

Chapter 3 Selections

The boolean Type and Operators

2

```
boolean b = (1 > 2);
```

Problem: A Simple Math Learning Tool

This example creates a program to let a first grader practice additions. The program randomly generates two single-digit integers number1 and number2 and displays a question such as “What is $7 + 9$?” to the student. After the student types the answer, the program displays a message to indicate whether the answer is true or false.

```
import java.util.Scanner;

public class AdditionQuiz {
    public static void main(String[] args) {
        int number1 = (int)(System.currentTimeMillis() % 10);
        int number2 = (int)(System.currentTimeMillis() / 7 % 10);

        // Create a Scanner
        Scanner input = new Scanner(System.in);

        System.out.print(
            "What is " + number1 + " + " + number2 + "? ");

        int answer = input.nextInt();

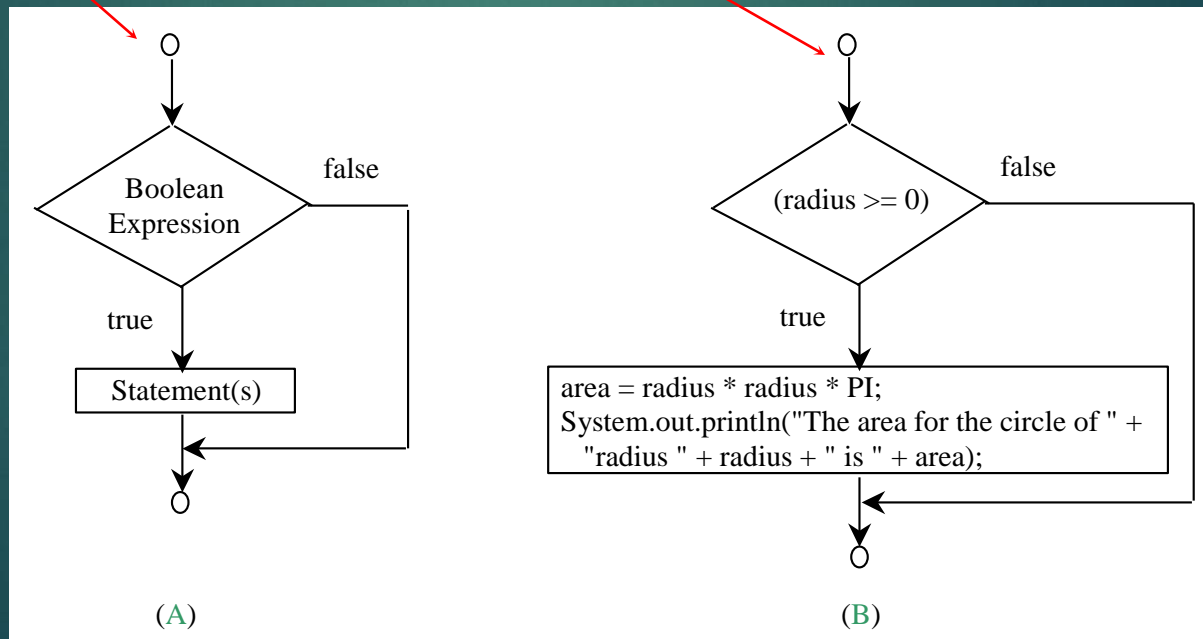
        System.out.println(
            number1 + " + " + number2 + " = " + answer + " is " +
            (number1 + number2 == answer));
    }
}
```

