

ASSIGNMENT-6(29 JAN)

Q. What are the Conditional operators in java?

Ans- When a condition consist of multiple Boolean expressions, they are used. Use conditional operators to combine two expressions, such as when we wish to print a number only if it is larger than 2 and less than 5, respectively.

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

Ans- Operators in java can be classified into 6 types:-

Arithmetic operators, Assignment operators, Relational operators, Logical operators, Unary operators, Bitwise Operators.

Q. What is the use of switch case in java programming?

Ans- If we want to do multiple operations based upon what value it is storing, in such cases switch cases comes into play. It is like an if else ladder with multiple condition, where we check for equality of a variable with various values.

Q. What are the priority levels of arithmetic operations in java?

Ans- It stands for parenthesis, Exponents, Multiplication/ Division , Addition/ Subtraction.

Q. What are the conditional statements and use of conditional statements in java?

Ans- Java follows suit by conditional statements to manage the program's flow. This is crucial because at some time, in order to advance with our code, we must fulfil requirements.

Java has the following conditional statements:-

If statement, else statement, if-else ternary, Nested if-else, Switch statement.

Q. What is the syntax of if-else statement?

Ans- if(condition)

{

Statement1;

}else{

Statement2;

}

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

Ans- For Loop, While Loop, do-while loop

Q. Write the difference between for loop and do while loop.?

Ans- In for loop, we do initialization, condition checking, and increment in one single line whereas in do-while loop we first print the statement once and then check the condition whether it is true or false.

Q. Write a program to print number from 1 to 10.

```
Ans- public class print_number {  
    public static void main(String[] args) {  
        for (int i = 1; i <= 10; i++) {  
            System.out.println(i);  
        }  
    }  
}
```