

Java Programming AP Edition

U2C4 Decisions (Selections)

IF

ERIC Y. CHOU, PH.D.

IEEE SENIOR MEMBER



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);
```



Boolean Data Type

The Boolean data type declares a variable with the value either true or false.

Relational Operators				
Java Operator	Math Symbol	Name	Example	Result
<	<	Less than	radius < 0	false
<=	≤	Less than or Equal to	radius <= 0	false
>	>	Greater than	radius > 0	true
>=	≥	Greater than or equal to	radius >= 0	true
==	=	Equal to	radius == 0	false
!=	≠	Not Equal to	radius != 0	true

Boolean literals: **true** and **false**. These are the only values that will be returned by the Boolean expressions.



if statement

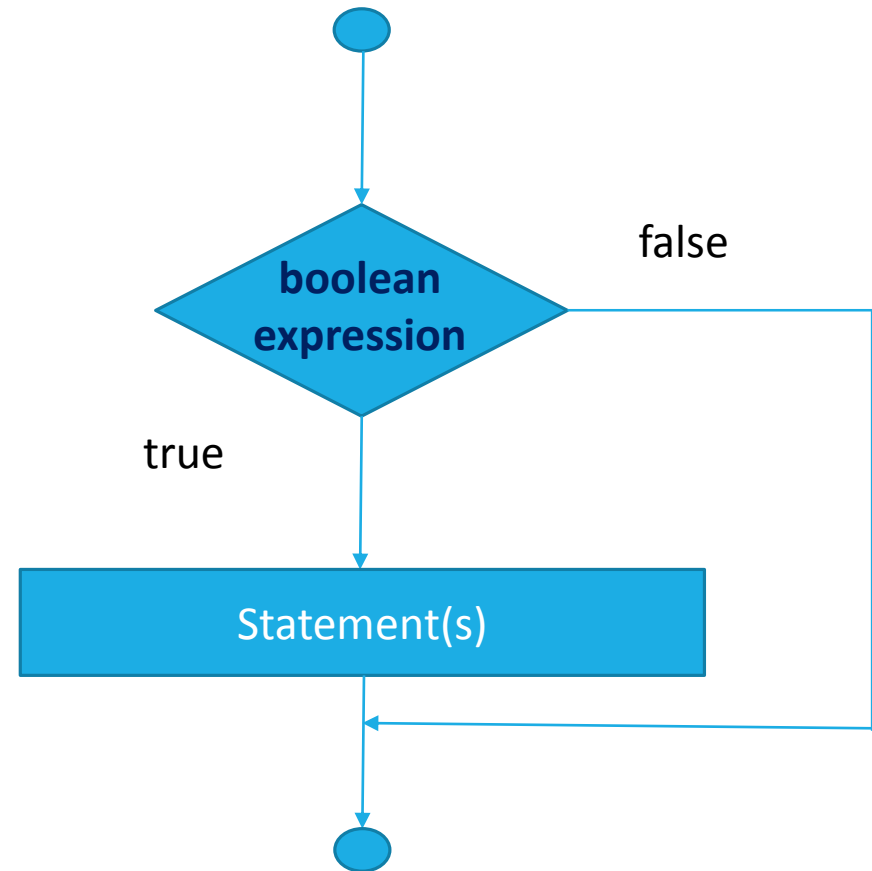
An if statement is a construct that enables a program to specify alternative paths of execution.

