ASSIGNMENT

Q1 ans-

The mutable string can be changed after it is created. This is because the StringBuffer and StringBuilder class is mutable, which means that its contents can be changed without creating a new object.

Example-

```
import java.util.Scanner;


public class MutableStringExample {


    public static void main(String[] args) {
        // Create a mutable string
        StringBuffer mutableString = new StringBuffer("Hello, world!");


        // Append a new string to the mutable string
        mutableString.append(" This is a mutable string.");


        // Print the mutable string
        System.out.println(mutableString);
    }
}
```

Q2 ans-

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

```
 StringBuilder sb = new StringBuilder("PWSKILLS");
 sb.reverse();
 System.out.println(sb);
 }
}
```

Q3 ans-

```
import java.util.Scanner;

public class Assingnmenr {
 public static void main(String[] args)
 {
 String str1 = "THINK TWICE";
 char ch[] = str1.toCharArray();
 String str2 = "";
 str1 = str1.toLowerCase();

 String[] arr = str1.split(" ");

 for(String ans : arr)
 {
 for(int i = ans.length()-1;i>=0;i--)
 {
 str2 += ans.charAt(i);
 }
 str2 += " ";
 }
 System.out.println(str2);
 }
}
```

Q4 ans-

```
import java.util.Arrays;

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

```
 String s1 = "DCBA";
 s1 = s1.toUpperCase();
 char[] ch = s1.toCharArray();
 Arrays.sort(ch);
 System.out.println(ch);
 }
}
```