

**(DIGITAL ASSIGNMENT - 5) SYNCHRONIZTION,INTER THREADING,FILES,COLLECTION, JAVAFX AND JDBC**

**CSE1007(JAVA PROGRAMMING)LAB:L31-L32**



**April 18, 2022**

**ANIRUDH VADERA**

**20BCE2940**

**ACTIVITY – 7(INTER THREADING)**

**QUESTION 1:**

****

**CODE:**

package activity7;

class Array {

    static int[] a = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

    public static synchronized void print(String odd\_even) {

        if (odd\_even.equals("ODD")) {

            System.out.println("Printing Odd Numbers");

            System.out.print("[ ");

            for (int i = 0; i < a.length; i++) {

                if (a[i] % 2 != 0) {

                    System.out.print(a[i] + " ");

                }

            }

            System.out.println("]");

        } else {

            System.out.println("Printing Even Numbers");

            System.out.print("[ ");

            for (int i = 0; i < a.length; i++) {

                if (a[i] % 2 == 0) {

                    System.out.print(a[i] + " ");

                }

            }

            System.out.println("]");

        }

    }

}

public class activity7q1 {

    public static void main(String[] args) throws InterruptedException {

        System.out.println("ANIRUDH VADERA(20BCE2940)");

        Array a = new Array();

        Thread t1 = new Thread(() -> {

            System.out.println("Thread 1 - ODD");

            System.out.println(Thread.currentThread().getName());

            Array.print("ODD");

        });

        Thread t2 = new Thread(() -> {

            System.out.println("Thread 2 - EVEN");

            System.out.println(Thread.currentThread().getName());

            Array.print("EVEN");

        });

        t1.start();

        t2.start();

        t1.join();

        t2.join();

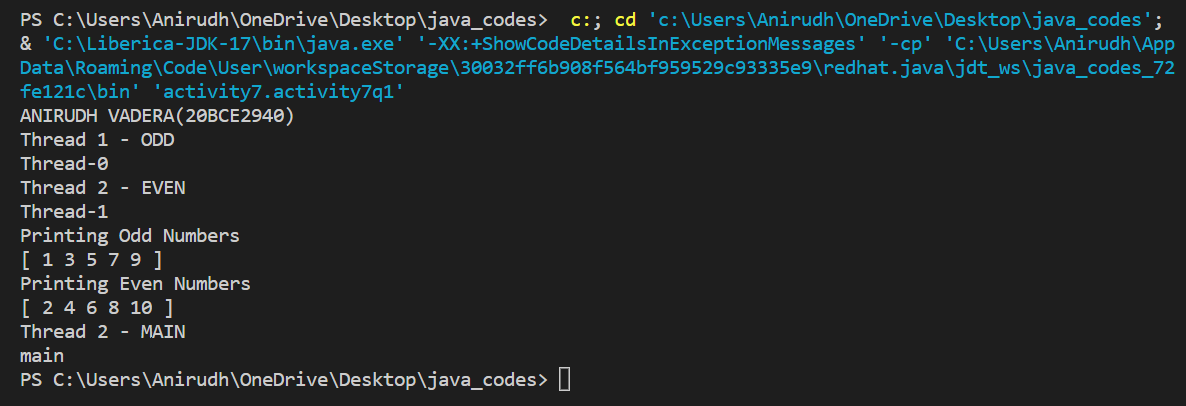
        System.out.println("Thread 2 - MAIN");

        System.out.println(Thread.currentThread().getName());

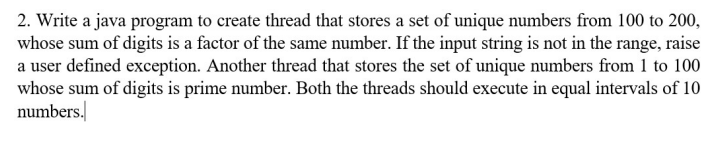
    }

}

**OUTPUT:**

****

**QUESTION 2:**

****

**CODE:**

package activity7;

import java.util.Scanner;

class invalidInputNotRange1 extends Exception {

    public invalidInputNotRange1(String message) {

        super(message);

    }

}

class invalidInputNotRange2 extends Exception {

    public invalidInputNotRange2(String message) {

        super(message);

    }

}

class Thread1 implements Runnable {

    int[] number;

    int range1\_low;

    int range1\_high;

    int sum = 0;

    int[] array\_factor = new int[100];

    int factor\_elements = 0;

    int[] unique = new int[100];

    int unique\_elements = 0;

    int start;

    int end;

    Thread1(int[] number, int start, int end, int range1\_low, int range1\_high) {

        this.number = number;

        this.range1\_high = range1\_high;

        this.range1\_low = range1\_low;

        this.start = start;

        this.end = end;

    }

    void validate1(int input) throws invalidInputNotRange1 {

        if (input >= range1\_low && input <= range1\_high) {

        } else {

            throw new invalidInputNotRange1(

                    "The Input (" + input + ") is not in the Specified Range of " + "[" +

                            range1\_low + "," +

                            range1\_high + "]");