One-way if Statements

5

```
if (boolean-expression) {  
    statement(s);  
}
```

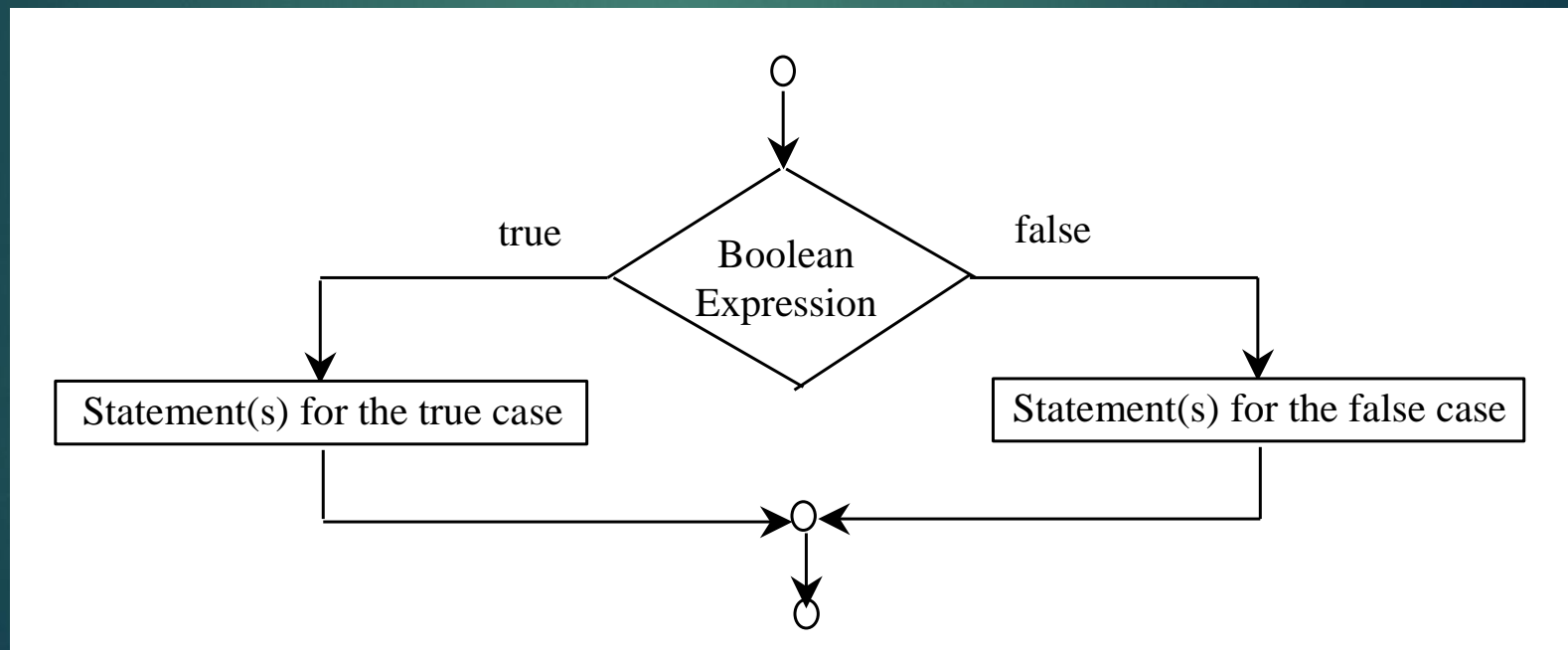
```
if (radius >= 0) {  
    area = radius * radius * PI;  
    System.out.println("The area"  
        + " for the circle of radius "  
        + radius + " is " + area);  
}
```



The Two-way if Statement

6

```
if (boolean-expression) {  
    statement(s)-for-the-true-case;  
}  
else {  
    statement(s)-for-the-false-case;  
}
```



if...else Example

7

```
if (radius >= 0) {
    area = radius * radius * 3.14159;

    System.out.println("The area for the "
        + "circle of radius " + radius +
        " is " + area);
}
else {
    System.out.println("Negative input");
}
```

Multiple Alternative if Statements

8

```
if (score >= 90.0)
    grade = 'A';
else
    if (score >= 80.0)
        grade = 'B';
    else
        if (score >= 70.0)
            grade = 'C';
        else
            if (score >= 60.0)
                grade = 'D';
            else
                grade = 'F';
```

Equivalent

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```


Trace if-else statement

9

Suppose score is 70.0

The condition is false

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

trace if-else statement

10

Suppose score is 70.0

The condition is false

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

trace if-else statement

11

Suppose score is 70.0

The condition is true

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

trace if-else statement

12

Suppose score is 70.0

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

grade is C

trace if-else statement

13

Suppose score is 70.0

Exit the if statement

```
if (score >= 90.0)
    grade = 'A';
else if (score >= 80.0)
    grade = 'B';
else if (score >= 70.0)
    grade = 'C';
else if (score >= 60.0)
    grade = 'D';
else
    grade = 'F';
```

Note

14

The else clause matches the most recent if clause in the same block.

```
int i = 1;
int j = 2;
int k = 3;

if (i > j)
    if (i > k)
        System.out.println("A");
else
    System.out.println("B");
```

(a)

Equivalent

```
int i = 1;
int j = 2;
int k = 3;

if (i > j)
    if (i > k)
        System.out.println("A");
else
    System.out.println("B");
```

(b)

Note, cont.

15

Nothing is printed from the preceding statement. To force the else clause to match the first if clause, you must add a pair of braces:

```
int i = 1;
int j = 2;
int k = 3;
if (i > j) {
    if (i > k)
        System.out.println("A");
}
else
    System.out.println("B");
```

This statement prints B.

Common Errors

16

Adding a semicolon at the end of an if clause is a common mistake.

```
if (radius >= 0); ← Wrong
{
    area = radius*radius*PI;
    System.out.println(
        "The area for the circle of radius " +
        radius + " is " + area);
}
```

This mistake is hard to find, because it is not a compilation error or a runtime error, it is a logic error.

This error often occurs when you use the next-line block style.

