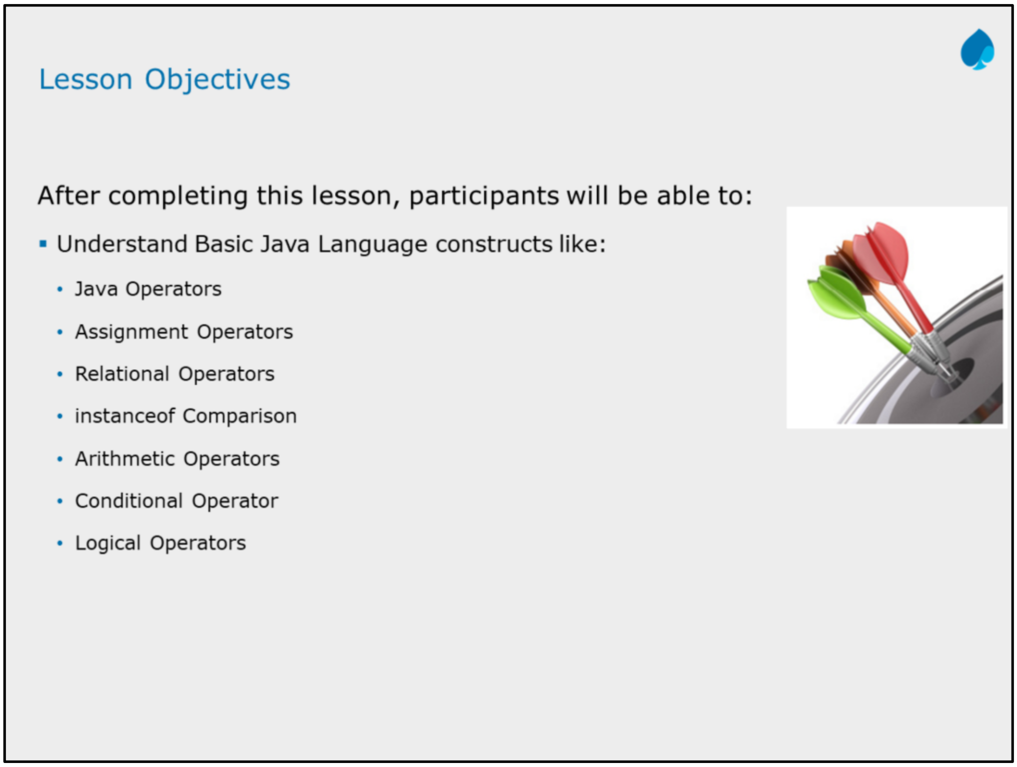
Core Java 8

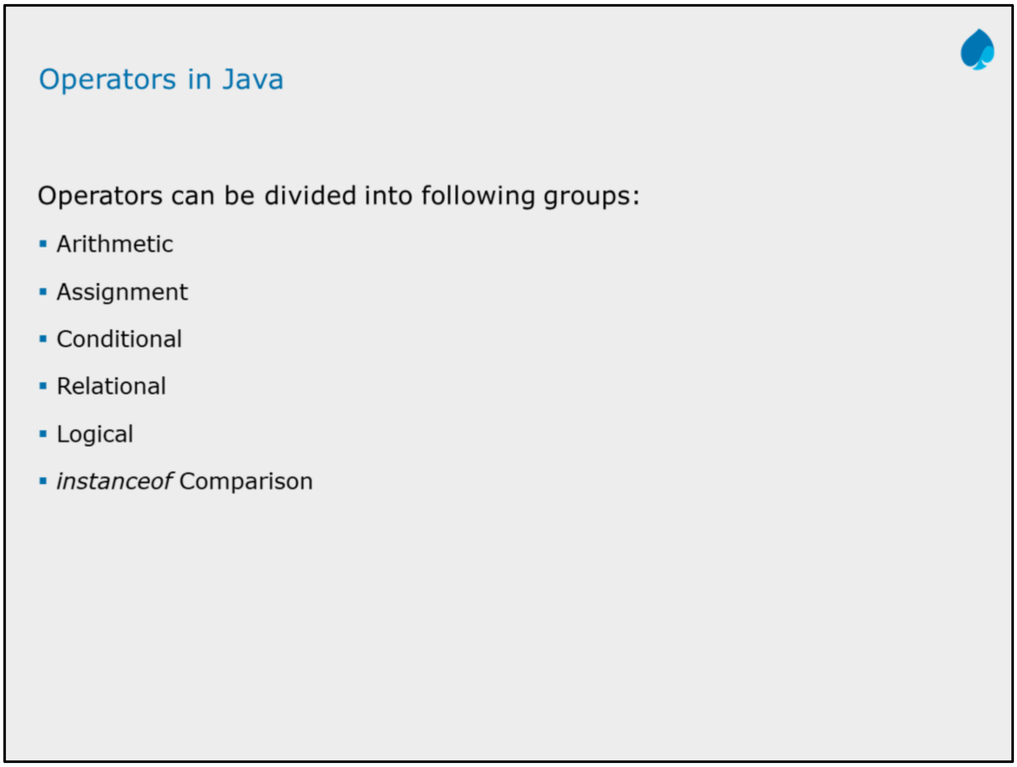
Lesson 04 : Operators



Lesson Objectives

After completing this lesson, participants will be able to:

- Understand Basic Java Language constructs like:
  - Java Operators
