AI Assisted Coding

Assignment – 1.5

Name : G.Samshray

Ht.no : 2303A51670

Batch: 23

Task – 1 Prompt : Use GitHub Copilot to generate a Java program that reverses a user-input string using inline logic without functions.

Code :

```
import java.util.Scanner;

public class ReverseString {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");

        String input = scanner.nextLine();

        String reversed = "";

        for (int i = input.length() - 1; i >= 0; i--) {

            reversed += input.charAt(i);

        }

        System.out.println("Reversed string: " + reversed);

    }

}
```

Output :

```
Enter a string: SR University

Reversed string: ytisrevinU RS
```

Task – 2 Prompt : Use GitHub Copilot to optimize the string reversal code by simplifying logic, reducing variables, and improving readability, and explain how the changes improve efficiency.

Code :

```
import java.util.Scanner;
public class ReverseString{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        StringBuilder reversed = new StringBuilder();
        for (int i = input.length() - 1; i >= 0; i--) {
            reversed.append(input.charAt(i));
        }
        System.out.println("Reversed string: " + reversed.toString());
    }
}
```

Output :

```
Enter a string: SR University
Reversed string: ytisrevinU RS
```

Task – 3 Prompt : Use GitHub Copilot to generate a function-based Java program that reverses a string, returns the result, and includes meaningful comments.

Code :

```
import java.util.Scanner;
public class ReverseString {
    public static String reverseString(String input) {
        StringBuilder reversed = new StringBuilder();
        for (int i = input.length() - 1; i >= 0; i--) {
            reversed.append(input.charAt(i));
        }
        return reversed.toString();
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        String reversed = reverseString(input);
        System.out.println("Reversed string: " + reversed);
    }
}
```

Output :

```
Enter a string: AI Assisted Coding
Reversed string: gnidoC detsissA IA
```

Task – 4 Prompt : Use GitHub Copilot to compare procedural and modular string reversal programs and analyze them based on clarity, reusability, debugging, and scalability.

Code :

```
import java.util.Scanner;
public class ReverseString {
    // Function to reverse a given string
    public static String reverseString(String input) {
        StringBuilder reversed = new StringBuilder();
        for (int i = input.length() - 1; i >= 0; i--) {
            reversed.append(input.charAt(i));
        }
        return reversed.toString();
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();
        // Call the reverseString function and store the result
        String reversed = reverseString(input);
        System.out.println("Reversed string: " + reversed);
    }
}
```

Output :

```
Enter a string: Samshray
Reversed string: yarhsmaS
```

Task – 5 Prompt : Use GitHub Copilot to generate loop-based and slicing-based string reversal methods and compare them based on execution flow, time complexity, performance, and suitability.

Code :

```
import java.util.Scanner;
public class ReverseString {
    // Loop-based method to reverse a string
    public static String reverseStringLoop(String input) {
        StringBuilder reversed = new StringBuilder();
        for (int i = input.length() - 1; i >= 0; i--) {
            reversed.append(input.charAt(i));
        }
        return reversed.toString();
    }

    // Slicing-based method to reverse a string
    public static String reverseStringSlicing(String input) {
        return new StringBuilder(input).reverse().toString();
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        // Using loop-based method
        String reversedLoop = reverseStringLoop(input);
        System.out.println("Reversed string (loop-based): " + reversedLoop);

        // Using slicing-based method
        String reversedSlicing = reverseStringSlicing(input);
        System.out.println("Reversed string (slicing-based): " + reversedSlicing);
    }
}
```

Output :

```
Enter a string: SR University
Reversed string (loop-based): ytisrevinU RS
Reversed string (slicing-based): ytisrevinU RS
```