TIP

17

```
if (number % 2 == 0)
    even = true;
else
    even = false;
```

(a)

Equivalent

```
boolean even
    = number % 2 == 0;
```

(b)

CAUTION

18

```
if (even == true)
    System.out.println(
        "It is even.");
```

(a)

Equivalent

```
if (even)
    System.out.println(
        "It is even.");
```

(b)

Problem: An Improved Math Learning Tool

This example creates a program to teach a first grade child how to learn subtractions. The program randomly generates two single-digit integers number1 and number2 with number1 > number2 and displays a question such as "What is 9 - 2?" to the student. After the student types the answer in the input dialog box, the program displays a message dialog box to indicate whether the answer is correct.

```
import java.util.Scanner;

public class SubtractionQuiz {
    public static void main(String[] args) {
        // 1. Generate two random single-digit integers
        int number1 = (int)(Math.random() * 10);
        int number2 = (int)(Math.random() * 10);

        // 2. If number1 < number2, swap number1 with number2
        if (number1 < number2) {
            int temp = number1;
            number1 = number2;
            number2 = temp;
        }

        // 3. Prompt the student to answer "what is number1 - number2?"
        System.out.print ("What is " + number1 + " - " + number2 + "? ");
        Scanner input = new Scanner(System.in);
        int answer = input.nextInt();

        // 4. Grade the answer and display the result
        if (number1 - number2 == answer)
            System.out.println("You are correct!");
        else
            System.out.println("Your answer is wrong.\n" + number1 + " - "
                + number2 + " should be " + (number1 - number2));
    }
}
```

Problem: Body Mass Index

21

Body Mass Index (BMI) is a measure of health on weight. It can be calculated by taking your weight in kilograms and dividing by the square of your height in meters. The interpretation of BMI for people 16 years or older is as follows:

BMI	Interpretation
below 16	serious underweight
16-18	underweight
18-24	normal weight
24-29	overweight
29-35	seriously overweight
above 35	gravely overweight

```
import java.util.Scanner;
```

22

```
public class ComputeBMI {  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
  
        // Prompt the user to enter weight in pounds  
        System.out.print("Enter weight in pounds: ");  
        double weight = input.nextDouble();  
  
        // Prompt the user to enter height in inches  
        System.out.print("Enter height in inches: ");  
        double height = input.nextDouble();  
  
        final double KILOGRAMS_PER_POUND = 0.45359237; // Constant  
        final double METERS_PER_INCH = 0.0254; // Constant  
  
        // Compute BMI  
        double weightInKilogram = weight * KILOGRAMS_PER_POUND;  
        double heightInMeters = height * METERS_PER_INCH;  
        double bmi = weightInKilogram /  
            (heightInMeters * heightInMeters);
```



```
// Display result
System.out.printf("Your BMI is %5.2f\n", bmi);
if (bmi < 16)
    System.out.println("You are seriously underweight");
else if (bmi < 18)
    System.out.println("You are underweight");
else if (bmi < 24)
    System.out.println("You are normal weight");
else if (bmi < 29)
    System.out.println("You are overweight");
else if (bmi < 35)
    System.out.println("You are seriously overweight");
else
    System.out.println("You are gravely overweight");
}
}
```

Logical Operators

24

<i>Operator</i>	<i>Name</i>
!	not
& &	and
	or
^	exclusive or

Examples

25

Here is a program that checks whether a number is divisible by 2 and 3, whether a number is divisible by 2 or 3, and whether a number is divisible by 2 or 3 but not both:

```
System.out.println("Is " + number + " divisible by 2 and 3? " +  
((number % 2 == 0) && (number % 3 == 0)));
```

```
System.out.println("Is " + number + " divisible by 2 or 3? " +  
((number % 2 == 0) || (number % 3 == 0)));
```

```
System.out.println("Is " + number +  
" divisible by 2 or 3, but not both? " +  
((number % 2 == 0) ^ (number % 3 == 0)));
```

Examples

Problem: Determining Leap Year? 27

This program first prompts the user to enter a year as an int value and checks if it is a leap year.

A year is a leap year if it **is divisible by 4** but **not by 100**, or it is divisible by 400.

```
(year % 4 == 0 && year % 100 != 0)
```

```
|| (year % 400 == 0)
```

