Day 8 PW Skills Assignment

1. **What are the Conditional operators in java?**

Ans. There are three types of the conditional operators in java:

- Conditional AND
- Conditional OR
- Ternary Operator

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

Ans. Operators in Java can be classified based on the number of operands they work with.

1. **Unary Operators**: Unary operators work with a single operand.
    - Unary plus (+): Represents positive values. For example, **+5**.
    - Unary minus (-): Negates the value. For example, **-5**.
    - Increment (++) and Decrement (--): Increase or decrease the value by 1. For example, **x++** or **--y**.
    - Logical complement (!): Negates a Boolean value. For example, **!true**.
2. **Binary Operators**: Binary operators work with two operands.
    - Arithmetic Operators: Perform mathematical operations.
        - Addition (+).
        - Subtraction (-).
        - Multiplication (*).
        - Division (/).
        - Modulo (%).
    - **Relational Operators**: Compare two values and return a Boolean result.
        - Equality (==)
        - Inequality (!=)
        - Greater than (>)
        - Less than (<)
        - Greater than or equal to (>=)
        - Less than or equal to (<=).
    - **Logical Operators**: Perform logical operations on boolean values.
        - Logical AND (&&): Returns true if both operands are true. For example, **true && false**.
        - Logical OR (||): Returns true if either operand is true. For example, **true || false**.
        - Logical NOT (!): Negates a boolean value. For example, **!true**.
    - **Assignment Operators**: Assign values to variables.
        - Simple assignment (=).
        - Compound assignment (+=, -=, *=, /=, %=).

	3. Ternary Operator: The ternary operator takes three operands and is represented as **condition ? expression1 : expression2**.
3. **Ans.**	**What is the use of Switch case in Java programming?** The **switch** statement in Java is a control statement that allows you to select one of many code blocks to be executed based on the value of a variable or an expression. It provides an alternative to using multiple **if-else** statements when you have a series of conditions to evaluate. Here are some use cases for the **switch** statement in Java: 1. **Case Selection**: The switch statement evaluates the value of an expression and compares it to the values specified in the case labels. When a match is found, the corresponding code block is executed. 2. **Simplified Syntax**: The switch statement provides a concise and readable way to handle multiple conditions compared to using nested if-else statements. 3. **Multiple Choices**: With switch, you can handle multiple choices more efficiently. Each case represents a different possibility, and you can define 4. **Default Case**: The default case is optional and represents the code block that executes when none of the case values match the expression. It serves as a fallback option when no specific match is found.
4. **Ans.**	**What are the priority levels of arithmetic operation in java?** In Java, arithmetic operations have predefined priority levels that determine the order in which they are evaluated. The priority levels, from highest to lowest, are as follows: 1. Parentheses: Operations enclosed in parentheses are evaluated first. They override the default order of operations. 2. Unary Operators: Unary operators, such as unary plus (+) and unary minus (-), have the next highest priority. They are applied to a single operand. 3. Multiplicative Operators: Multiplication (*), division (/), and modulo (%) operators have the same priority level and are evaluated from left to right. 4. Additive Operators: Addition (+) and subtraction (-) operators also have the same priority level and are evaluated from left to right.
5. **Ans.**	**What are the conditional statements and use of conditional statements in Java?** Conditional statements in Java allow you to control the flow of execution based on certain conditions. They enable your program to make decisions and execute different blocks of code depending on the outcome of those conditions. In Java, the main conditional statements are: 1. if statement: The if statement allows you to specify a condition, and if it evaluates to true, the associated block of code is executed. If the condition is false, the code block is skipped.

Example:

```
int num = 10;
if (num > 0) {
   System.out.println("The number is positive.");
}
```

2. if-else statement: The if-else statement extends the if statement by providing an alternative code block to execute when the condition is false.

Example:
javaCopy code

```
int num = -5;
if (num > 0) {
   System.out.println("The number is positive.");
} else {
   System.out.println("The number is non-positive.");
}
```

3. if-else-if ladder: The if-else-if ladder allows you to test multiple conditions in a sequence and execute the code block associated with the first true condition. It provides a series of if-else statements one after another.

Example:
javaCopy code

```
int num = 7;
if (num == 0) {
   System.out.println("The number is zero.");
} else if (num > 0) {
   System.out.println("The number is positive.");
} else {
   System.out.println("The number is negative.");
}
```

4. switch statement: The switch statement provides a way to select one code block from multiple alternatives based on the value of an expression. It is often used when you have a specific value to match against multiple cases.

Example:
javaCopy code

```
int dayOfWeek = 3;
switch (dayOfWeek) {
   case 1:
      System.out.println("Monday");
      break;
   case 2:
      System.out.println("Tuesday");
      break;
```

	` case 3:`
	`    System.out.println("Wednesday");`
	`    break;`
	` default:`
	`    System.out.println("Invalid day");`
	`    break;`
	`}`

6.
 	
Ans.
	What is the syntax of if else statement?
	
	Syntax:
	If(condition)
	{
	// statements.
	}
	Else
	{
	// alternate statement
	}

7.
 	
Ans.
	What are the 3 types of iterative statements in java?
	
	• For loop
	`for (initialization; condition; increment/decrement)`
	`{`
	`  // Code block to be repeated`
	`}`
	
	• While loop
	`while (condition)`
	`{`
	`// Code block to be repeated`
	`        // (condition must be eventually false to exit the loop)`
	`}`
	
	• Do while loop
	`    do {`
	`     // Code block to be repeated`
	`} while (condition);`

8. **Ans.**	**Write the difference between for loop and do-while loop?** The main difference between a for loop and a do-while loop in Java lies in when the loop condition is evaluated. 1. **For Loop**: • In a for loop, the initialization, condition, and increment/decrement statements are all part of the loop header.

- The loop condition is evaluated before each iteration. If the condition is true, the loop body is executed. If the condition is false, the loop is terminated.
- The for loop is commonly used when you know the number of iterations in advance.

Example:

```
for (int i = 1; i <= 5; i++)
{
        System.out.println(i);
}
```

2. **Do-While Loop:**

- In a do-while loop, the loop body is executed first, and then the condition is evaluated.
- The loop body is guaranteed to be executed at least once, regardless of the condition's initial value.
- After the loop body is executed, the condition is evaluated. If the condition is true, the loop body is executed again. If the condition is false, the loop is terminated.

Example:

```
int i = 1;
do
{
        System.out.println(i);
        i++;
} while (i <= 5);
```

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

Ans.

```
import.java.util.*

class PrintNumbers
{
        public static void main(String args[])
        {
                int firstNumber = 1;
                for( firstNumber; firstNumber<11; firstNumber++)
                {
                        System.out.println(firstNumber);
                }
        }// end of main method
}//end of class
```