



* Assignment Solution *

* Operators and Loops *

Q.1 what are the Conditional operators in Java ?

Ans 1 They are used when a Condition comprises more than one boolean expression. For instance, if we want to print a number only if it is greater than 2 and less than 5, then we will use Conditional operators to combine the 2 expressions. We have 3 types of Conditional operators - logical-and, logical-or and ternary operator.

Logical-and operator (&)

It is used when we want the Condition to be true iff both the expressions are true.

* Syntax *

```
if (Condition-1 & Condition-2) {  
    Statement;  
}
```

-

Example

print the Number if the input value is greater than 5 and less than 10.

Code

```
if (val > 5 && val < 10) {  
    System.out.print(val);  
}
```

Case-1: val = 3

output - No output

Explanation - The input value is less than 10 but it is not greater than 5.

Case-2: val = 7

output - 7

Explanation - The input value is ^{both} greater than 5 and Less than 10.

Logical-or operator (||)

This operator is used when we are satisfied as long as any one of the boolean expression is evaluated as true.

Syntax

if (Condition-1 || Condition-2) {

Statements;

}

* Example *

print the number if the input value is greater than 10 or less than 5.

* Code *

```
if ( val < 5 || val > 10 ) {
```

```
    System.out.print(val);
```

```
}
```

Case-1: val = 3

output - 3

Explanation - The input value is less than 5. it is enough to satisfy the condition so the second condition won't be tested and the val will be printed.

Case-2: val = 7

output - No output

Explanation - Both the conditions are evaluated as false.

* Ternary operator (?:) *



It is a smaller version for the if-else statement. If the condition is true then the statement-1 is executed else the statement-2 is executed.

* Syntax *

Condition ? statement-1 : statement-2 ;

* Example *

```
val % 2 == 1 ? System.out.println("value entered is odd") : System.out.println("value entered is even");
```

Case-1: val = 1

output - value entered is odd

* Q.2 * What are the types of operators based on the number of operands?

there are 3 types of operators based on the number of operands.

1. unary operator
2. Binary operator
3. Ternary operator

* Unary operators *

unary operators work with a single operand.

* Example *

- ++ (increment)
- (decrement)
- (negation)
- ! (logical NOT)

* Binary operators *

Binary operators work with two operands.

* Example *

+	(addition)	!=	(inequality)
-	(subtraction)	>	(greater than)
*	(multiplication)	>=	(greater or equal to)
/	(division)	<	(less than)
%	(modulus)	<=	(less than or equal to)
==	(equality)	&&	(logical AND)
			(logical OR)

- & (Bitwise AND)
- | (Bitwise OR)
- ^ (Bitwise XOR)
- << (left shift)
- >> (right shift)
- >>> (unsigned right shift)

* Ternary operation *

ternary operation is the only operation in Java that takes three operands.

* Syntax *

Condition ? expression1 : expression2
if evaluates the Condition and returning 'expression1' if the Condition is true, or 'expression2' if the Condition is false.

* Q.3 * what is the use of Switch case in Java programming?

* Ans * Let's say we have a variable. Now, we want to do multiple operations on it based upon what value it is storing. In such cases the Switch statements comes into play.

It is like an if-else ladder with multiple conditions, where we check for equality of a variable with various values.

It works with byte, short, int, long, enum types, String and some wrapper types like Byte, Short, int, and Long.

Since Java 7, you can use String in

the switch statement.

Syntax

```
Switch( expression ) {
```

```
    case X:
```

```
        // code
```

```
        break;
```

```
    case Y:
```

```
        // code
```

```
        break;
```

```
    .
```

```
    .
```

```
    .
```

```
    .
```

```
    default:
```

```
        // code
```

```
}
```

Note: The case value must be literal or constant, and must be unique.

Q.4 What are the priority levels of arithmetic operators in Java?

Ans following are the priority levels of arithmetic operators in Java?



operation	Meaning	precedence
-	unary minus	highest
+	unary plus	highest
*	Multiplication	Middle
/	division	Middle
%	remainder	Middle
+	addition	low
-	subtraction	low

Q.5 What are the Conditional statements and use of Conditional statements in Java?

Soln They are used in

Conditional Statements in Java, allow you to Control the flow of program based on certain condition. They enable you to Make decision and execute different code blocks depending on whether a specified condition is true or false. In Java, there are three main types of Conditional Statements: 'if', 'else if' and 'else'.

* Syntax of if Statement *

```
if (Condition) {
```

```
// code to be executed if the condition is true
```

```
}
```


else if Statement

* Syntax of else if statement *

```
if (Condition1) {
```

```
    // Code to be executed if Condition1 is true
```

```
}
```

```
else if (Condition2) {
```

```
    // Code to be executed if Condition2 is true
```

```
}
```

```
else {
```

```
    /* Code to be executed if none of  
    conditions are true */
```

```
}
```

* Syntax of else Statement *

```
if (Condition) {
```

```
    // Code to be executed if Condition is true
```

```
}
```

```
else {
```

```
    // Condition is true
```

```
    // Code to be executed if the Condition is true
```

```
}
```

Q.6 What is the syntax of if else statements?

Solⁿ following are the syntax of if else statements: —

```
if (Condition) {
```

```
    // Code to be executed if Condition is true
```

```
    ?  
    else {
```

```
        // Code to be executed if Condition is false
```

```
    }
```

Q.7 What are the 3 types of iterative statements in Java?

Solⁿ following are the 3 types of iterative statements in Java: —

- (1) The while loop
- (2) The for loop
- (3) The do-while loop

* Syntax of the while loop *

```
Initialization;  
while (Condition) {  
    statements;  
    Inc/dec  
}
```

* Syntax of the for loop *

```
for (initialization; condition; updation) {  
    statement;  
}
```

* Syntax of do-while loop *

```
Initialization;  
do {  
    statement;  
    updation;  
} while (condition)
```

Q.8 What are the difference b/w for loop and do-while loop?

for loop: - (1.) The 'for' loop is used when you know in advance how many times you want to execute a block of code.

(2) It has specific structure with initialization, a condition, and an iteration expression.

(3) It first checks the condition and then executes the code block.

(4) If the condition is false from the beginning, the code block inside the loop ~~is not~~ ~~not~~ not executed.

execute at all.

* Do-while loop *

- (1) The 'do-while' loop is used when you want to execute a block of code at least once, and then repeat it as long as a certain condition is true.
- (2) It has a structure where the code block is executed first and then the condition is checked.
- (3) It guarantees that the code block will run at least once, even the condition is initially false.

Q.8 Write a program to print numbers from 1 to 10

```
for (int i = 1; i <= 10; i++) {  
    System.out.print(i);  
}
```