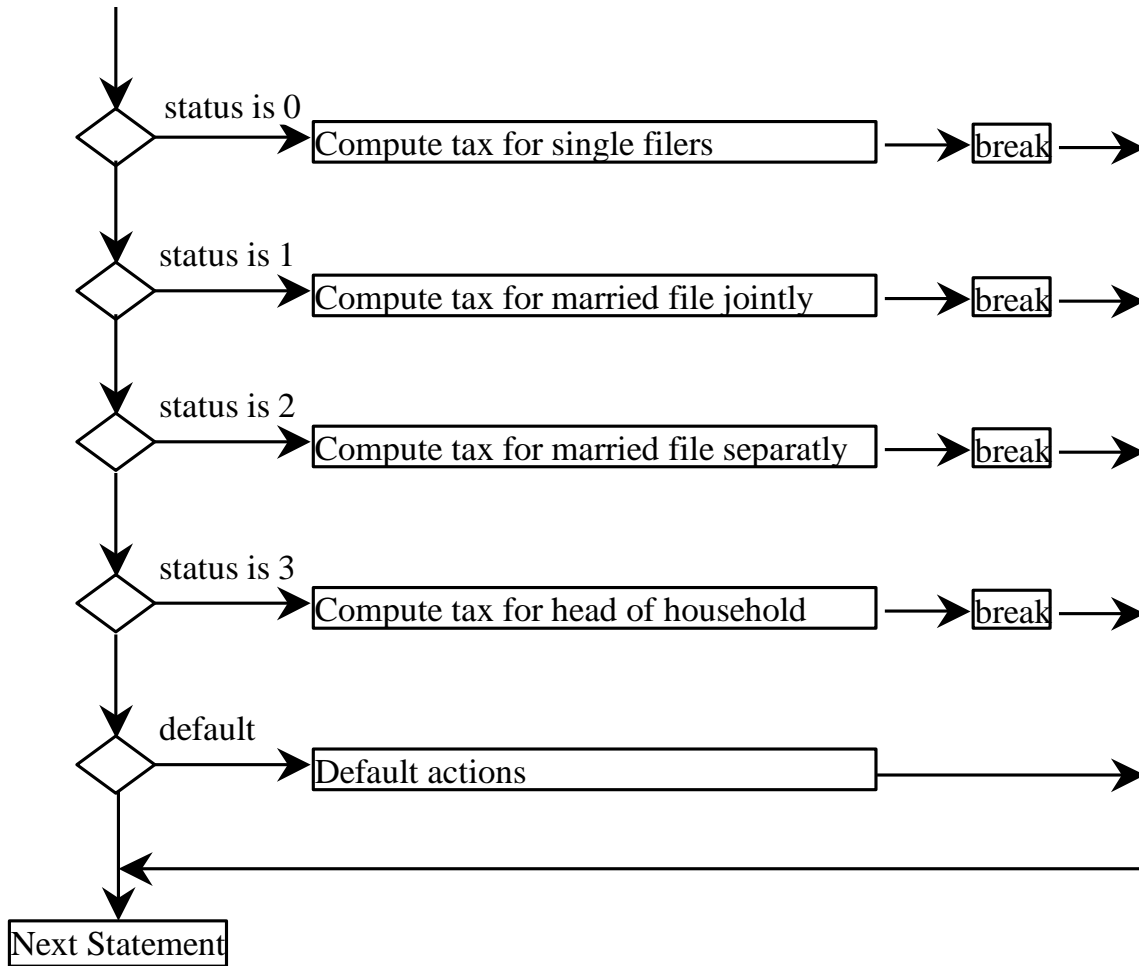
switch Statements

28

```
switch (status) {  
    case 0: compute taxes for single filers;  
            break;  
    case 1: compute taxes for married file jointly;  
            break;  
    case 2: compute taxes for married file separately;  
            break;  
    case 3: compute taxes for head of household;  
            break;  
    default: System.out.println("Errors: invalid status");  
            System.exit(0);  
}
```

switch Statement Flow Chart

29



Trace switch statement

30

Suppose ch is 'a':

```
switch (ch) {  
  case 'a': System.out.println(ch);  
  case 'b': System.out.println(ch);  
  case 'c': System.out.println(ch);  
}
```

Trace switch statement

31



ch is 'a':

```
switch (ch) {  
  case 'a': System.out.println(ch);  
  case 'b': System.out.println(ch);  
  case 'c': System.out.println(ch);  
}
```

Trace switch statement

32



Execute this line

```
switch (ch) {  
  case 'a': System.out.println(ch);  
  case 'b': System.out.println(ch);  
  case 'c': System.out.println(ch);  
}
```


Trace switch statement

Execute this line

```
switch (ch) {  
  case 'a': System.out.println(ch);  
  case 'b': System.out.println(ch);  
  case 'c': System.out.println(ch);  
}
```

Trace switch statement

34



Execute this line

```
switch (ch) {  
  case 'a': System.out.println(ch);  
  case 'b': System.out.println(ch);  
  case 'c': System.out.println(ch);  
}
```

Trace switch statement

35



Execute next statement

```
switch (ch)
  case 'a': System.out.println(ch);
  case 'b': System.out.println(ch);
  case ' ': System.out.println(ch);
}
```

Next statement;

Trace switch statement

36

Suppose ch is 'a':

```
switch (ch) {  
  case 'a': System.out.println(ch);  
             break;  
  case 'b': System.out.println(ch);  
             break;  
  case 'c': System.out.println(ch);  
}
```

Trace switch statement

37



ch is 'a':

```
switch (ch) {  
  case 'a': System.out.println(ch);  
             break;  
  case 'b': System.out.println(ch);  
             break;  
  case 'c': System.out.println(ch);  
}
```

Trace switch statement

38



Execute this line

```
switch (ch) {  
  case 'a': System.out.println(ch);  
            break;  
  case 'b': System.out.println(ch);  
            break;  
  case 'c': System.out.println(ch);  
}
```

Trace switch statement

39



Execute this line

```
switch (ch) {  
  case 'a': System.out.println(ch);  
            break;  
  case 'b': System.out.println(ch);  
            break;  
  case 'c': System.out.println(ch);  
}
```

