branching 3

FIGURE 3.1 The Action of the `if-else` Statement in Listing 3.1

```
if (balance >= 0)
    balance = balance + (INTEREST_RATE * balance) / 12;
else
    balance = balance - OVERDRAWN_PENALTY;
```



LISTING 3.1 A Program Using if-else

```
import java.util.Scanner;
public class BankBalance
{
    public static final double OVERDRAWN_PENALTY = 8.00;
    public static final double INTEREST_RATE = 0.02; //2% annually

    public static void main(String[] args)
    {
        double balance;

        System.out.print("Enter your checking account
        balance: $");
        Scanner keyboard = new Scanner(System.in);
        balance = keyboard.nextDouble();
        System.out.println("Original balance $" + balance);

        if (balance >= 0)
            balance = balance + (INTEREST_RATE * balance)
            / 12;
        else
            balance = balance - OVERDRAWN_PENALTY;

        System.out.print("After adjusting for one month ");
        System.out.println("of interest and penalties,");
        System.out.println("your new balance is $" + balance);
    }
}
```

Sample Screen Output 1

```
Enter your checking account balance: $505.67
Original balance $505.67
After adjusting for one month of interest and penalties,
your new balance is $506.51278
```

Sample Screen Output 2

```
Enter your checking account balance: $-15.53
Original balance $ -15.53
After adjusting for one month of interest and penalties,
your new balance is $ -23.53
```

FIGURE 3.2 The Semantics of the if-else Statement

```
if (Boolean_Expression)  
    Statement_1  
else  
    Statement_2
```

