22ND MAY ASSIGNMENTS

1. **What are the Conditional Operators in Java?**

1. **Ans**: The Ternary Operator (? :): The ternary operator is the only conditional operator in Java that takes three operands. It is also known as the conditional operator. Its syntax is as follows:

code

condition? expression1: expression2

The condition is evaluated first, and if it is true, the expression1 is executed. Otherwise, if the condition is false, the expression2 is executed. The result of the ternary operator is the value of either expression1 or expression2, depending on the condition.

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

Ans: Operators in programming languages can be categorized based on the number of operands they work with. The three main categories of operators based on the number of operands are:

1. Unary Operators: Unary operators work with a single operand. They perform operations on a single value or variable. Some examples of unary operators are:
   - Unary plus (+): Represents the positive value of an expression.
   - Unary minus (-): Negates the value of an expression.
   - Increment (++): Increases the value of a variable by 1.
   - Decrement (--): Decreases the value of a variable by 1.
   - Logical negation (!): Negates the Boolean value of an expression.
   - Bitwise complement (~): Inverts the bits of an integer.
2. Binary Operators: Binary operators work with two operands. They perform operations between two values or variables. Some examples of binary operators are:
   - Arithmetic operators: Addition (+), subtraction (-), multiplication (*), division (/), modulus (%).
   - Relational operators: Equality (==), inequality (! =), greater than (>), less than (<), greater than or equal to (>=), less than or equal to (<=).
   - Logical operators: Logical AND (&&), logical OR (||).

- **Assignment operators: Assigns a value to a variable, such as (=), addition and assignment (+=), subtraction and assignment (-=), multiplication and assignment (*=), etc.**

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

Ans: The switch statement is particularly useful when you have a variable with multiple possible values and want to execute different code blocks based on those values. It provides a more concise and structured way to handle such scenarios compared to using multiple if-else statements.

The general syntax of a switch statement in Java is as follows:

code

```
switch (expression) {

case value1: // code to be executed if expression matches value1.

break;

case value2: // code to be executed if expression matches value2

break;

// more cases can be added

default: // code to be executed if expression does not match any case

}
```

4. **What are the priority levels of arithmetic operation in Java?**

1. **Ans: Postfix Operators: ++, -- These operators are used to increment or decrement a value after it has been used in an expression.**
2. **Unary Operators: +, -, ++, --, ! Unary operators perform operations on a single operand. They can be used to negate a value, change its sign, or perform logical negation.**
3. **Multiplicative Operators: *, /, % Multiplicative operators perform multiplication, division, and modulus operations.**
4. **Additive Operators: +, - Additive operators perform addition and subtraction operations.**

5. **What are the conditional Statements and use of conditional statements in Java?**

Ans: Conditional statements in Java are used to control the flow of a program based on certain conditions. They allow you to make decisions and execute different blocks of code depending on whether a condition is true or false.

if-else if-else statement: The if-else if-else statement allows you to test multiple conditions and execute different code blocks based on the results. It provides a way to handle multiple possible cases. The syntax of the if-else if-else statement is as follows:

code

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 all conditions are false }

The conditions are evaluated in order. If the condition in the if statement is true, its corresponding code block is executed. If the condition is false, the next else if condition is evaluated. If any of the else if conditions are true, their corresponding code blocks are executed. If none of the conditions are true, the code block in the final else statement is executed.

Conditional statements allow you to add logic and decision-making capabilities to your programs. They enable you to control the flow of execution based on different conditions, making your programs more flexible and capable of handling various scenarios.

6. **What is the syntax of if else statement?**

Ans: The syntax of the if-else statement in Java is as follows:

code

if (condition) { // code to be executed if the condition is true }

else { // code to be executed if the condition is false }

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

Ans: In Java, there are three types of iterative statements, also known as loops, which are used to repeatedly execute a block of code as long as a certain condition is met or for a specified number of iterations. The three types of iterative statements in Java are:

1. for loop: The for loop is used when you know the number of iterations in advance. It consists of three parts: initialization, condition, and increment/decrement. The syntax of a for loop is as follows:

code

for (initialization; condition; increment/decrement)

{ // code to be executed in each iteration }

The initialization step initializes the loop control variable, the condition is the condition that is checked before each iteration, and the increment/decrement is performed after each iteration. The code block inside the for loop is executed repeatedly as long as the condition is true.

2. <u>while loop:</u> The while loop is used when you don't know the number of iterations in advance and want to repeat a block of code until a certain condition becomes false. The syntax of a while loop is as follows:

code

while (condition)

{ // code to be executed in each iteration // condition update (if applicable) }

The condition is checked before each iteration, and if it is true, the code block inside the while loop is executed. The loop continues until the condition becomes false. It's important to include an update to the condition inside the loop if you want to eventually terminate the loop.

3. <u>do-while loop:</u> The do-while loop is similar to the while loop but with a slight difference. It executes the code block first and then checks the condition. If the condition is true, it continues to execute the code block repeatedly. The syntax of a do-while loop is as follows:

code

do { // code to be executed in each iteration // condition update (if applicable) } while (condition);

The code block is executed at least once because the condition is checked after the execution. If the condition is true, the loop continues. Similar to the while loop, it's important to update the condition inside the loop if you want to eventually terminate the loop.

8. **Write the difference between for loop and do-while loop?**

Ans: The for loop and do-while loop are both iterative statements in Java used to repeat a block of code. However, they have some key differences in their syntax and behaviour:

Syntax:

- for loop:

code

for (initialization; condition; increment/decrement)

{ // code to be executed in each iteration }

- do-while loop:

code

do { // code to be executed in each iteration // condition update (if applicable) } while (condition);

9. **Write program to print numbers from 1 to 10?**

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