Trace switch statement

40



Execute next statement

```
switch (ch)
  case 'a': System.out.println(ch);
            break;
  case 'b': System.out.println(ch);
            break;
  case 'c': System.out.println(ch);
}
}
```

Next statement;

Conditional Operator

41

```
if (x > 0)
```

```
    y = 1
```

```
else
```

```
    y = -1;
```

is equivalent to

```
y = (x > 0) ? 1 : -1;
```

```
(boolean-expression) ? expression1 : expression2
```

Ternary operator

Binary operator

Unary operator

Conditional Operator

42

```
if (num % 2 == 0)
    System.out.println(num + "is even");
else
    System.out.println(num + "is odd");
```

```
System.out.println(
    (num % 2 == 0)? num + "is even" :
    num + "is odd");
```

Conditional Operator, cont.

43

```
(boolean-expression) ? exp1 : exp2
```

Formatting Output

44

Use the printf statement.

```
System.out.printf(format, items);
```

Where format is a string that may consist of substrings and format specifiers. A format specifier specifies how an item should be displayed. An item may be a numeric value, character, boolean value, or a string. Each specifier begins with a percent sign.

Frequently-Used Specifiers 45

Specifier	Output	Example
<u>%b</u>	a boolean value	true or false
<u>%c</u>	a character	'a'
<u>%d</u>	a decimal integer	200
<u>%f</u>	a floating-point number	45.460000
<u>%e</u>	a number in standard scientific notation	4.556000e+01
<u>%s</u>	a string	"Java is cool"

```
int count = 5;
double amount = 45.56;
System.out.printf("count is %d and amount is %f", count, amount);
```

