Assignment Answers

1.What are the Conditional Operators in Java?

Answer: Conditional operators are used when a condition comprises of more than one Boolean expression .

Conditional operators are

1. **Logical Operator 'AND' "&&"**
2. **Logical Operator 'OR' "||"**
3. **Ternary Operator**

1.Logical Operator 'AND' "&&": It is used when we want a condition to be true only when both the condition are evaluated as true.

Syntax :

```
if( expression 1 && expression 2)
{
        Statement;
}
```

2.Logical Operator 'OR' "||" : It is used when we want a condition to be true even either of the expressions are evaluated as true.

Syntax :

```
if(expression 1 || expression 2)
{
        Statement;
}
```

3.Ternary Operator : It is the short form of " if – else " ladder .

Syntax :

```
(condition)? Statement 1  : Statement 2 ;
```

Statement 1 will be executed when the condition was evaluated as true else Statement 2 will be executed.

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

Answers: There are two types of operators based on the number of operands. They are ,

A. Unary operators : It takes only one operand

B. Binary operators : : It takes two operands.

(A) Unary Operators :

a . '+' (Unary plus operator) : This indicates positive value(numbers are positive without this , however.)

b. '-' (Unary minus operator) : Negates an expression value.

c. '++' (Increment operator) : Increments a value by 1 .

d. '--' (Decrement operator) : Decrements a value by 1 .

e. '!' (Logical complement operator) : Inverts the value of Boolean.

(B) Binary Operators :

a. '+' (performs Addition) : Adds the values on either sides of operator.

b. '-' (performs Substraction) : Subtracts the right hand operand from left hand operand .

c. '*' (performs Multiplication) : Multiply values on either side of the operater.

d. '/' (performs Division) : Divides left hand operand by right hand operand return the quotient .

e. '%' (Modulus operater) : Divides left hand operand by right hand operand and return the reminder.

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

Answer: Let's say, we have a variable , we want to do multiple operations on it based upon what value the variable is storing . Then we use Switch case .

- It is like if-else-if ladder with multiple conditions , where we check the equality of variable with several values specified in the test cases.
- A switch works with the byte , int , char , short primitive data types.

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

Answer: Java has a well defined rules for evaluating expressions , including these 3 rules are Operator precedence , operator associativity , Order of operand evaluation. We may call these as 3 levels. So,

a. Operator Precedence : This specifies the manner in which the operands are grouped with operators. For example 1+2*3 is treated as 1+(2*3) and 1*2+3 is treated as (1*2)+3 because multiplication operator has higher precedence than addition operator.

- We can use parenthesis ( ) to override the default operator precedence rules.
- Parenthesis ( ) and Array Subscript [ ] have the highest priority in java.

b. Operator Associative : When an expression has two operators with same precedence , then operators and operands are grouped according to their associativity . For example 72 / 2 / 3 is treated as (72 / 2) / 2 since division operator has left to right associative

- We can use parenthesis ( ) to override the default operator associative rules.
- Most of the operators are left to right associative { Including arithmetic (+ , - , / , %, *) , relational (== , != , < , > , <= , >=) , bitwise (& , | , !) } . One notable exception is the assignment operators{ = , += , -= , /= , *= , %= } , which is right to left associative. As a result x = y = a = z is treated as (x= (y= (a=z)))

c. Order of operand evaluation : Operator precedence and associative determines in which way java groups operand and operators but doesn't tell which way operands are evaluated.

- In Java, operands are always evaluated left to right . Similarly argument lists are evaluated from left to right.

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

Answer: Conditional Statements comes under Control Structures. As name suggests , it is controlling the flow of execution . Here controlling means branching , decision making , and iterating.

Generally there are 3 types of Control structures . They are

- Conditional Statements or Decisional statements (if , if – else , switch)
- Iteration Statements or Loops (for , while , do – while)
- Jump Statements (break , continue , return)

Conditional Statements : These statements are executable block of code dependent on certain conditions

1. If Statement : If , a statement executes set of statements based on certain condition.

```
Syntax :
If( condition )
{
        // statement to be executed
}
```

If clause accepts a condtion and executes the set of statements falling under it only when condition is evaluated as true.

