

Selections



Liang, Introduction to Java programming, 11th Edition, © 2017 Pearson Education, Inc.
All rights reserved



By: Mamoun Nawahdah (Ph.D.)
2022

The boolean Type and Operators

- ❖ Often in a program you need to compare two values, such as whether *i* is greater than *j*.
- ❖ Java provides six comparison operators (also known as relational operators) that can be used to compare two values.
- ❖ The result of the comparison is a **Boolean** value: **true** or **false**.

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



Comparison Operators

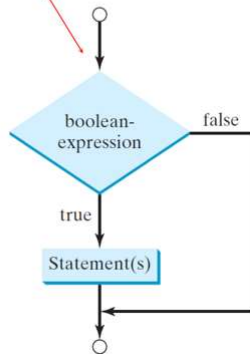
Java Operator	Mathematics Symbol	Name	Example (radius is 5)	Result
<	<	less than	<code>radius < 0</code>	<code>false</code>
<=	≤	less than or equal to	<code>radius <= 0</code>	<code>false</code>
>	>	greater than	<code>radius > 0</code>	<code>true</code>
>=	≥	greater than or equal to	<code>radius >= 0</code>	<code>true</code>
==	=	equal to	<code>radius == 0</code>	<code>false</code>
!=	≠	not equal to	<code>radius != 0</code>	<code>true</code>



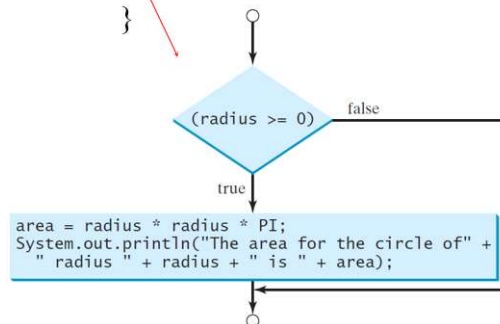
3

One-way if Statements

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

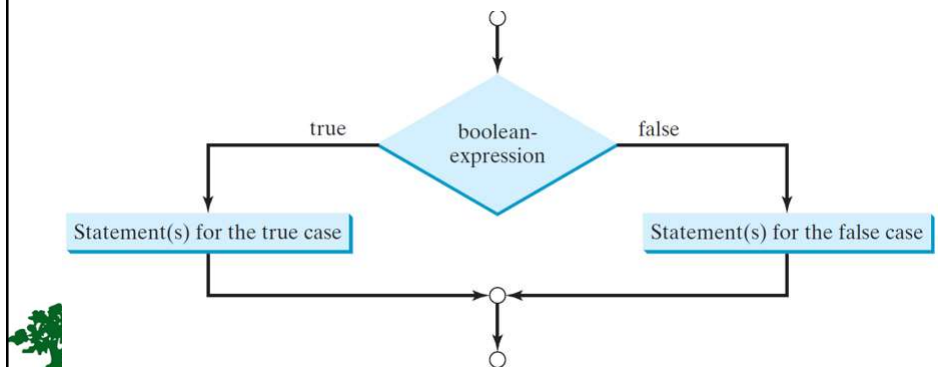


```
if (radius >= 0) {
    area = radius * radius * PI;
    System.out.println("The area "
        + " for the circle of radius "
        + radius + " is " + area);
}
```



The Two-way **if** Statement

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



if-else example

```
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("Error: Negative input");  
}
```

Multiple Alternative if Statements

```

if (score >= 90.0)
    System.out.print("A");
else
    if (score >= 80.0)
        System.out.print("B");
    else
        if (score >= 70.0)
            System.out.print("C");
        else
            if (score >= 60.0)
                System.out.print("D");
            else
                System.out.print("F");
    
```

(a)