display count is 5 and amount is 45.560000

Operator Precedence

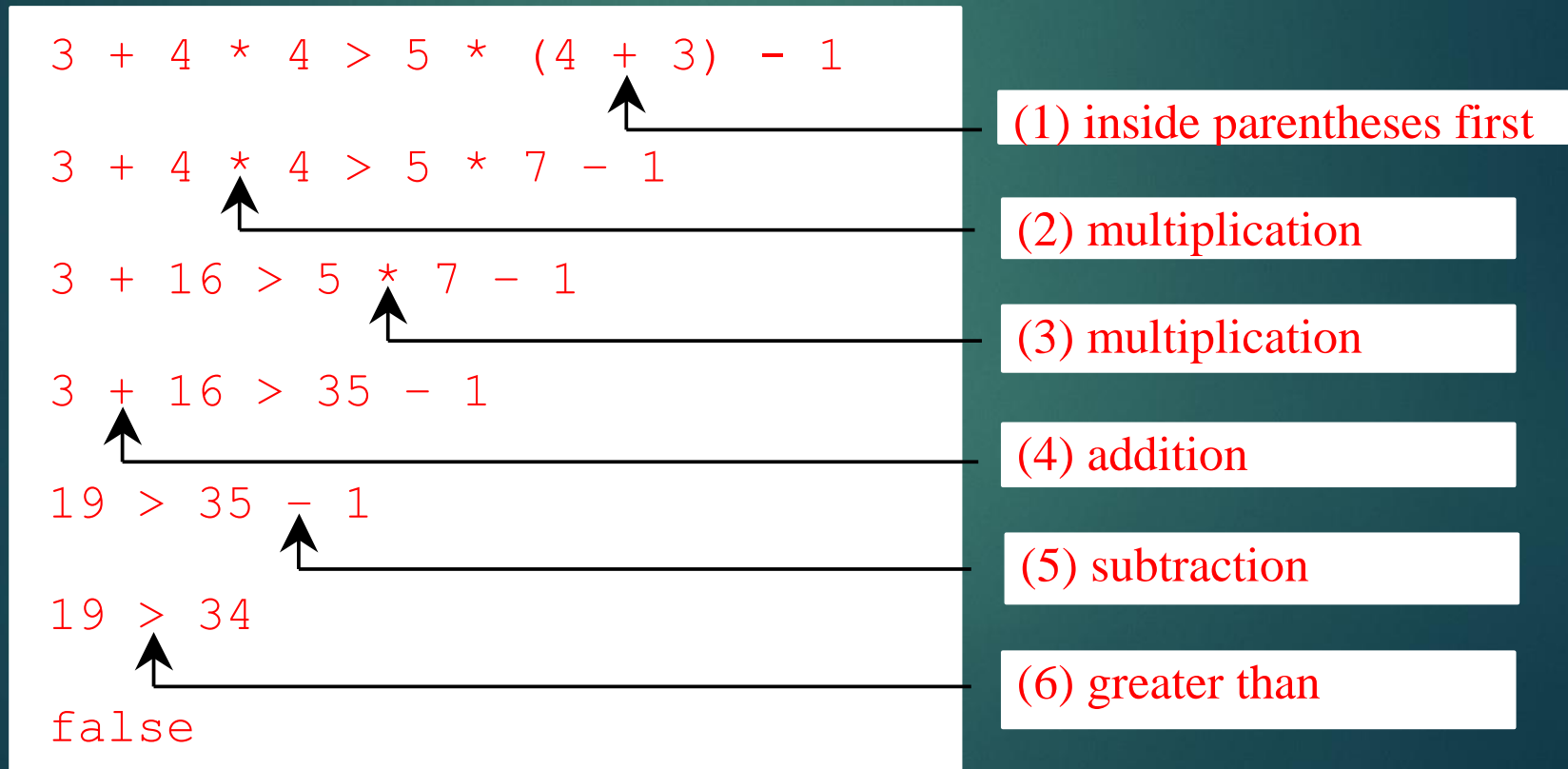
46

- ▶ `var++`, `var--`
- ▶ `+`, `-` (Unary plus and minus), `++var`, `--var`
- ▶ `(type)` Casting
- ▶ `!` (Not)
- ▶ `*`, `/`, `%` (Multiplication, division, and remainder)
- ▶ `+`, `-` (Binary addition and subtraction)
- ▶ `<`, `<=`, `>`, `>=` (Comparison)
- ▶ `==`, `!=`; (Equality)
- ▶ `^` (Exclusive OR)
- ▶ `&&` (Conditional AND) Short-circuit AND
- ▶ `||` (Conditional OR) Short-circuit OR
- ▶ `=`, `+=`, `-=`, `*=`, `/=`, `%=` (Assignment operator)

Example

47

Applying the operator precedence and associativity rule, the expression $3 + 4 * 4 > 5 * (4 + 3) - 1$ is evaluated as follows:



Chapter 4 Loops

Introducing while Loops

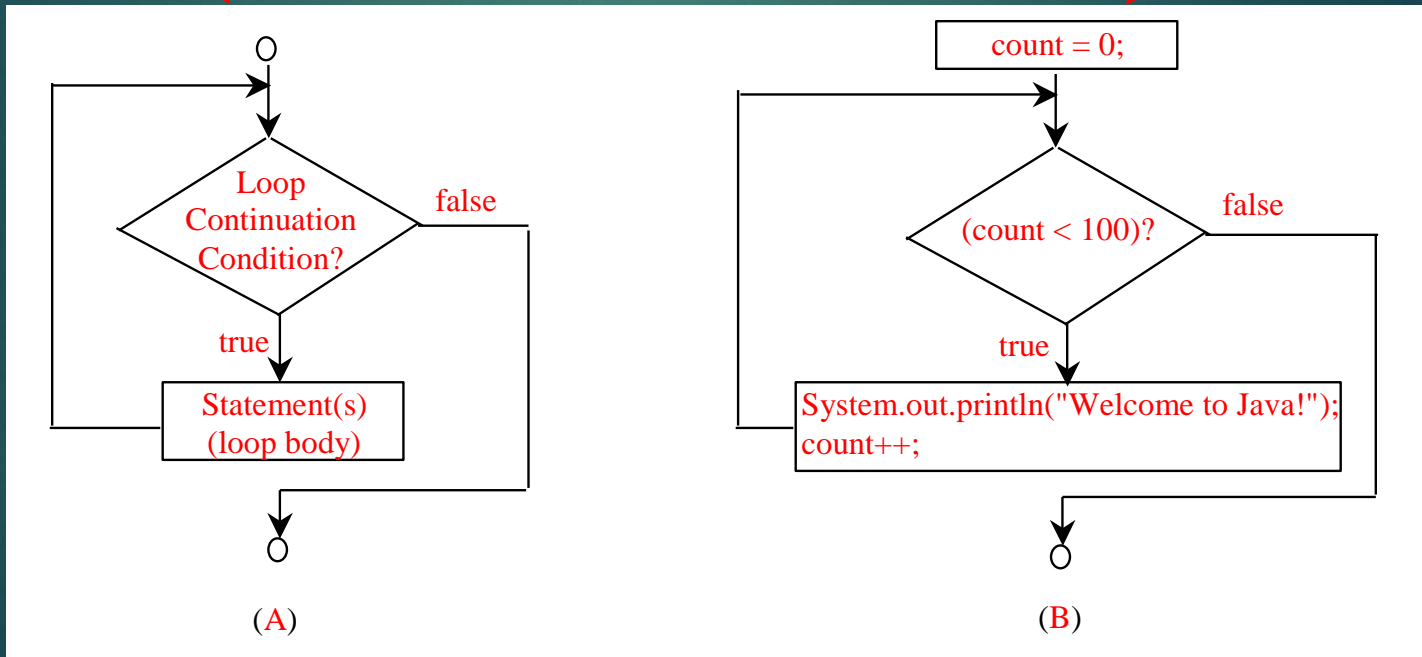
```
int count = 0;  
while (count < 100) {  
    System.out.println("Welcome to Java");  
    count++;  
}
```

while Loop Flow Chart

```
while (loop-continuation-condition) {  
    // loop-body;  
    Statement(s);  
}
```

```
int count = 0;  
while (count < 100) {  
    System.out.println("Welcome to Java!");  
    count++;  
}
```