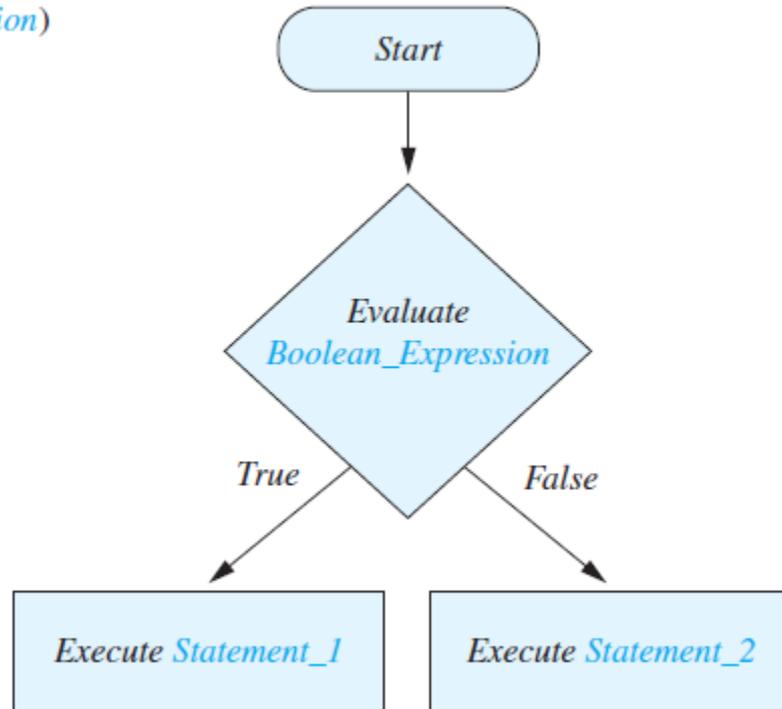


FIGURE 3.3 The Semantics of an if Statement Without an else

*if (Boolean_Expression)
Statement*

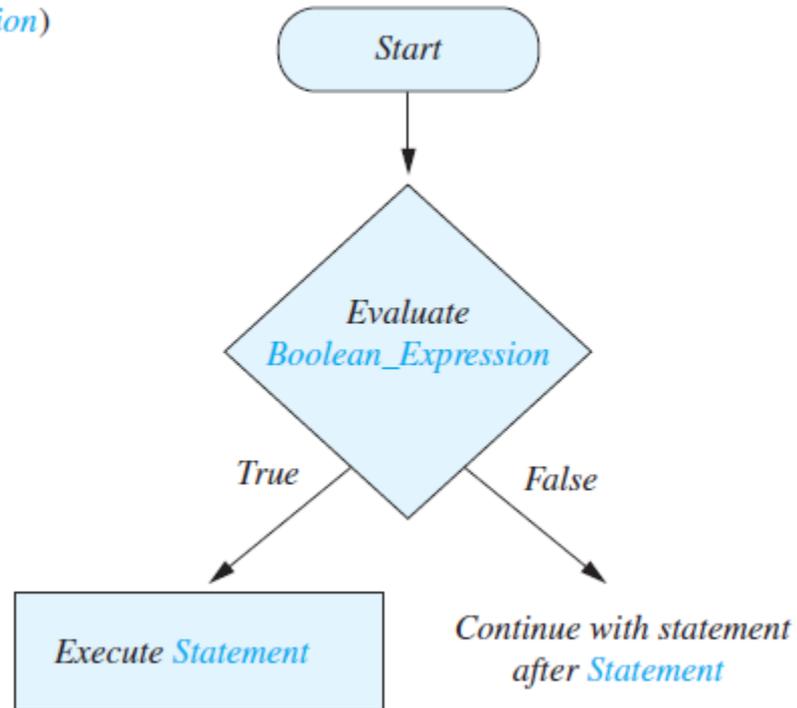


FIGURE 3.4 Java comparison operators

| Math Notation | Name | Java Notation | Java Examples |
|---------------|--------------------------|---------------|--------------------------------|
| = | Equal to | == | balance == 0 answer == 'y' |
| ≠ | Not equal to | != | income != tax answer != 'y' |
| > | Greater than | > | expenses > income |
| ≥ | Greater than or equal to | >= | points >= 60 |
| < | Less than | < | pressure < max |
| ≤ | Less than or equal to | <= | expenses <= income |

FIGURE 3.5 Avoiding the Negation Operator

| ! (A Op B) Is Equivalent to (A Op B) | |
|---|----|
| < | >= |
| <= | > |
| > | <= |
| >= | < |
| == | != |
| != | == |

FIGURE 3.6 Java Logical Operators

| Name | Java Notation | Java Examples |
|--------------------|---------------|------------------------------------|
| Logical <i>and</i> | && | (sum > min) && (sum < max) |
| Logical <i>or</i> | | (answer == 'y') (answer == 'Y') |
| Logical <i>not</i> | ! | !(number < 0) |

FIGURE 3.7 The Effect of the Boolean Operators `&&` (*and*), `||` (*or*), and `!` (*not*) on Boolean Values

| Value of <i>A</i> | Value of <i>B</i> | Value of <i>A</i> && <i>B</i> | Value of <i>A</i> <i>B</i> | Value of ! (<i>A</i>) |
|-------------------|-------------------|-------------------------------|-------------------------------|-------------------------|
| true | true | true | true | false |
| true | false | false | true | false |
| false | true | false | true | true |
| false | false | false | false | true |

LISTING 3.2 Testing Strings for Equality (part 1 of 2)

```
import java.util.Scanner;
public class StringEqualityDemo
{
    public static void main(String[] args)
    {
        String s1, s2;
        System.out.println("Enter two lines of text:");
        Scanner keyboard = new Scanner(System.in);
        s1 = keyboard.nextLine();
        s2 = keyboard.nextLine();

        if (s1.equals(s2))
            System.out.println("The two lines are equal.");
        else
            System.out.println("The two lines are not equal.");

        if (s2.equals(s1))
            System.out.println("The two lines are equal.");
        else
            System.out.println("The two lines are not equal.");

        if (s1.equalsIgnoreCase(s2))
            System.out.println(
                "But the lines are equal, ignoring case.");
        else
            System.out.println(
                "Lines are not equal, even ignoring case.");
    }
}
```

These two invocations of the method equals are equivalent.