  - Assignment Operators
  - Relational Operators
  - instanceof Comparison
  - Arithmetic Operators
  - Conditional Operator
  - Logical Operators

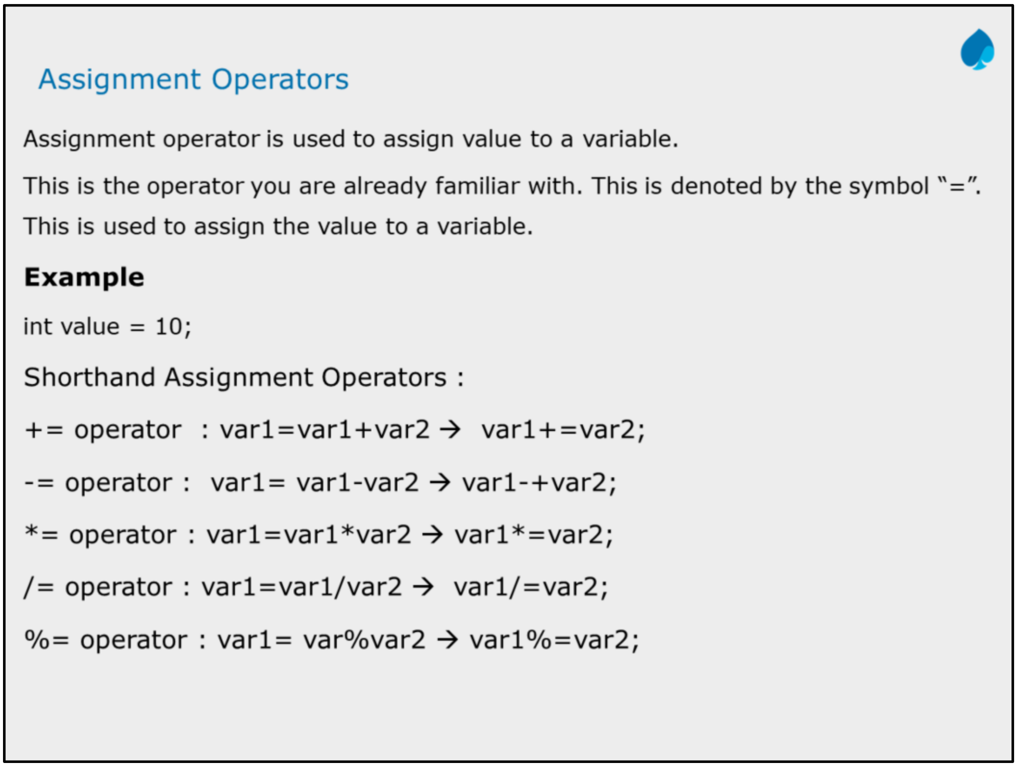


Operators in Java

Operators can be divided into following groups:

- Arithmetic
- Assignment
- Conditional
- Relational
- Logical
- *instanceof* Comparison

Java provides a rich set of operators to manipulate variables. These are classified into several groups as shown above.



Assignment Operators

Assignment operator is used to assign value to a variable.

This is the operator you are already familiar with. This is denoted by the symbol "=". This is used to assign the value to a variable.

Example

int value = 10;

Shorthand Assignment Operators :

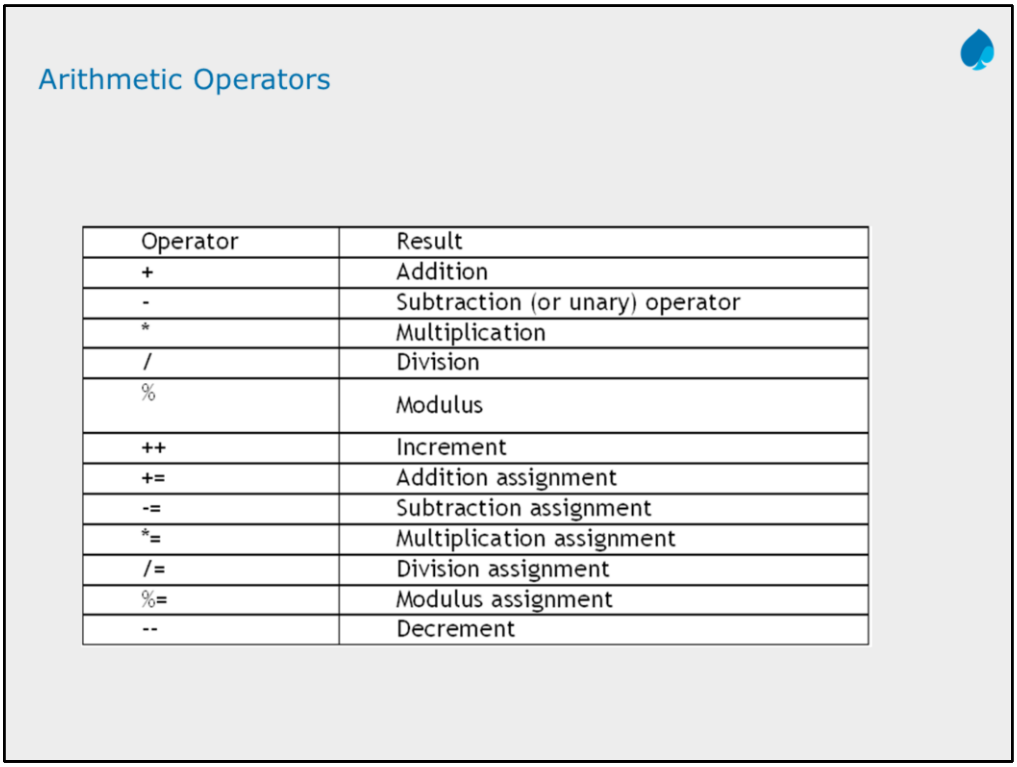
+= operator : var1=var1+var2 → var1+=var2;

-= operator : var1= var1-var2 → var1-+var2;

*= operator : var1=var1*var2 → var1*=var2;

/= operator : var1=var1/var2 → var1/=var2;

%= operator : var1= var%var2 → var1%=var2;



Arithmetic Operators

Operator	Result
+	Addition
-	Subtraction (or unary) operator
*	Multiplication
/	Division
%	Modulus
++	Increment
+=	Addition assignment
-=	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%=	Modulus assignment
--	Decrement

Arithmetic operators are summarized in the table above:

Integer division yields an integer quotient for example, the expression 7 / 4 evaluates to 1, and the expression 17 / 5 evaluates to 7. Any fractional part in integer division is simply discarded (i.e., truncated) no rounding occurs.