50



Trace while Loop

```
int count = 0;
```

```
while (count < 2) {
```

```
    System.out.println("Welcome to Java!");
```

```
    count++;
```

```
}
```

Initialize count

Trace while Loop, cont.

```
int count = 0;  
while (count < 2) {  
    System.out.println("Welcome to Java!");  
    count++;  
}
```

(count < 2) is true

Trace while Loop, cont.

```
int count = 0;
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}
```

Print Welcome to Java

Trace while Loop, cont.

```
int count = 0;  
while (count < 2) {  
    System.out.println("Welcome to Java!");  
    count++;  
}
```

Increase count by 1
count is 1 now

Trace while Loop, cont.

```
int count = 0;
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}
```

(count < 2) is still true since count is 1

Trace while Loop, cont.

```
int count = 0;  
while (count < 2) {  
    System.out.println("Welcome to Java!");  
    count++;  
}
```



Print Welcome to Java

Trace while Loop, cont.

```
int count = 0;
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}
```

Increase count by 1
count is 2 now

Trace while Loop, cont.

```
int count = 0;  
while (count < 2) {  
    System.out.println("Welcome to Java!");  
    count++;  
}
```

(count < 2) is false since count is 2
now

Trace while Loop

```
int count = 0;
while (count < 2) {
    System.out.println("Welcome to Java!");
    count++;
}
```

The loop exits. Execute the next statement after the loop.

Problem: Guessing Numbers

Write a program that randomly generates an integer between 0 and 100, inclusive. The program prompts the user to enter a number continuously until the number matches the randomly generated number. For each user input, the program tells the user whether the input is too low or too high, so the user can choose the next input intelligently. Here is a sample run:

```
import java.util.Scanner;

public class GuessNumber {
    public static void main(String[] args) {
        // Generate a random number to be guessed
        int number = (int)(Math.random() * 101);

        Scanner input = new Scanner(System.in);
        System.out.println("Guess a magic number between 0 and 100");

        int guess = -1;
        while (guess != number) {
            // Prompt the user to guess the number
            System.out.print("\nEnter your guess: ");
            guess = input.nextInt();

            if (guess == number)
                System.out.println("Yes, the number is " + number);
            else if (guess > number)
                System.out.println("Your guess is too high");
            else
                System.out.println("Your guess is too low");
        } // End of loop
    }
}
```

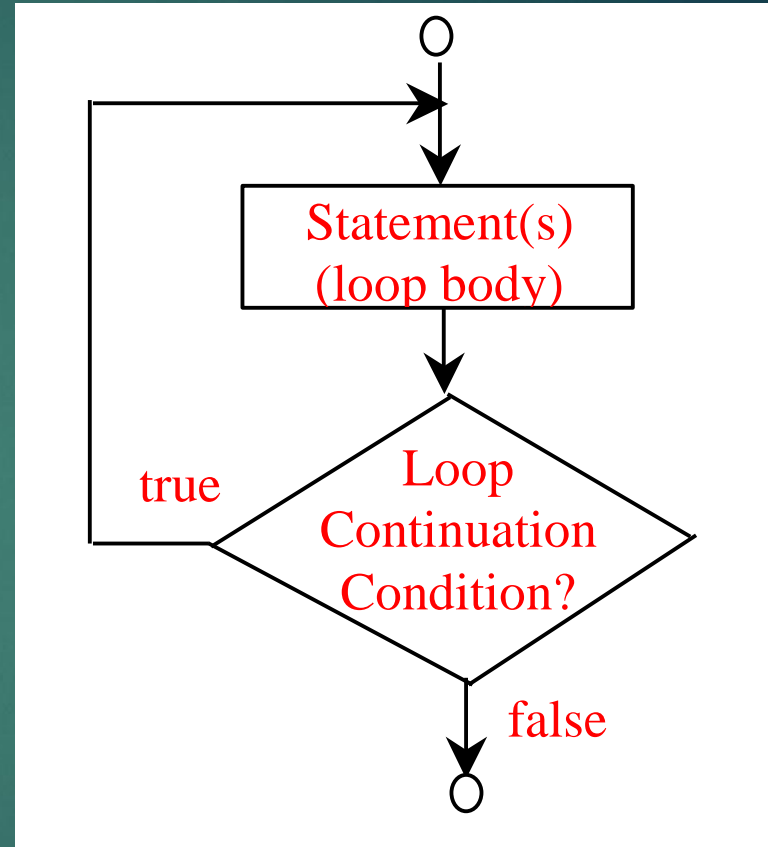
Ending a Loop with a Sentinel Value

Often the number of times a loop is executed is not predetermined. You may use an input value to signify the end of the loop. Such a value is known as a *sentinel value*.