Sample Screen Output

```
Enter two lines of text:
```

```
Java is not coffee.
```

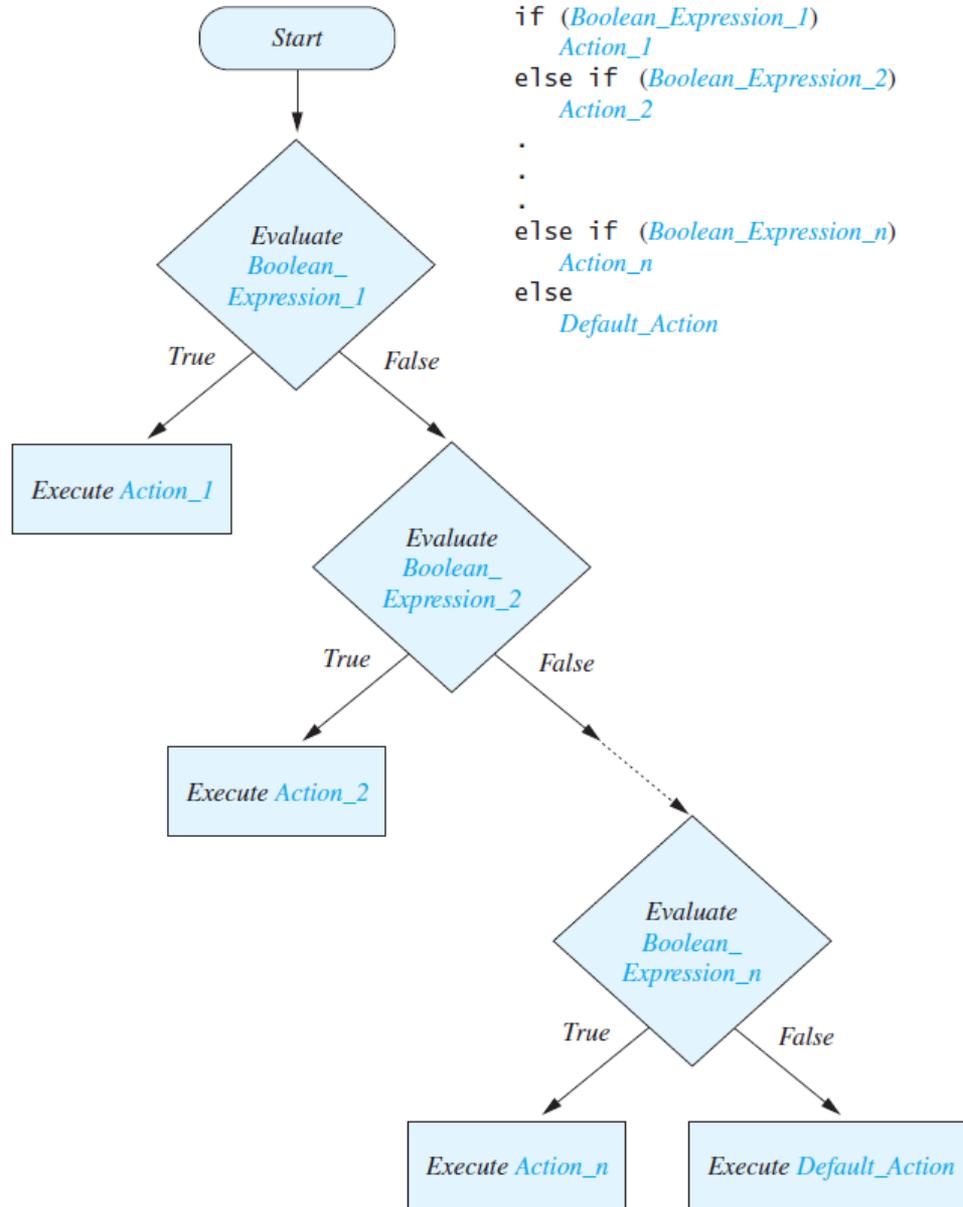
```
Java is NOT COFFEE.
```

```
The two lines are not equal.
```

```
The two lines are not equal.
```

```
But the lines are equal, ignoring case.
```