        }

    }

    public void generate() {

        for (int number = range1\_low; number < (range1\_high + 1); number++) {

            sum = 0;

            int temp = number;

            while (temp != 0) {

                int c = temp % 10;

                sum = sum + c;

                temp = temp / 10;

            }

            if (number % sum == 0) {

                array\_factor[factor\_elements++] = number;

            }

        }

    }

    public void run() {

        generate();

        System.out.println("Thread 1 - Numbers whose sum of digits is a factor ofnumber itself");

        for (int j = start; j < end; j++) {

            try {

                validate1(number[j]);

                int flag = 0;

                for (int i = 0; i < factor\_elements; i++) {

                    if (number[j] == array\_factor[i]) {

                        flag = 1;

                        break;

                    }

                }

                if (flag == 1) {

                    unique[unique\_elements++] = number[j];

                }

            } catch (invalidInputNotRange1 e) {

                System.out.println(e);

            }

        }

        print();

    }

    public void print() {

        System.out.println("The Numbers that satisfies the conditions are : ");

        System.out.println("Printing Numbers whose sum of digits is a factor ofnumber itself");

        System.out.print("[ ");

        for (int i = 0; i < unique\_elements; i++) {

            System.out.print(unique[i] + " ");

        }

        System.out.println("]");

    }

}

class Thread2 implements Runnable {

    int[] number;

    int range2\_low;

    int range2\_high;

    int sum = 0;

    int[] array\_factor = new int[100];

    int factor\_elements = 0;

    int[] unique = new int[100];

    int unique\_elements = 0;

    int start;

    int end;

    Thread2(int[] number, int start, int end, int range2\_low, int range2\_high) {

        this.number = number;

        this.range2\_high = range2\_high;

        this.range2\_low = range2\_low;

        this.start = start;

        this.end = end;

    }

    void validate2(int input) throws invalidInputNotRange2 {

        if (input >= range2\_low && input <= range2\_high) {

        } else {

            throw new invalidInputNotRange2(

                    "The Input (" + input + ") is not in the Specified Range of " + "[" + range2\_low + "," + range2\_high

                            + "]");

        }

    }

    public void generate() {

        for (int number = range2\_low; number < (range2\_high + 1); number++) {

            sum = 0;

            int temp = number;

            while (temp != 0) {

                int c = temp % 10;

                sum = sum + c;

                temp = temp / 10;

            }

            int flag = 0;

            for (int i = 2; i < sum; i++) {

                if (sum % i == 0) {

                    flag = 1;

                    break;

                }

            }

            if (flag == 0) {

                array\_factor[factor\_elements++] = number;

            }

        }

    }

    public void run() {

        generate();

        System.out.println("Thread 2 - Numbers whose sum of digits is a PrimeNumber");

        for (int j = start; j < end; j++) {

            try {

                validate2(number[j]);

                int flag = 0;

                for (int i = 0; i < factor\_elements; i++) {

                    if (number[j] == array\_factor[i]) {

                        flag = 1;

                        break;

                    }

                }

                if (flag == 1) {

                    unique[unique\_elements++] = number[j];

                }

            } catch (invalidInputNotRange2 e) {

                System.out.println(e);

            }

        }

        print();

    }

    public void print() {

        System.out.println("The Numbers that satisfies the conditions are : ");

        System.out.println("Printing Numbers whose sum of digits is a Prime Number");

        System.out.print("[ ");

        for (int i = 0; i < unique\_elements; i++) {

            System.out.print(unique[i] + " ");

        }

        System.out.println("]");

    }

}

public class activity7q2 {

    static int range1\_low;

    static int range1\_high;

    static int range2\_low;

    static int range2\_high;

    public static void main(String[] args) throws InterruptedException {

        System.out.println("ANIRUDH VADERA(20BCE2940)");

        Scanner in = new Scanner(System.in);

        System.out.println("Enter the range1\_low : ");

        activity7q2.range1\_low = in.nextInt();

        System.out.println("Enter the range1\_high : ");

        activity7q2.range1\_high = in.nextInt();

        System.out.println("Enter the range2\_low : ");

        activity7q2.range2\_low = in.nextInt();

        System.out.println("Enter the range2\_high : ");

        activity7q2.range2\_high = in.nextInt();

        System.out.println("Enter The set of Number Strings to be checked(Inmultiples of 10) : ");

        int n = in.nextInt();

        n = n \* 10;

        int[] set = new int[n];

        for (int i = 0; i < (n); i++) {

            set[i] = (i + 1) \* 10;

        }

        for (int i = 0; i < n / 10; i++) {

            int start = (i \* 10);

            int end = (i \* 10) + 10;

            Thread1 factor = new Thread1(set, start, end, range1\_low, range1\_high);

            Thread t1 = new Thread(factor);

            Thread2 prime = new Thread2(set, start, end, activity7q2.range2\_low,

                    range2\_high);

            Thread t2 = new Thread(prime);

            t1.start();

            t1.join();

            t2.start();

            t2.join();

        }

        System.out.println("Thread 2 - MAIN");