Equivalent

```

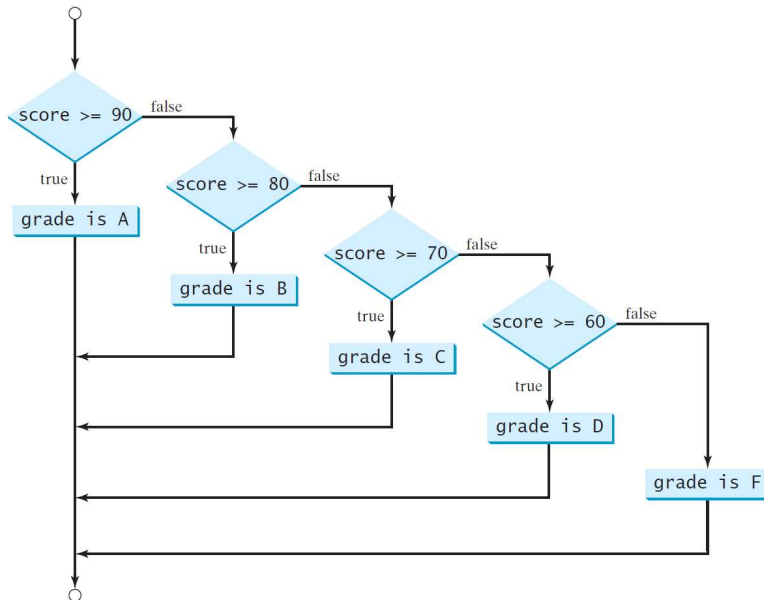
if (score >= 90.0)
    System.out.print("A");
else if (score >= 80.0)
    System.out.print("B");
else if (score >= 70.0)
    System.out.print("C");
else if (score >= 60.0)
    System.out.print("D");
else
    System.out.print("F");
    
```

This is better

(b)



Multi-Way if-else Statements



Note

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

<pre>int i = 1, j = 2, k = 3; if (i > j) if (i > k) System.out.println("A"); else System.out.println("B");</pre>	<p>Equivalent</p> <hr style="width: 50%; margin: 0 auto;"/> <p>This is better with correct indentation</p>	<pre>int i = 1, j = 2, k = 3; if (i > j) if (i > k) System.out.println("A"); else System.out.println("B");</pre>
(a)		(b)

<pre>if (even == true) System.out.println("It is even.");</pre>	<p>Equivalent</p> <hr style="width: 50%; margin: 0 auto;"/>	<pre>if (even) System.out.println("It is even.");</pre>
(a)		(b)



9

Common Errors

❖ 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 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.



10

Logical Operators

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



11

switch Statements

```
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(1);  
}
```



12

Trace **switch** statement

```

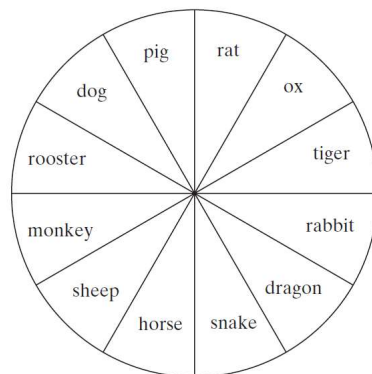
switch (day) {
  case 1:
  case 2:
  case 3:
  case 4:
  case 5: System.out.println("Weekday"); break;
  case 0:
  case 6: System.out.println("Weekend");
}

```



Problem: Chinese Zodiac

Write a program that prompts the user to enter a year and displays the animal for the year.



$\text{year \% 12} = \left\{ \begin{array}{l} 0: \text{monkey} \\ 1: \text{rooster} \\ 2: \text{dog} \\ 3: \text{pig} \\ 4: \text{rat} \\ 5: \text{ox} \\ 6: \text{tiger} \\ 7: \text{rabbit} \\ 8: \text{dragon} \\ 9: \text{snake} \\ 10: \text{horse} \\ 11: \text{sheep} \end{array} \right.$



Conditional Operator

```

if (x > 0)
    y = 1;
else
    y = -1;

```

❖ is equivalent to:

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

(boolean-expression) ? expression1 : expression2



15

Conditional Operator

```

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" );

```



16

Formatting Output

❖ 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.



17

Frequently-Used Specifiers

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

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

items

```
display      count is 5 and amount is 45.560000
```

18

