slip21

```
import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;


public class s21q1 {

   public static void main(String[] args) {


      String filePath = "file5

.txt";


      try (Scanner scanner = new Scanner(new File(filePath))) {

         while (scanner.hasNextLine()) {


            String line = scanner.nextLine();



            String[] words = line.split("\\s+");



            for (String word : words) {

               System.out.print(reverseString(word) + " ");

            }

            System.out.println();

         }
```

```
        } catch (FileNotFoundException e) {
            System.out.println("File not found: " + e.getMessage());
        }
    }

    private static String reverseString(String word) {
        return new StringBuilder(word).reverse().toString();
    }
}
```

```
import java.util.Hashtable;
import java.util.Scanner;

public class s21q2 {
    public static void main(String[] args) {

        Hashtable<String, String> cc = new Hashtable<>();


        cc.put("Mumbai", "022");
        cc.put("Delhi", "011");
        cc.put("Bangalore", "080");
        cc.put("Kolkata", "033");
```

```
        System.out.println("City STD Codes:");
        for (String city : cc.keySet()) {
            System.out.println("City: " + city + ", STD Code: " + cc.get(city));
        }

        Scanner scanner = new Scanner(System.in);
        System.out.print("\nEnter the city name to search for its STD code: ");
        String searchCity = scanner.nextLine();

        String stdCode = cc.get(searchCity);
        if (stdCode != null) {
            System.out.println("The STD code for " + searchCity + " is: " + stdCode);
        } else {
            System.out.println("City not found in the hashtable.");
        }

        // Closing the scanner
        scanner.close();
    }
}
```

```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def area(self):
        return self.length * self.width

    def perimeter(self):
        return 2 * (self.length + self.width)


length = float(input("Enter the length of the rectangle: "))
width = float(input("Enter the width of the rectangle: "))

rectangle = Rectangle(length, width)

rectangle_area = rectangle.area()
rectangle_perimeter = rectangle.perimeter()


print(f"Rectangle with Length: {length} and Width: {width}")
print(f"Area = {rectangle_area:.2f}")
print(f"Perimeter = {rectangle_perimeter:.2f}")
```

```
otuple = (('333', '33'), ('1416', '55'))
```

```
ctuple = tuple(
    tuple(int(value) for value in inner_tuple) for inner_tuple in otuple
)

print("Original tuple values:", otuple)
print("New tuple values:", ctuple)
```