Java provides the remainder operator, %, which yields the remainder after division. The expression x % y yields the remainder after x is divided by y. Thus, 7 % 4 yields 3, and 17 % 5 yields 2. This operator is most commonly used with integer operands, but can also be used with other arithmetic types.

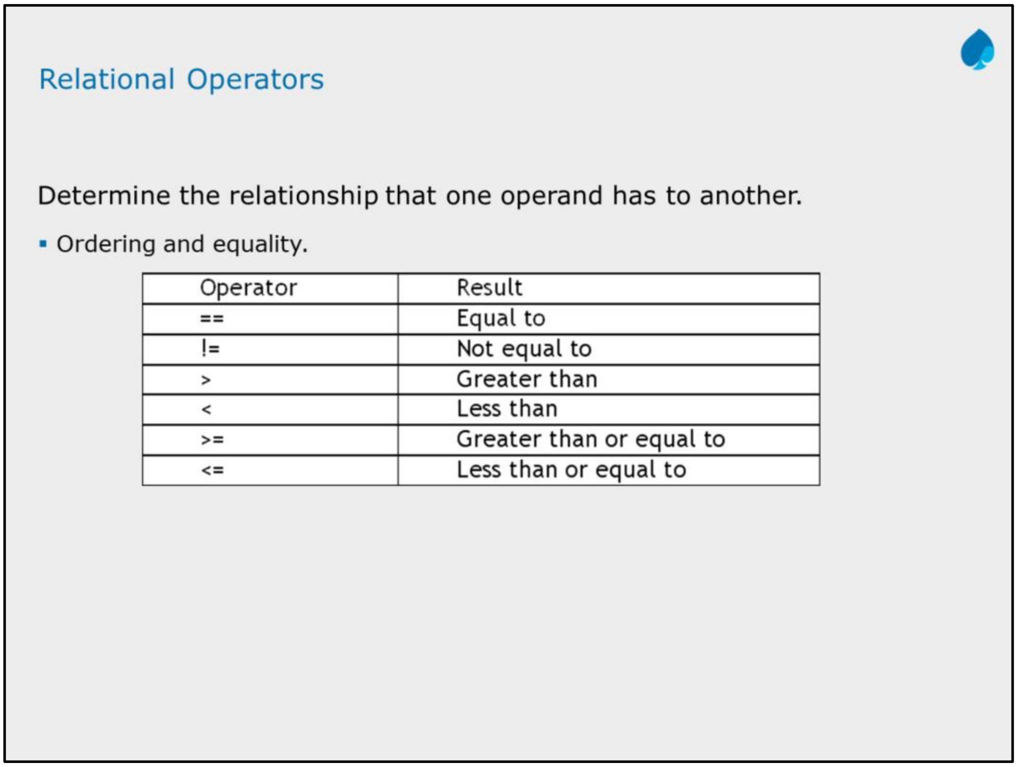
Parentheses are used to group terms in Java expressions in the same manner as in algebraic expressions. For example, to multiply a times the quantity b + c, we write **a*(b+c).**

If an expression contains nested parentheses, such as **((a+b)*c)** the expression in the innermost set of parentheses (a + b in this case) is evaluated first.

Order of Precedence:

Multiplication, division and remainder operations are applied first. If an expression contains several such operations, the operators are applied from left to right. **Multiplication, division and remainder operators have the same level of precedence**.