One way if statement:

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

If there is only single statement:

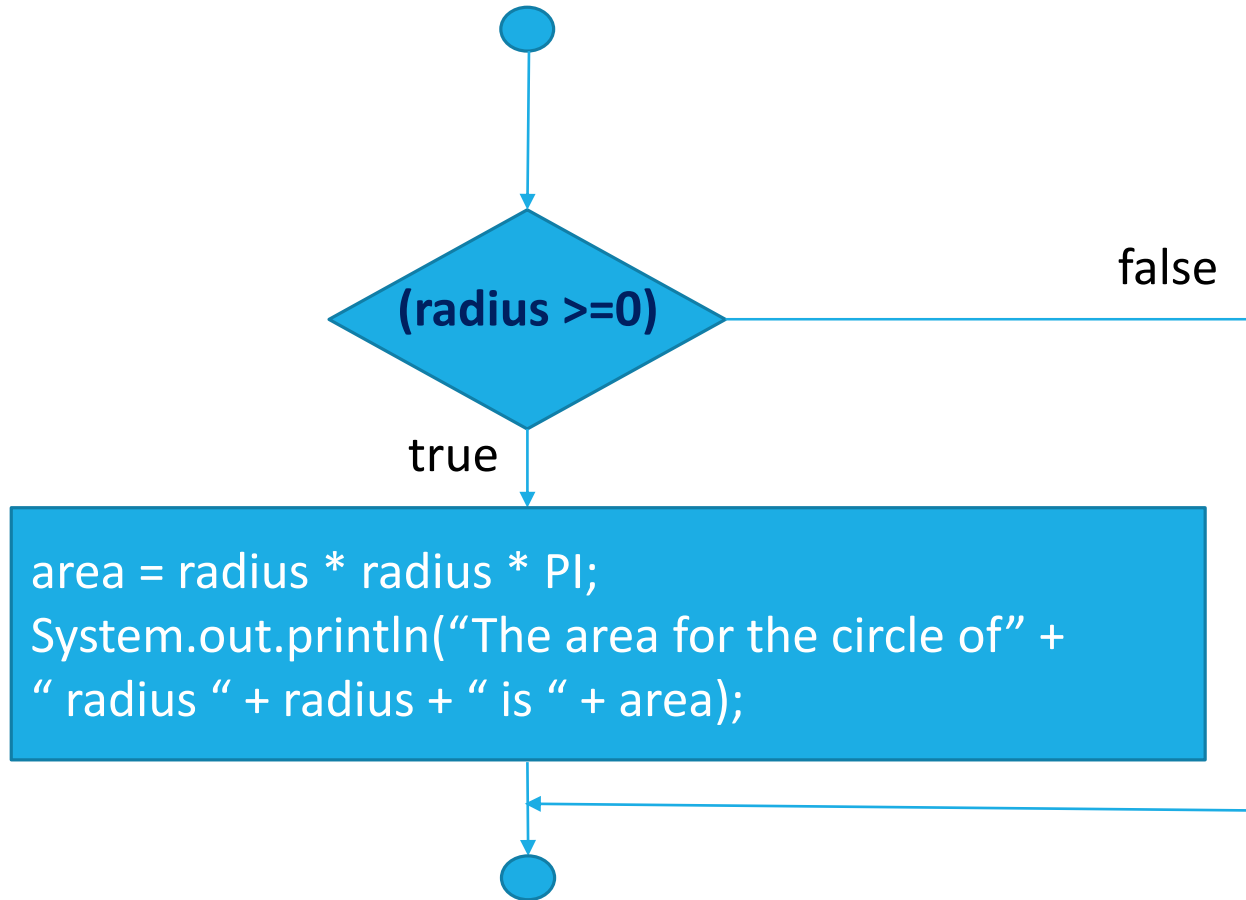
```
if (i>0) i++; is equivalent to  
if (i>0) {  
    i++;  
}
```





if statement

An if statement is a construct that enables a program to specify alternative paths of execution.

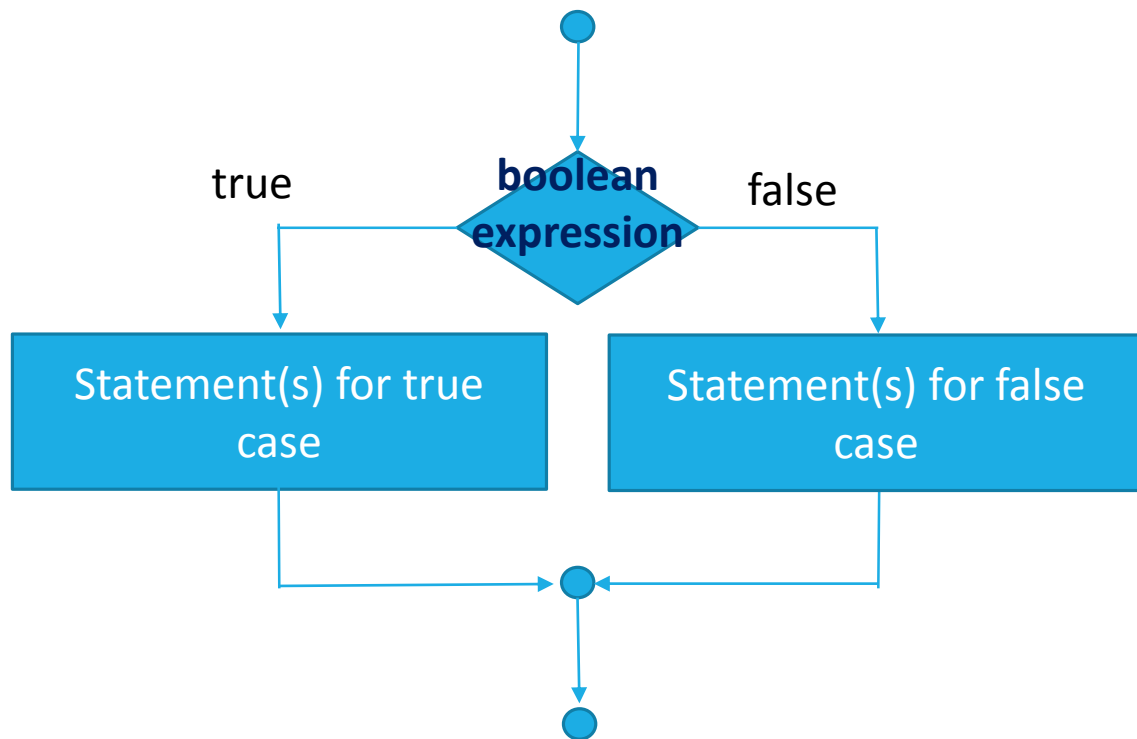