2. If – else Statement :

```
Syntax :
If(condition)
{
        // statements are executed if condition evaluates as true
}
else
{
        // statements are executed if condition evaluates as false.
}
```

If clause evaluates the condition . If it comes out as true , statements under if block gets executed . Else , statements under the else block gets executed.

3. Switch :

```
Syntax :

switch( expression )
{
case    value 1 : Statement 1 ;
break;
case    value 2 : Statement 2;
break;
.
.
.
case     value n :  Statement n ;
default:  default statement;
}
```

Unlike if else statement , Switch has multiple paths of execution . It evaluates the expression and matches with case values and execute the respective statement

6. What is the syntax of if else statement?

Answer: Syntax :

```
if ( condition )

{

        //Statements to be executed when condition evaluates as true

}

else
```

{

// Statements to be executed when condition evaluates as false

}

If clause evaluates the condition , if it comes out as true statements under if block gets executed . Else , statements under else block gets executed.

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

Answer: Iterative (itrate – perform repeatedly) statements or Loops. Loops helps us to perform a task repeatedly so that we don't need to write explicitly code again and again . There are 3 types of loops . They are for , while , do while .

- For loop :

  Syntax:

  for (initialization ; termination condition ; increment / decrement)
  {
  // set of statements to be executed repeatedly.
  }

  Initialization : Entry point of loop with an initial value assigned to a variable

  Termination condition : acts as an exiting condition in a loop . The condition always evaluates to a Boolean value . Hence, the loop runs till the time the condition is true. The flow exits as soon as condition is false.

  Increment / Decrement : This operation applies to progressively execute the loop . Hence , it can proceed onto the next itration

- **while loop :** It works the same as of for loop

  Syntax :

  // initialization is outside of while loop
  While (condition) // termination condition
  {
  // Execute set of statements
  // increment/ decrement
  }

- **do while loop :**

  Syntax :

```
do
{
        // Statements to be iterated
}while( condition )
```

Do while loop first execute the statements under do block and then check the condition under while to proceed onto next iterations. Hence , the statements under do will be executed atleast once even if the condition is evaluated as false .

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

Answer: Both are the Iterative statements , but main difference is that the while loop give prior to condition means the statements in the while only to be executed if the condition is evaluated as true . When it comes to do-while the statements are executed first and check the condition to proceed to next iteration . So, statements under do-while be executed atleast once even if the condition is evaluated a false.

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

Answer:

SOURCE CODE

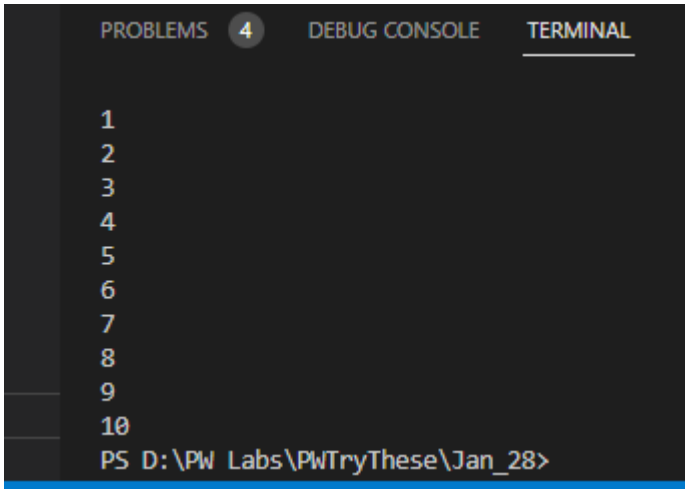
```
public class j {

    public static void main(String[] args) {

  for(int i = 1 ; i <= 10 ; i++)
  {
    System.out.println(i);

  }
}
}
```

Output:


PROBLEMS 4
DEBUG CONSOLE
TERMINAL
1
2
3
4
5
6
7
8
9
10
PS D:\PW Labs\PWTryThese\Jan_28>