

Lab Assignment – 25

Name: M Bhagyalaxmi

Student id: AF0339449

1) Write a Java program that uses lambda expressions to manipulate strings. Create lambda expressions to perform the following operations on a given string:

- Convert the string to uppercase
- Convert the string to lowercase
- Reverse the string

```
package com.wrapperclass.examples;
import java.util.function.Function;
public class StringManipulation {
    public static void main(String[] args) {
        String inputString = "Hello world!";

        // Convert the string to uppercase
        String resultUppercase =
        manipulateString(inputString, s -> s.toUpperCase());
        System.out.println("Uppercase: " +
        resultUppercase);

        // Convert the string to lowercase
        String resultLowercase =
        manipulateString(inputString, s -> s.toLowerCase());
        System.out.println("Lowercase: " +
        resultLowercase);

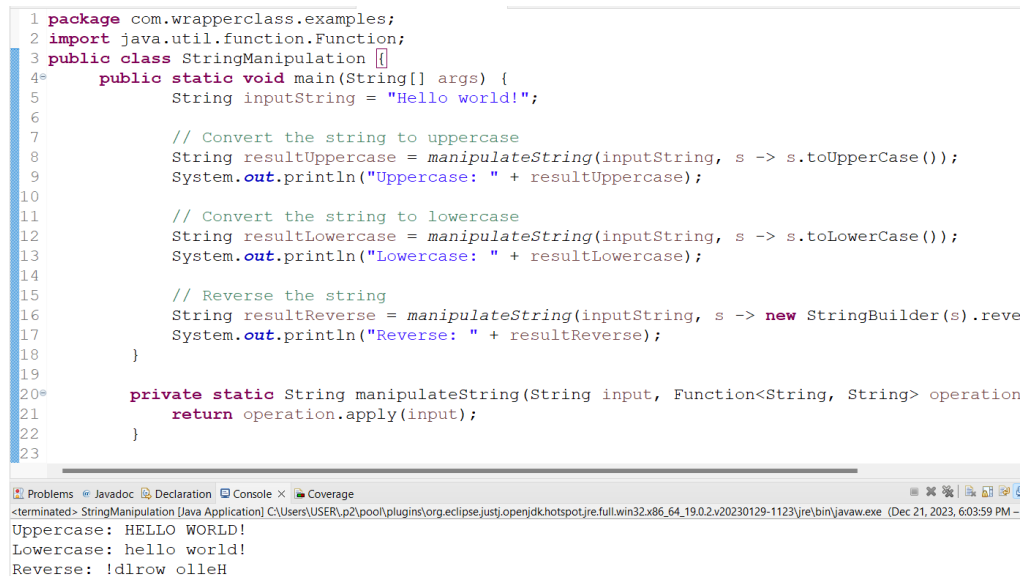
        // Reverse the string
        String resultReverse =
        manipulateString(inputString, s -> new
        StringBuilder(s).reverse().toString());
        System.out.println("Reverse: " +
        resultReverse);
    }
}
```

```

        private static String manipulateString(String
input, Function<String, String> operation) {
            return operation.apply(input);
        }
    }
}

```

Output:



```

1 package com.wrapperclass.examples;
2 import java.util.function.Function;
3 public class StringManipulation {
4     public static void main(String[] args) {
5         String inputString = "Hello world!";
6
7         // Convert the string to uppercase
8         String resultUppercase = manipulateString(inputString, s -> s.toUpperCase());
9         System.out.println("Uppercase: " + resultUppercase);
10
11        // Convert the string to lowercase
12        String resultLowercase = manipulateString(inputString, s -> s.toLowerCase());
13        System.out.println("Lowercase: " + resultLowercase);
14
15        // Reverse the string
16        String resultReverse = manipulateString(inputString, s -> new StringBuilder(s).reverse());
17        System.out.println("Reverse: " + resultReverse);
18    }
19
20    private static String manipulateString(String input, Function<String, String> operation)
21    {
22        return operation.apply(input);
23    }
24 }

```

Problems Javadoc Declaration Console X Coverage
<terminated> StringManipulation [Java Application] C:\Users\USER\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_19.0.2.v20230129-1123\jre\bin\javaw.exe (Dec 21, 2023, 6:03:59 PM -
Uppercase: HELLO WORLD!
Lowercase: hello world!
Reverse: !dlrow olleH

2) Write a Java program that demonstrates the use of method references for static methods. Create a functional interface and use a method reference to call a static method that calculates the square of a number.

```

package com.wrapperclass.examples;
interface SquareCalculator {
    int calculateSquare(int num);
}

```

```

// Utility class with a static method for calculating
the square
class MathUtils {

```

```

        static int square(int num) {
            return num * num;
        }
    }

    public class MethodReferenceExample {
        public static void main(String[] args) {
            // Using method reference to call the static
            method
            SquareCalculator squareCalculator =
            MathUtils::square;

            // Using the functional interface to calculate
            the square
            int result =
            squareCalculator.calculateSquare(5);

            System.out.println("Square of 5 is: " +
            result);
        }
    }

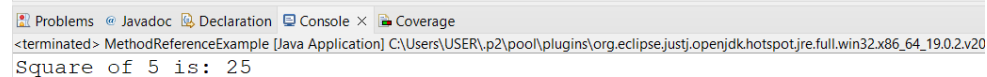
```

Output:

```

1 package com.wrapperclass.examples;
2 interface SquareCalculator {
3     int calculateSquare(int num);
4 }
5
6 // Utility class with a static method for calculating the square
7 class MathUtils {
8     static int square(int num) {
9         return num * num;
10    }
11 }
12 public class MethodReferenceExample {
13     public static void main(String[] args) {
14         // Using method reference to call the static method
15         SquareCalculator squareCalculator = MathUtils::square;
16
17         // Using the functional interface to calculate the square
18         int result = squareCalculator.calculateSquare(5);
19
20         System.out.println("Square of 5 is: " + result);
21     }
22 }
23 }

```


 The screenshot shows the Eclipse IDE interface. At the bottom, there is a console window with the output "Square of 5 is: 25". The console window title is "<terminated> MethodReferenceExample [Java Application] C:\Users\USER\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_19.0.2.v20".