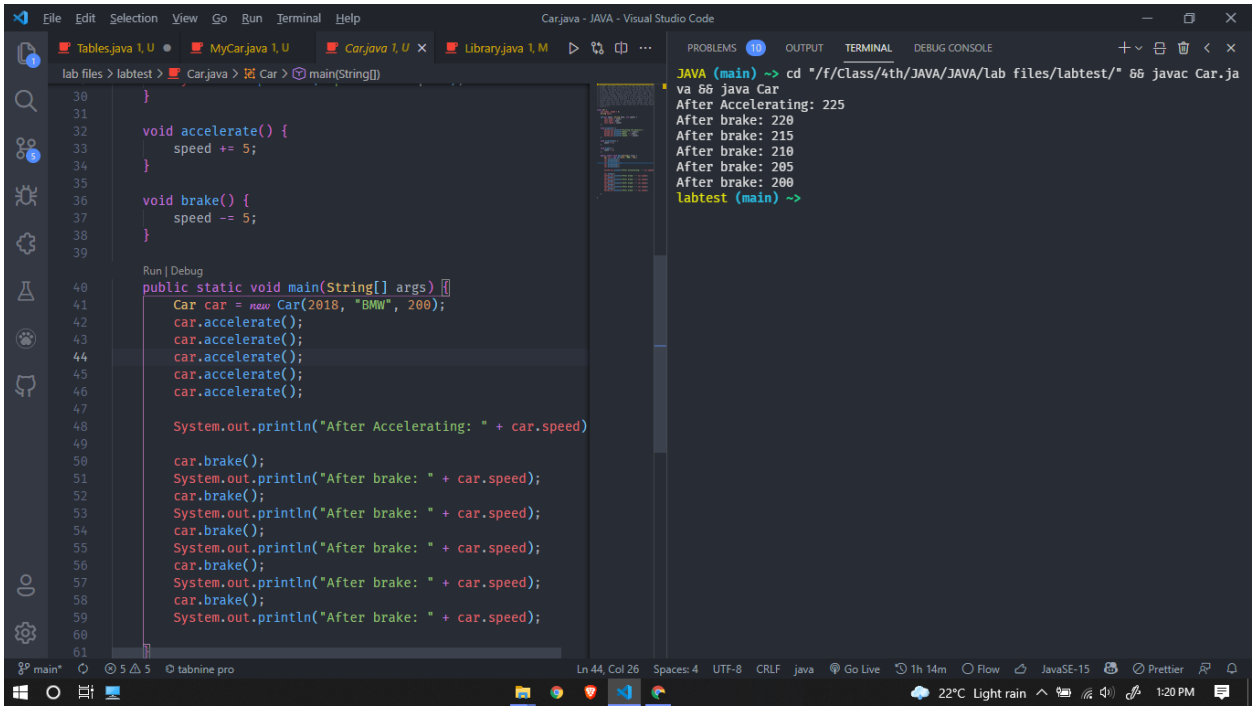
```
1b.Write a class named Car that has the following data members:
• model. The model field is an int that holds the car's year.
• make. The make field references a String object that holds the make of
the car.
• speed. The speed field is an int that holds the car's current speed.
The class should have the following constructor and other methods.
• The constructor should accept the car's year model, make and speed as
arguments.
• Accessor methods should get the values stored in an object's year,
Model, make, and speed fields.
• Accelerate method should add 5 to the speed field each time it is
called.
• Brake method should subtract 5 from the speed field each time it is
called.
Demonstrate the class in a program that creates a Car object, and then
calls the accelerate method five
times. After each call to the accelerate method, get the current speed of
the car and display it. Call the
brake method five times. After each call to the brake method, get the
current speed of the car and
display it.
```

Car.java

```
class Car {
    int model, speed = 0;
    String make;

    Car(int model, String make, int speed) {
        this.model = model;
        this.make = make;
        this.speed = speed;
    }

    void accessor() {
        System.out.println("Printing the details");
```