Addition and subtraction operations are applied next. If an expression contains several such operations, the operators are applied from **left to right**. **Addition and subtraction operators have the same level of precedence.**



Relational Operators

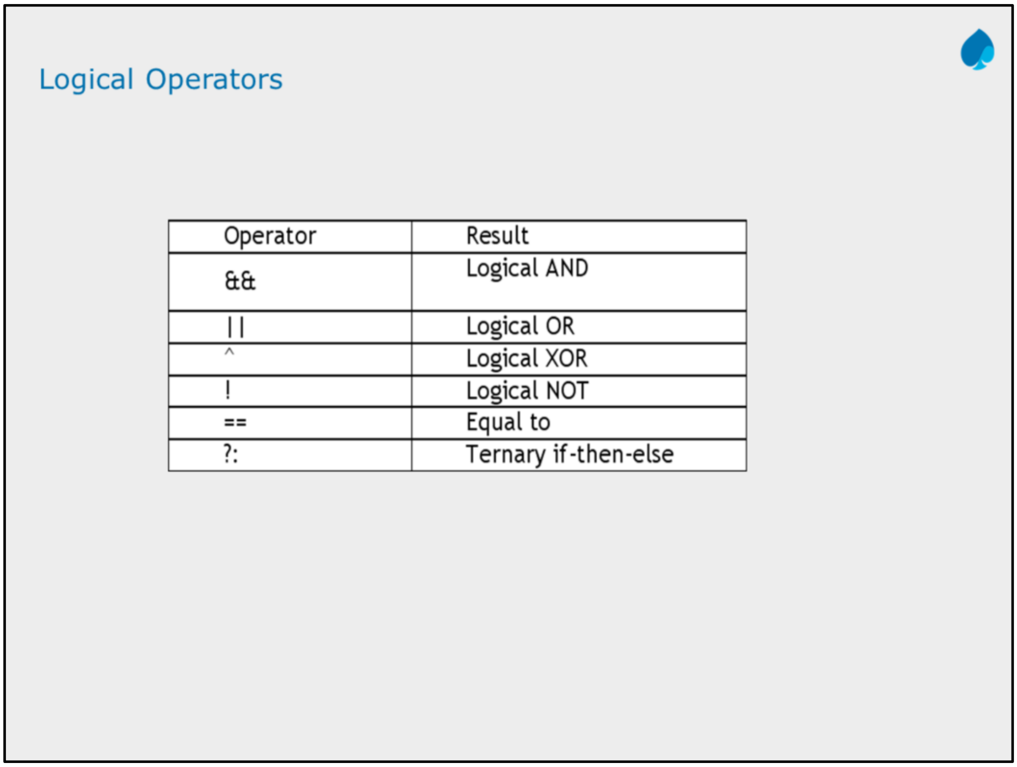
Determine the relationship that one operand has to another.

- Ordering and equality.

Operator	Result
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

A condition is an expression that can be either true or false. For example, the condition "grade is greater than or equal to 60" determines whether a student passed a test. If the condition in an if statement is true, the body of the if statement executes. If the condition is false, the body does not execute.