FIGURE 3.8 The Semantics of a Multibranch if-else Statement



LISTING 3.3 Assigning Letter Grades Using a Multibranch if-else Statement *(part 1 of 2)*

```
import java.util.Scanner;
public class Grader
{
    public static void main(String[] args)
    {
        int score;
        char grade;

        System.out.println("Enter your score: ");
        Scanner keyboard = new Scanner(System.in);
        score = keyboard.nextInt();

        if (score >= 90)
            grade = 'A';
        else if (score >= 80)
            grade = 'B';
        else if (score >= 70)
            grade = 'C';
        else if (score >= 60)
            grade = 'D';
        else
            grade = 'F';

        System.out.println("Score = " + score);
        System.out.println("Grade = " + grade);
    }
}
```

Sample Screen Output

```
Enter your score:
```

```
85
```

```
Score = 85
```

```
Grade = B
```

LISTING 3.4 A Body Mass Index Calculation Program (part 1 of 2)

```
import java.util.Scanner;
public class BMI
{
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner(System.in);
        int pounds, feet, inches;
        double heightMeters, mass, BMI;
        System.out.println("Enter your weight in pounds.");
        pounds = keyboard.nextInt();
        System.out.println("Enter your height in feet" +
            "followed by a space" +
            "then additional inches.");
        feet = keyboard.nextInt();
        inches = keyboard.nextInt();
        heightMeters = ((feet * 12) + inches) * 0.0254;
        mass = (pounds / 2.2);
        BMI = mass / (heightMeters * heightMeters);
        System.out.println("Your BMI is " + BMI);
        System.out.print("Your risk category is ");
        if (BMI < 18.5)
            System.out.println("Underweight.");
        else if (BMI < 25)
            System.out.println("Normal weight.");
    }
}
```

LISTING 3.4 A Body Mass Index Calculation Program *(part 2 of 2)*

```
        else if (BMI < 30)
            System.out.println("Overweight.");
        else
            System.out.println("Obese.");
    }
}
```

Sample Screen Output

```
Enter your weight in pounds.
150
Enter your height in feet followed
by a space then additional inches.
5 5
Your BMI is 25.013498117367398
Your risk category is Overweight.
```

FIGURE 3.9 Operator Precedence

| |
|--|
| <p><i>Highest Precedence</i></p> <p>First: the unary operators +, -, ++, --, and !</p> <p>Second: the binary arithmetic operators *, /, %</p> <p>Third: the binary arithmetic operators +, -</p> <p>Fourth: the boolean operators <, >, <=, >=</p> <p>Fifth: the boolean operators ==, !=</p> <p>Sixth: the boolean operator &</p> <p>Seventh: the boolean operator </p> <p>Eighth: the boolean operator &&</p> <p>Ninth: the boolean operator </p> <p><i>Lowest Precedence</i></p> |
|--|

LISTING 3.5 A switch Statement (part 1 of 2)

```
import java.util.Scanner;
public class MultipleBirths
{
    public static void main(String[] args)
    {
        int numberOfBabies;
        System.out.print("Enter number of babies: ");
        Scanner keyboard = new Scanner(System.in);
        numberOfBabies = keyboard.nextInt();

        switch (numberOfBabies) ← Controlling expression
        {
            case 1: ← Case label
                System.out.println("Congratulations.");
                break;
            case 2:
                System.out.println("Wow. Twins.");
                break; ← break statement
            case 3:
                System.out.println("Wow. Triplets.");
                break;
            case 4: ← Case with nobreak
            case 5:
                System.out.print("Unbelievable; ");
                System.out.println(numberOfBabies +
                    " babies.");

                break;
            default:
                System.out.println("I don't believe you.");
                break;
        }
    }
}
```