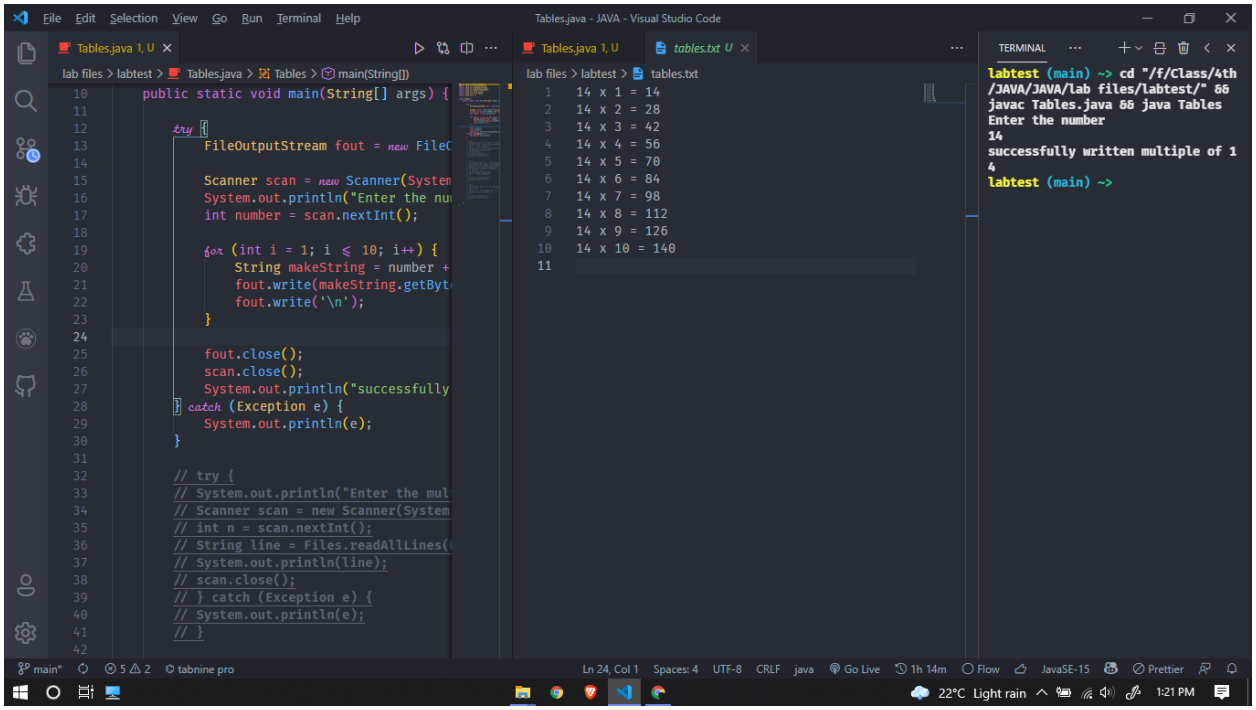
```
        System.out.println("Model: " + model);
        System.out.println("Make: " + make);
        System.out.println("Speed: " + speed);
    }

    void accelerate() {
        speed += 5;
    }

    void brake() {
        speed -= 5;
    }

    public static void main(String[] args) {
        Car car = new Car(2018, "BMW", 200);
        car.accelerate();
        car.accelerate();
        car.accelerate();
        car.accelerate();
        car.accelerate();

        System.out.println("After Accelerating: " + car.speed);

        car.brake();
        System.out.println("After brake: " + car.speed);
        car.brake();
        System.out.println("After brake: " + car.speed);
        car.brake();
        System.out.println("After brake: " + car.speed);
        car.brake();
        System.out.println("After brake: " + car.speed);
        car.brake();
        System.out.println("After brake: " + car.speed);

    }

}
```