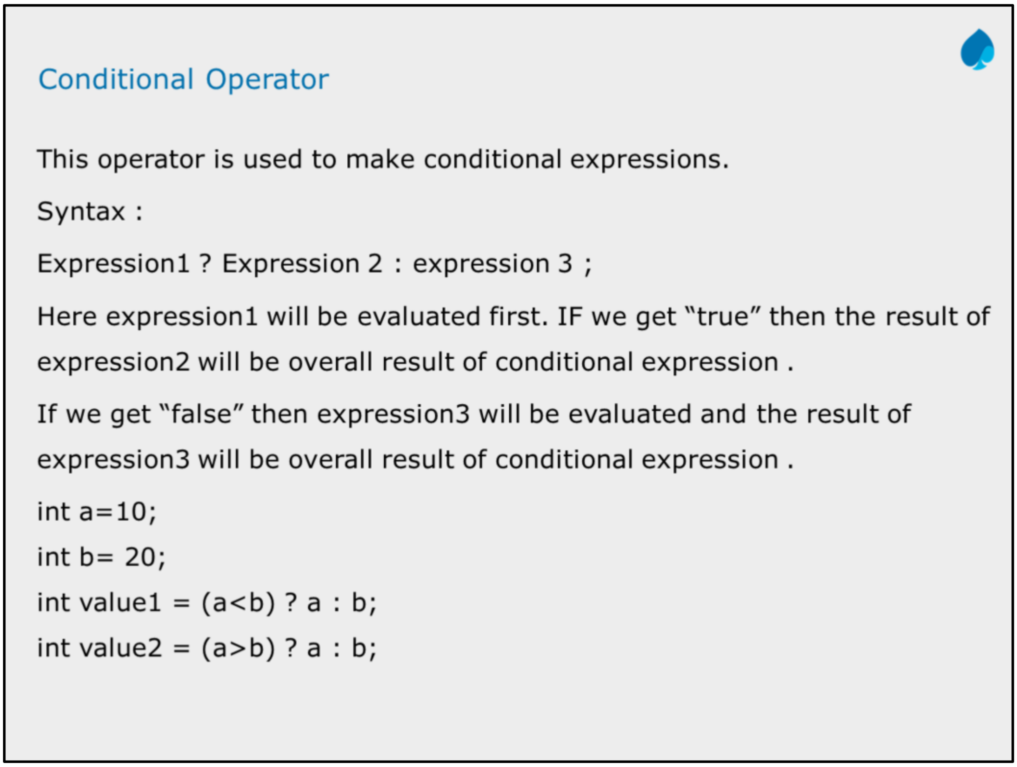
Conditions in if statements can be formed by using the equality operators (== and !=) and relational operators (>, <, >= and <=). Both equality operators have the same level of precedence, which is lower than that of the relational operators. The equality operators associate from left to right. The relational operators all have the same level of precedence and also associate from left to right.



Logical Operators

Operator	Result
&&	Logical AND
||	Logical OR
^	Logical XOR
!	Logical NOT
==	Equal to
?:	Ternary if-then-else

Java provides logical operators to enable programmers to form more complex conditions by combining simple conditions. The logical operators are && (conditional AND), || (conditional OR), & (boolean logical AND), | (boolean logical inclusive OR), ^ (boolean logical exclusive OR) and ! (logical NOT).



Conditional Operator

This operator is used to make conditional expressions.

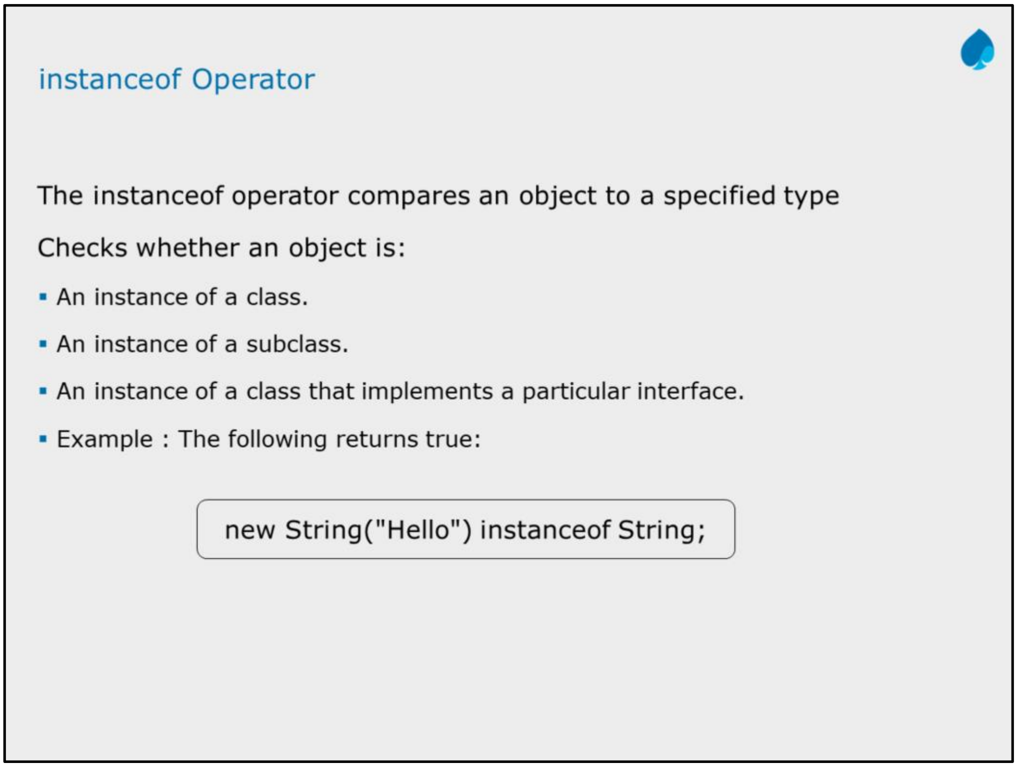
Syntax :