Sample Screen Output 1

```
Enter number of babies: 1  
Congratulations.
```

Sample Screen Output 2

```
Enter number of babies: 3  
Wow. Triplets.
```

Sample Screen Output 3

```
Enter number of babies: 4  
Unbelievable; 4 babies.
```

Sample Screen Output 4

```
Enter number of babies: 6  
I don't believe you.
```

LISTING 3.6 Adding Color (part 1 of 2)

```
import javax.swing.JApplet;
import java.awt.Color;
import java.awt.Graphics;

public class YellowFace extends JApplet
{
    public static final int FACE_DIAMETER = 200;
    public static final int X_FACE = 100;
    public static final int Y_FACE = 50;
    public static final int EYE_WIDTH = 10;
    public static final int EYE_HEIGHT = 20;
    public static final int X_RIGHT_EYE = 155;
    public static final int Y_RIGHT_EYE = 100;
    public static final int X_LEFT_EYE = 230;
    public static final int Y_LEFT_EYE = Y_RIGHT_EYE;
    public static final int NOSE_DIAMETER = 10;
    public static final int X_NOSE = 195; //Center of nose will
                                         //be at 200

    public static final int Y_NOSE = 135;
    public static final int MOUTH_WIDTH = 100;
    public static final int MOUTH_HEIGHT = 50;
```

```

public static final int X_MOUTH = 150;
public static final int Y_MOUTH = 160;
public static final int MOUTH_START_ANGLE = 180;
public static final int MOUTH_EXTENT_ANGLE = 180;
public void paint(Graphics canvas)
{
    super.paint(canvas);
    //Draw face interior and outline:
    canvas.setColor(Color.YELLOW);
    canvas.fillOval(X_FACE, Y_FACE, FACE_DIAMETER,
    FACE_DIAMETER);
    canvas.setColor(Color.BLACK);
    canvas.drawOval(X_FACE, Y_FACE, FACE_DIAMETER,
    FACE_DIAMETER);
    //Draw eyes:
    canvas.setColor(Color.BLUE);
    canvas.fillOval(X_RIGHT_EYE, Y_RIGHT_EYE, EYE_WIDTH,
    EYE_HEIGHT);
    canvas.fillOval(X_LEFT_EYE, Y_LEFT_EYE, EYE_WIDTH,
    EYE_HEIGHT);
    //Draw nose:
    canvas.setColor(Color.BLACK);
    canvas.fillOval(X_NOSE, Y_NOSE, NOSE_DIAMETER,
    NOSE_DIAMETER);
    //Draw mouth:
    canvas.setColor(Color.RED);
    canvas.drawArc(X_MOUTH, Y_MOUTH, MOUTH_WIDTH, MOUTH_HEIGHT,
    MOUTH_START_ANGLE, MOUTH_EXTENT_ANGLE);
}
}

```

The filled yellow circle is drawn first so that the other drawings will be on top of the yellow.

FIGURE 3.10 Predefined Colors for the `setColor` Method

| | |
|-------------------------------|----------------------------|
| <code>Color.BLACK</code> | <code>Color.MAGENTA</code> |
| <code>Color.BLUE</code> | <code>Color.ORANGE</code> |
| <code>Color.CYAN</code> | <code>Color.PINK</code> |
| <code>Color.DARK_GRAY</code> | <code>Color.RED</code> |
| <code>Color.GRAY</code> | <code>Color.WHITE</code> |
| <code>Color.GREEN</code> | <code>Color.YELLOW</code> |
| <code>Color.LIGHT_GRAY</code> | |

FIGURE 3.11 A Yes-or-No Dialog Box

