

### Q1. What are the Conditional Operators in Java?

Ans:

1. Conditional AND Operator - && ( it is the optional operator if first expression is False then it would not evaluate/execute the second expression)
2. Conditional OR operator - || (it is also a optional operator if first expression is True the it would not evaluate/execute the second expression)
3. Conditional Ternary Operator – ?: (if expression is True then it will return first(first expression) value other wise False(second expression). Syntax => variable = (condition) ? exp1(return when condition would true): exp2(return when condition would False)

### Q2. What are the types of operators based on the number of operands?

1. Arithmetic operator- (+, -, \*, /)
2. Assignment operator- (+ =, -=, \*=, /=, %= )
3. Equality operator - (== , !=)
4. Increment and Decrement operator- (++ , --)
5. Relational operator- (<>, <=, >=)
6. Logical operator- (!, &, |)
7. Bitwise operator- (&, |, ~, ^, <<, >>, >>>)
8. Conditional operator- (&&, ||, ?:)

### Q3. What is the use of Switch case in Java programming?

Ans:

**Switch(expression)**

//if switch expression will match with any case then rest of block will execute

case 1:

//code block execute

case 2:

//code block execute

case 3:

//code block execute

default :

//when any case could not match with switch expression then default block will execute

### Q4. What are the conditional Statements and use of conditional statements in Java?

Ans:

if(condition){

//if condition is true

//code block execute

```

}

else if(condition){

    //if "if" block condition would false then flow will come here to check the condition of block

    //code bloc

}

else{

// if no condition would true of above block then finally this code block will execute

//code block

}

```

#### Q5.What is the syntax of if else statement?

Ans:

```

if(condition){

    //if condition is true

    //code block execute

}

else{

// if no condition would true of above block then finally this code block will execute

//code block

}

```

#### Q6.How do you compare two strings in Java?

Ans:

##### 1. By Using equals() method:

```

String s1 = "Prajjwal";

String s2 = "Prajjwal";

String s3 = "Vivek";

System.out.println(s1.equals(s2)); //True

System.out.println(s1.equalsIgnoreCase(s3)); //True

```

##### 2. By using == operation:

```

String s1 = "Prajjwal";

String s2 = "Prajjwal";

String s3 = "Vivek";

String s4 = new String("Vivek");

System.out.println(s1==s2); //True

```

```
System.out.println(s1==s3)); //True
```

System.out.println(s3==s4)); //False because s3 is a literal so it would store in the String pool storage and s4 is the new reference that would store in the Heap memory so both object(s3&s4) will point to different values

### 3. By using compareTo() method:

```
String s1 = "Prajwal";  
String s2 = "Prajwal";  
String s3 = "Vivek";
```

```
System.out.println(s1.compareTo(s2)); // 0 because s1 == s2  
System.out.println(s2.compareTo(s3)); // 1 because s2>s3  
System.out.println(s3.compareTo(s2)); // -1 because s3>s2
```

### Q7.What is Mutable String in Java Explain with an example

Ans:

Generally string is **immutable object** in java which are created by **String class** we cannot change any data of string once it is created.

But by using **StringBuffer** and **StringBuilder** class we can create such type of **string object** that will be **mutable** we can easily insert any character using **append()** method, delete using **delete()** method and using **replace()** method we can replace the character of string

Ex.

```
package com.edubca.mutablestringdemo;  
public class MutableStringDemo{  
    public static void main(String args[]){  
        StringBuffer sBuffer1=new StringBuffer("Welcome");  
        System.out.println("Original String is :: " + sBuffer1 + ":: having length " +  
            sBuffer1.length());  
        //using append method  
        sBuffer1.append(" To Edubca");  
        System.out.println("Modified String after append is :: " + sBuffer1 + " ::  
            having length " + sBuffer1.length());  
        //using reverse method  
        sBuffer1.reverse();  
        System.out.println("Modified String after Reverse is :: " + sBuffer1);  
    }  
}
```

### Q8.Write a program to sort a String Alphabetically

```
import java.util.*;
```

```
public class Main
```

```
{  
  
    public static void main(String[] args)
```

```

{
    Scanner sc = new Scanner(System.in);
    String str = sc.nextLine();
    char temp = 0;
    char[] chars = str.toCharArray(); // change string into the character array
    for (int i = 0; i < chars.length; i++) {
        for (int j = 0; j < chars.length; j++) {
            if (chars[j] > chars[i]) {
                //swapping
                temp = chars[i];
                chars[i] = chars[j];
                chars[j] = temp;
            }
        }
    }
    System.out.println("The sorted string is : ");
    for (int i = 0; i < chars.length; i++) {
        System.out.print(chars[i]);
    }
}
}

```

**Q9. Write a program to check if the letter 'e' is present in the word 'Umbrella'.**

```

public class Main
{
    public static void main(String[] args) {
        String ss = "Umbrella";
        Char[] chars = ss.toCharArray();
        for(int i=0; i<chars.lenth; i++){
            if (chars[i]=='e'){
                System.out.println("letter e is present");
            }
            else{

```

```
        System.out.println("letter e is not present");
    }

}

}
```

**Q10. Where exactly is the string constant pool located in the memory?**

Ans:

The java String Constant Pool is located inside the Heap memory where only literals are stored if any many literal already there and we create a new literal(with out using new keyword) then JVM would check the Same value is present in the String Constant Pool or not.