Assignment Questions 2

Q1. What are the Conditional Operators in Java?

In Java, **conditional operators** check the condition and decides the desired result on the basis of both conditions. In this section, we will discuss the **conditional operator in Java.**

Types of Conditional Operator

There are three types of the conditional operator in Java:

- Conditional AND
- Conditional OR
- Ternary Operator

Operator	Symbol
Conditional or Logical AND	&&
Conditional or Logical OR	||
Ternary Operator	?:

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

Number of operands – There are three types of operators based on the number of operands. An operator is called a unary, binary, or ternary operator based on the number of operands. If an operator takes one operand, it is called a unary operator; if it takes two operands, it is called a binary operator; if it takes three operands, it is called a ternary operator.

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

The switch statement is a multi-way branch statement. In simple words, the Java switch statement executes one statement from multiple conditions. It is like an if-else-if ladder statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression. Basically, the expression can be a byte, short, char, or int primitive data types. It basically tests the equality of variables against multiple values.

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

The conditional statement is a branch of code that can be executed depending on another condition. In Java, these clauses are called decision or selection statements.

There are five types of Java conditional statements:-

1. Java If Statement
2. Java If-Else Statement
3. Java If-Else-If Ladder Statement
4. Java Nested If Statement
5. Java Switch Statement

Q5.What is the syntax of if else statement?

```
if (Condition)
{
   Statement1;
}
Else
{
   Statement2;
}
```

Q6.How do you compare two strings in Java?

There are three ways to compare String in Java:

1. By Using equals() Method
2. By Using == Operator
3. By compareTo() Method

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

mutable strings are those strings whose content can be changed without creating a new object. StringBuffer and StringBuilder are mutable versions of String in java, whereas the java String class is immutable. Immutable objects are those objects whose contents cannot be modified once created.

Q8. Write a program to sort a String Alphabetically

```
import java.util.*;

public class Main
{
        public static void main(String[] args)
        {
                String str;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the string : ");
                str = sc.nextLine();
                int j = 0;
                char temp = 0;
                char[] chars = str.toCharArray();
                for (int i = 0; i < chars.length; i++) {
                        for (j = 0; j < chars.length; j++) {
                                if (chars[j] > chars[i]) {
                                        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'

```
class Check_Letter

{

    public static void main(String[] args)

    {

        String str = "String Exercises";

        boolean pre = false;

        for(int i = 0;i<str.length();i++)

        {

            if(str.charAt(i) == 'x')

            {

                pre=true;

                break;

            }

        }
```

```
                System.out.println(pre);
        }
}
```

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

Whenever we declare a variable or create an object, it is stored in the memory. At a high level, Java divides the memory into two blocks: stack and heap. Both memories store specific types of data and have different patterns for their storage and access.

The String constant pool is a special memory area. When we declare a String literal, the JVM creates the object in the pool and stores its reference on the stack. Before creating each String object in memory, the JVM performs some steps to decrease the memory overhead.

The String constant pool uses a Hashmap in its implementation. Each bucket of the Hashmap contains a list of Strings with the same hash code. In earlier versions of Java, the storage area for the pool was a fixed size and could often lead to the “Could not reserve enough space for object heap” error.

When the system loads the classes, String literals of all classes go to the application-level pool. It is because of the fact that equal String literals of different classes have to be the same Object. In these situations, data in the pool should be available to each class without any dependency.