Write a program that reads and calculates the sum of an unspecified number of integers. The input 0 signifies the end of the input.

do-while Loop

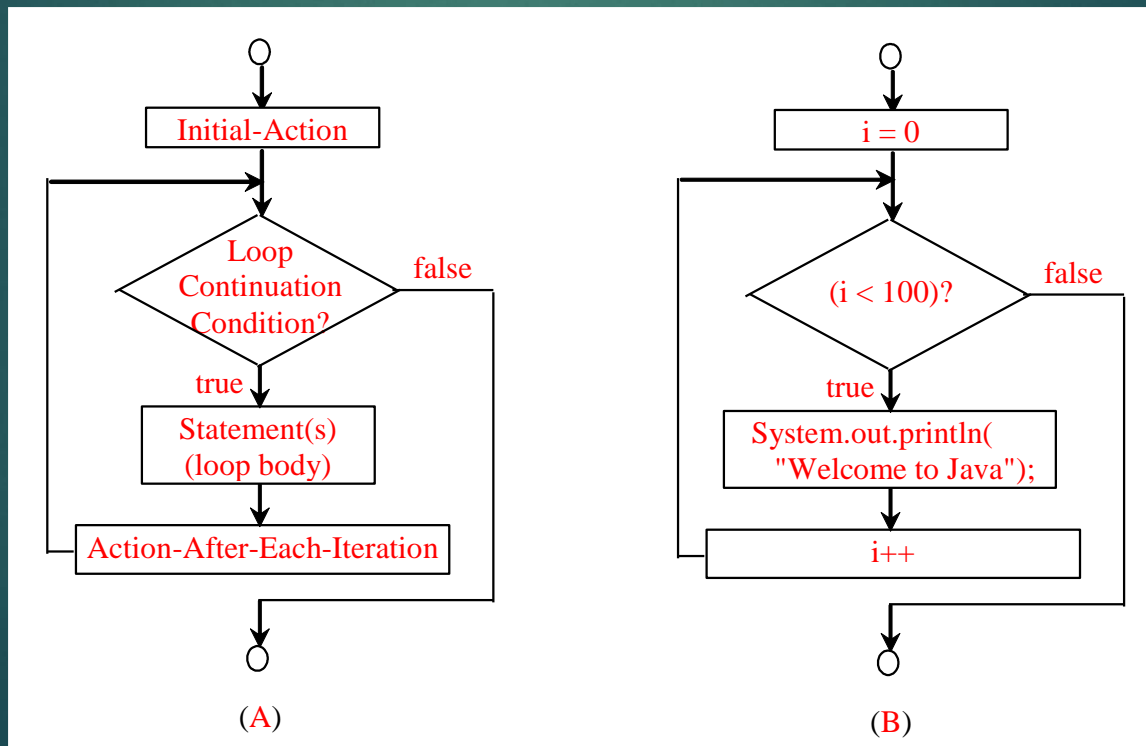
```
do {  
    // Loop body;  
    Statement(s);  
} while (loop-continuation-condition);
```



for Loops

```
for (initial-action; loop-  
continuation-condition;  
action-after-each-  
iteration) {  
  // loop body;  
  Statement(s);  
}
```

```
int i;  
for (i = 0; i < 100; i++) {  
  System.out.println(  
    "Welcome to Java!");  
}
```



Trace for Loop

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println(  
        "Welcome to Java!");  
}
```

Declare i

Trace for Loop, cont.

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println(  
        "Welcome to Java!");  
}
```

Execute initializer
i is now 0

Trace for Loop, cont.

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println( "Welcome to Java!");  
}
```

(i < 2) is true
since i is 0

Trace for Loop, cont.

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Welcome to Java!");  
}
```

Print Welcome to Java

Trace for Loop, cont.

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Welcome to Java!");  
}
```

Execute adjustment statement
i now is 1

Trace for Loop, cont.

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Welcome to Java!");  
}
```

($i < 2$) is still true
since i is 1

Trace for Loop, cont.

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Welcome to Java!");  
}
```

Print Welcome to Java

Trace for Loop, cont.

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Welcome to Java!");  
}
```

Execute adjustment statement
i now is 2

Trace for Loop, cont.

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Welcome to Java!");  
}
```

($i < 2$) is false
since i is 2

Trace for Loop, cont.

```
int i;  
for (i = 0; i < 2; i++) {  
    System.out.println("Welcome to Java");  
}
```

Exit the loop. Execute the next statement after the loop

Nested Loops

Problem: Write a program that uses nested for loops to print a multiplication table.

Problem: Predicating the Future Tuition

Problem: Suppose that the tuition for a university is \$10,000 this year and tuition increases 7% every year. In how many years will the tuition be doubled?

```
public class FutureTuition {
    public static void main(String[] args) {
        double tuition = 10000; // Year 1
        int year = 1;
        while (tuition < 20000) {
            tuition = tuition * 1.07;
            year++;
        }

        System.out.println("Tuition will be doubled in "
            + year + " years");
    }
}
```

Guessing Number Problem Revisited

Here is a program for guessing a number. You can rewrite it using a break statement.⁷⁸

GuessNumberUsingBreak

Run