Expression1 ? Expression 2 : expression 3 ;

Here expression1 will be evaluated first. IF we get “true” then the result of expression2 will be overall result of conditional expression .

If we get “false” then expression3 will be evaluated and the result of expression3 will be overall result of conditional expression .

```
int a=10;
int b= 20;
int value1 = (a<b) ? a : b;
int value2 = (a>b) ? a : b;
```



instanceof Operator

The instanceof operator compares an object to a specified type

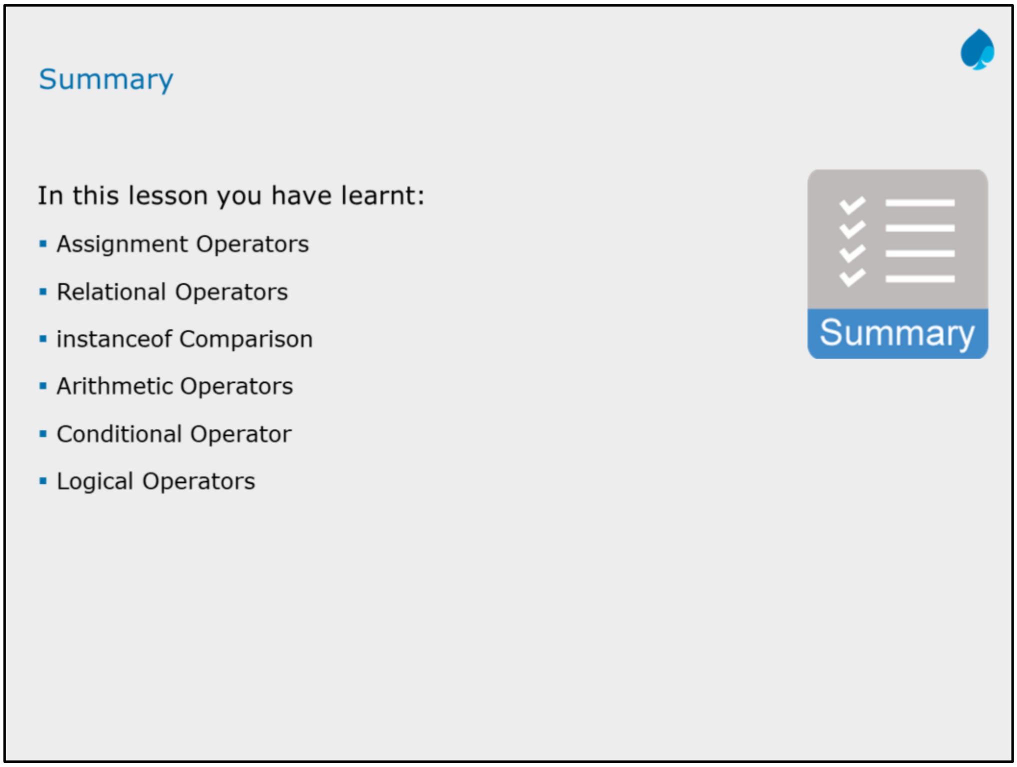
Checks whether an object is:

- An instance of a class.
- An instance of a subclass.
- An instance of a class that implements a particular interface.
- Example : The following returns true:

```
new String("Hello") instanceof String;
```

The instanceof operator is used to make a test whether the given object belongs to specified type. Consider the below example. The if statement returns true here as the child object is type of its superclass.

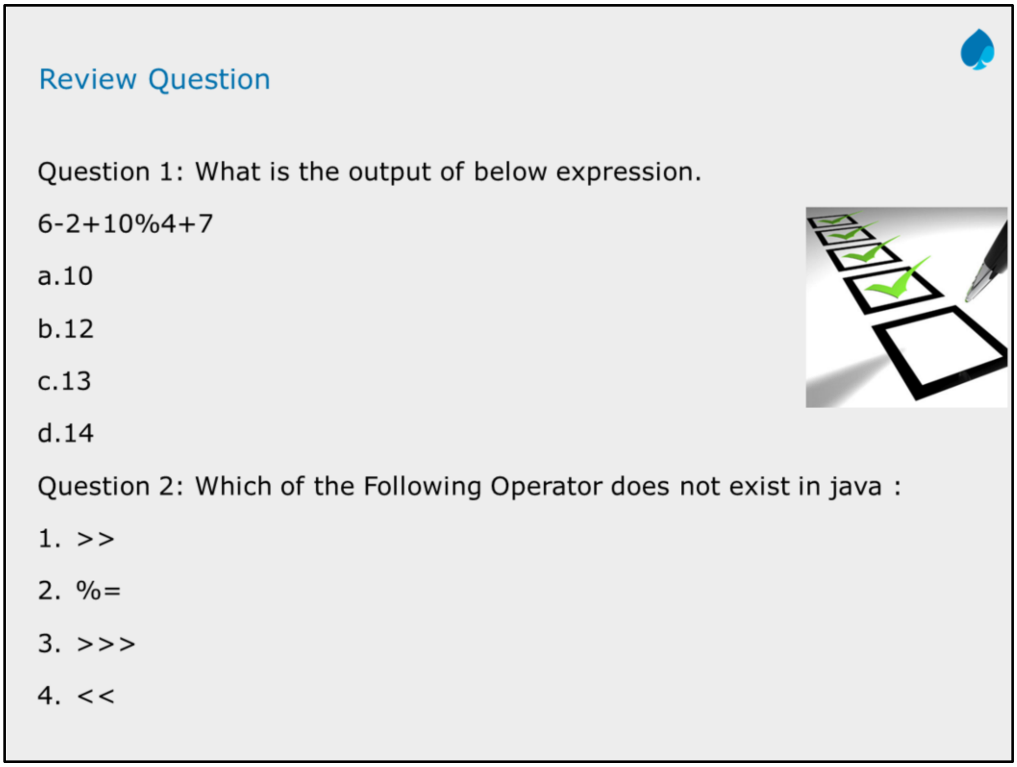
```
class Ticket{
}
class ConfirmedTicket extends Ticket {
}
…
…
ConfirmedTicket tkt= new ConfirmedTicket();
If(tkt instanceof Ticket) {
    //some processing
}
```



Summary

In this lesson you have learnt:

- Assignment Operators
- Relational Operators
- instanceof Comparison
- Arithmetic Operators
- Conditional Operator
- Logical Operators



Review Question

Question 1: What is the output of below expression.

6-2+10%4+7

a.10

b.12

c.13

d.14

Question 2: Which of the Following Operator does not exist in java :

1. >>
2. %=
3. >>>
4. <<