Assignment Questions 2

Q1. What are the Conditional Operators in Java?

conditional operators, also known as ternary operators, are used to make decisions based on a condition. The main conditional operator in Java is the ternary operator, represented by the question mark "?" and the colon ":" symbols.

Syntax:
variable = (condition) ? expression1 : expression2;

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

Based on the number of operands, operators in programming languages can be categorized into three main types:

- Unary Operators.
- Binary Operators.
- Ternary Operator.

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

The switch statement in Java is a control flow statement that allows you to execute different code blocks based on the value of

and expression. It provides an alternative to using multiple if-else statements when you have a single variable to compare against multiple possible values.

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

Conditional statements are used to make decisions in the program based on certain conditions. These statements allow the program to execute different blocks of code depending on whether a given condition is true or false. The main conditional statements in Java are:

if statement: The if statement allows you to execute a block of code if a certain condition is true. It has the following syntax:

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

if-else statement: The if-else statement allows you to execute one block of code if a condition is true, and another block of code if the condition is false. It has the following syntax:

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

```
} else {
        // code to be executed
```

Q5.What is the syntax of if else statement?

The syntax of the if-else statement varies depending on the programming language you are using. However, I can provide you with a general template that showcases the structure of an if-else statement:

Syntax:

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

Q6.How do you compare two strings in Java?

In Java, we can compare two strings using the equals() method or the compareTo() method. Using equals() method: The equals() method is used to compare the content of two strings. It returns true if the content of the two strings is the same, and false otherwise.

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

In Java, strings are immutable, which means they cannot be changed once created. However, Java provides another class

called StringBuilder that allows the creation of mutable strings. Mutable strings can be modified, appended, or deleted without creating a new object every time a modification is made, resulting in improved performance and memory efficiency.

Example to demonstrate the use of StringBuilder:

```
StringBuilder mutableString = new StringBuilder("Hello");
// Appending a string
mutableString.append(", World!");
System.out.println(mutableString); // Output: Hello, World!
// Modifying the string
mutableString.insert(5, " Awesome");
System.out.println(mutableString); // Output: Hello Awesome, World!
// Deleting a portion of the string
mutableString.delete(5, 13);
System.out.println(mutableString); // Output: Hello, World!
```

Q8.Write a program to sort a String Alphabetically.

```
import java.util.Arrays;

public class StringSorter {
    public static void main(String[] args) {
        String input = "openai";
```

```
        String sortedString = sortStringAlphabetically(input);
        System.out.println("Sorted String: " + sortedString);
    }

    public static String sortStringAlphabetically(String input) {
        // Convert the string to an array of characters
        char[] charArray = input.toCharArray();

        // Sort the character array
        Arrays.sort(charArray);

        // Convert the sorted character array back to a string
        String sortedString = new String(charArray);

        return sortedString;
    }
}
```

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

```
public class LetterChecker {

    public static void main(String[] args) {

        String word = "Umbrella";
```

```
        boolean isPresent = isLetterPresent(word, 'e');


        if (isPresent) {
            System.out.println("The letter 'e' is present in the
word.");
        } else {
            System.out.println("The letter 'e' is not present in the
word.");
        }
    }


    public static boolean isLetterPresent(String word, char
letter) {
        for (char c : word.toCharArray()) {
            if (c == letter) {
                return true;
            }
        }
        return false;
```

```
    }
}
```

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

In Java, the string constant pool is a part of the Java heap memory. The Java heap is the runtime data area where objects are allocated during the execution of a Java program. It is shared among all threads in a Java application.

The string constant pool is a special area within the Java heap that is used to store string literals. String literals are the string values defined directly in the source code of a Java program, enclosed in double quotation marks (""). For example, "Hello" is a string literal.

When the Java compiler encounters a string literal, it checks if the string already exists in the constant pool. If it does, the reference to that existing string is used. If the string is not found in the constant pool, a new string object is created and added to the pool.