```
if (radius >= 0) {  
    area = radius * radius * PI;  
    System.out.println(" The area "  
    + " for the circle of radius "  
    + radius + " is " + area);  
}
```



Two-way if-else statements

An if-else statement decides the execution path based on whether the condition is true or false.



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



Nested if and Multi-Way if-else Statements

An if statement can be inside another *if* statement to form a nested *if* statement.

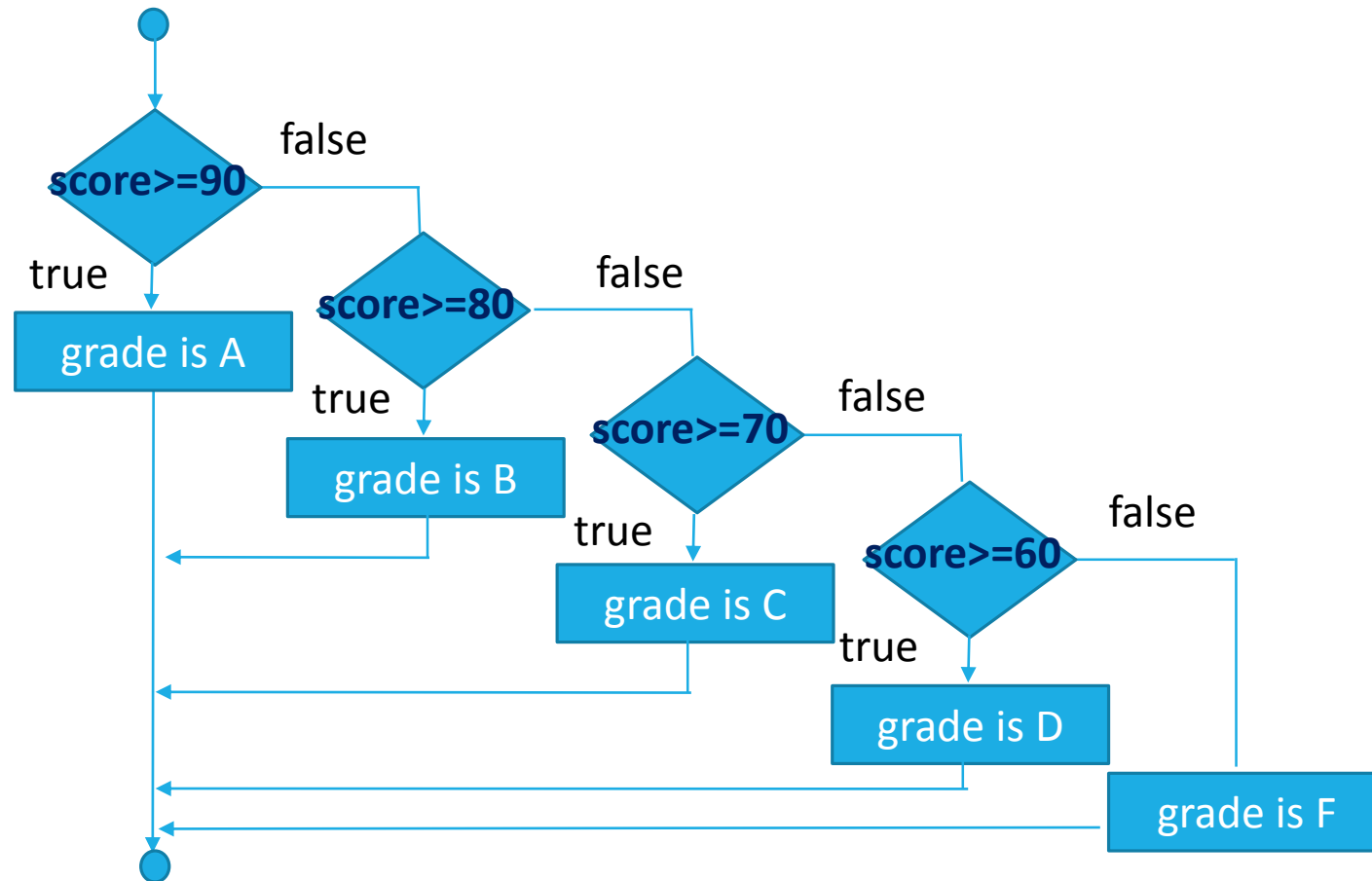
The statement in an *if* or *if-else* statement can be any legal Java statement, including another *if* or *if-else* statement.

The inner *if* statement is said to be nested inside the outer *if* statement.



Nested if and Multi-Way if-else Statements

An if statement can be inside another *if* statement to form a nested *if* statement.





Nested if and Multi-Way if-else Statements

An if statement can be inside another *if* statement to form a nested *if* statement.

```
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");
```

```
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");
```



Note

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

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.



TIP

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

(a)

Equivalent

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

(b)

CAUTION

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

(a)

Equivalent

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

(b)



Common Errors and Pitfalls

Common Error 1: Forgetting Necessary Braces.

- **if (logical expr.) a++; b++; c++;**

Common Error 2: Wrong Semicolon at the if Line.

- if (logical expr.) **}**; {.....}

Common Error 3: Redundant Testing of Boolean Values.

- if (**even == true**) ... when even is a **Boolean** variable.

Common Error 4: Dangling else Ambiguity.

- **if (A) if (B) statements; else statement; (Use parenthesis or combine the two)**



Common Errors and Pitfalls

Common Error 5: Equality Test of Two Floating Point Values.

- **Floating point is not accurate can not be tested for equality. It will cause trouble.**

Common Pitfall 1: Simplifying Boolean Variable Assignment

- Use Boolean assignment to replace if statement:
- **boolean even = number % 2 == 0; (mod(%), equality check have higher priority than assignment)**

Common Pitfall 2: Avoiding Duplicate Code in Different Case.

- A statement of same result is put in both if and else cases.