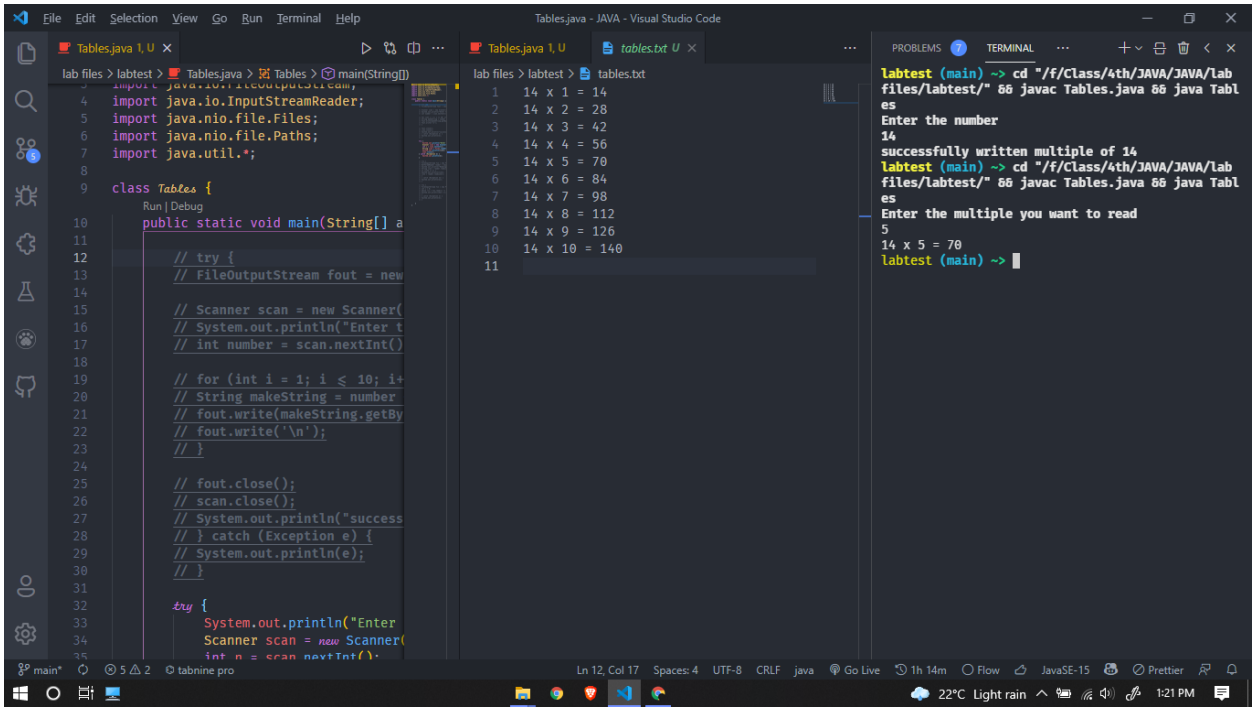


Modification.java

```java
import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.*;

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

        try {
            FileOutputStream fout = new FileOutputStream("tables.txt");

            Scanner scan = new Scanner(System.in);
            System.out.println("Enter the number");
            int number = scan.nextInt();

            for (int i = 1; i <= 10; i++) {
```

```
                String makeString = number + " x " + i + " = " + (number *
i);
                fout.write(makeString.getBytes());
                fout.write('\n');
            }

            fout.close();
            scan.close();
            System.out.println("successfully written multiple of " +
number);
        } catch (Exception e) {
            System.out.println(e);
        }

        try {
            System.out.println("Enter the multiple you want to read ");
            Scanner scan = new Scanner(System.in);
            int n = scan.nextInt();
            String line =
Files.readAllLines(Paths.get("tables.txt")).get(n - 1);
            System.out.println(line);
            scan.close();
        } catch (Exception e) {
            System.out.println(e);
        }

    }
}
```


File Edit Selection View Go Run Terminal Help
Tables.java - JAVA - Visual Studio Code
Tables.java 1, U
tables.txt U
lab files > labtest > Tables.java > Tables > main(String[])
lab files > labtest > tables.txt
TERMINAL
labtest (main) ~> cd "/f/Class/4th/JAVA/JAVA/lab files/labtest/" && javac Tables.java && java Tables
Enter the number
14
successfully written multiple of 14
labtest (main) ~>
14 x 1 = 14
14 x 2 = 28
14 x 3 = 42
14 x 4 = 56
14 x 5 = 70
14 x 6 = 84
14 x 7 = 98
14 x 8 = 112
14 x 9 = 126
14 x 10 = 140
Ln 24, Col 1
Spaces: 4
UTF-8
CRLF
java
22°C Light rain
1:21 PM


File Edit Selection View Go Run Terminal Help
Tables.java - JAVA - Visual Studio Code
Tables.java 1, U
tables.txt U
lab files > labtest > Tables.java > Tables > main(String[])
lab files > labtest > tables.txt
PROBLEMS 7
TERMINAL
labtest (main) ~> cd "/f/Class/4th/JAVA/JAVA/lab files/labtest/" && javac Tables.java && java Tables
Enter the number
14
successfully written multiple of 14
labtest (main) ~> cd "/f/Class/4th/JAVA/JAVA/lab files/labtest/" && javac Tables.java && java Tables
Enter the multiple you want to read
5
14 x 5 = 70
labtest (main) ~>
14 x 1 = 14
14 x 2 = 28
14 x 3 = 42
14 x 4 = 56
14 x 5 = 70
14 x 6 = 84
14 x 7 = 98
14 x 8 = 112
14 x 9 = 126
14 x 10 = 140
Ln 12, Col 17
Spaces: 4
UTF-8
CRLF
java
22°C Light rain
1:21 PM