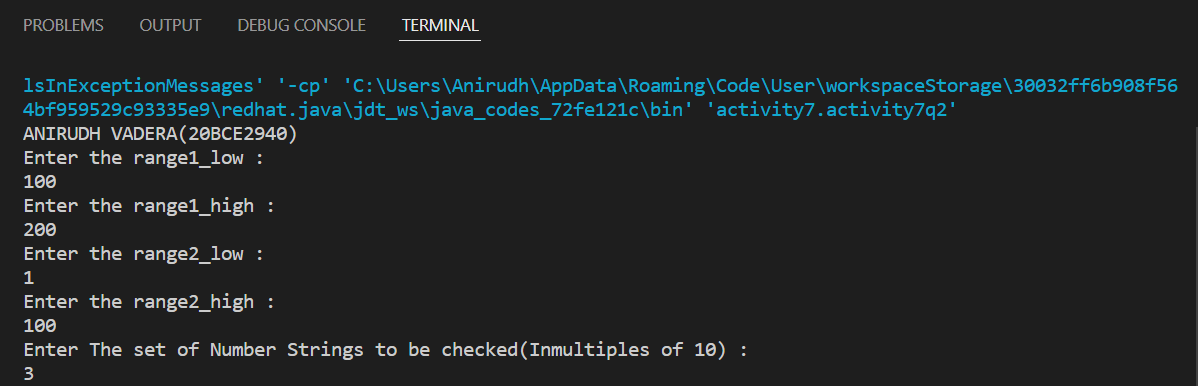
        System.out.println(Thread.currentThread().getName());

        in.close();

    }

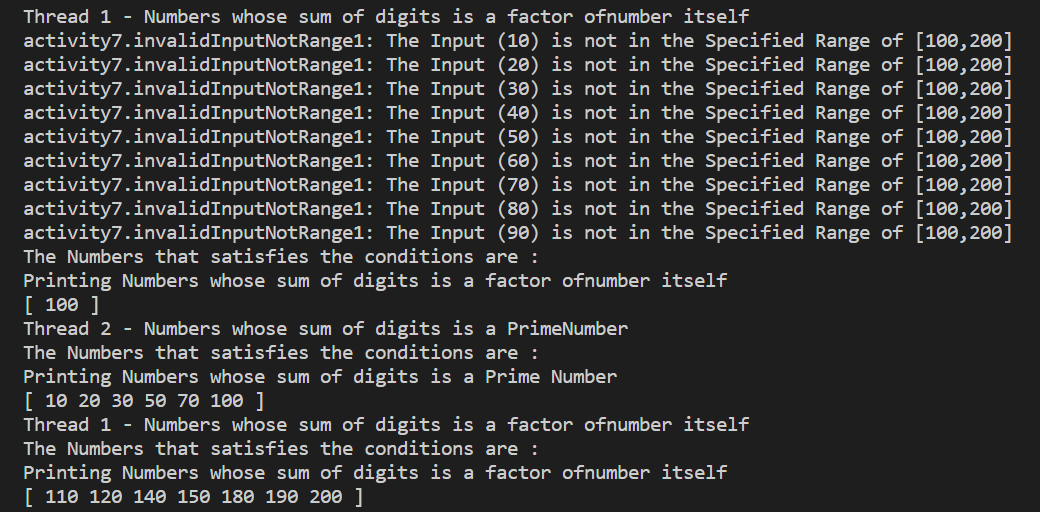
}

**OUTPUT:**

****

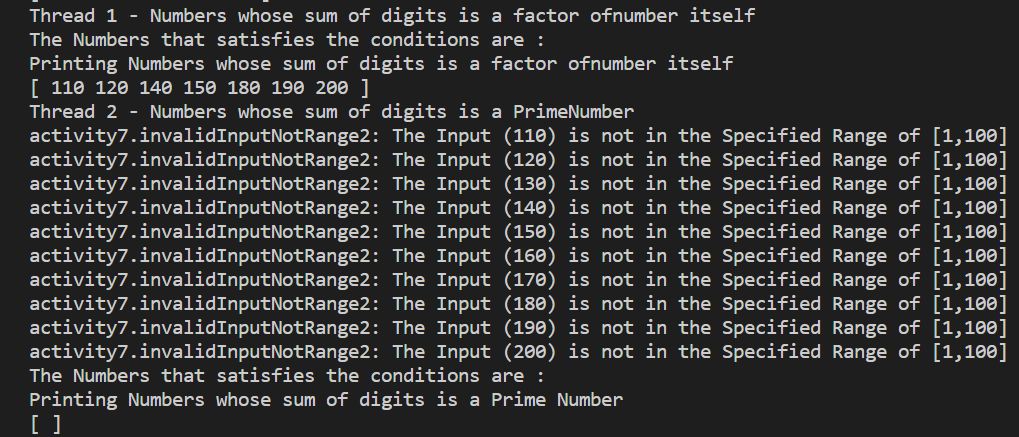
**FOR NUMBER 1-100**

**OUTPUT FOR THREAD 1 AND 2:**

****

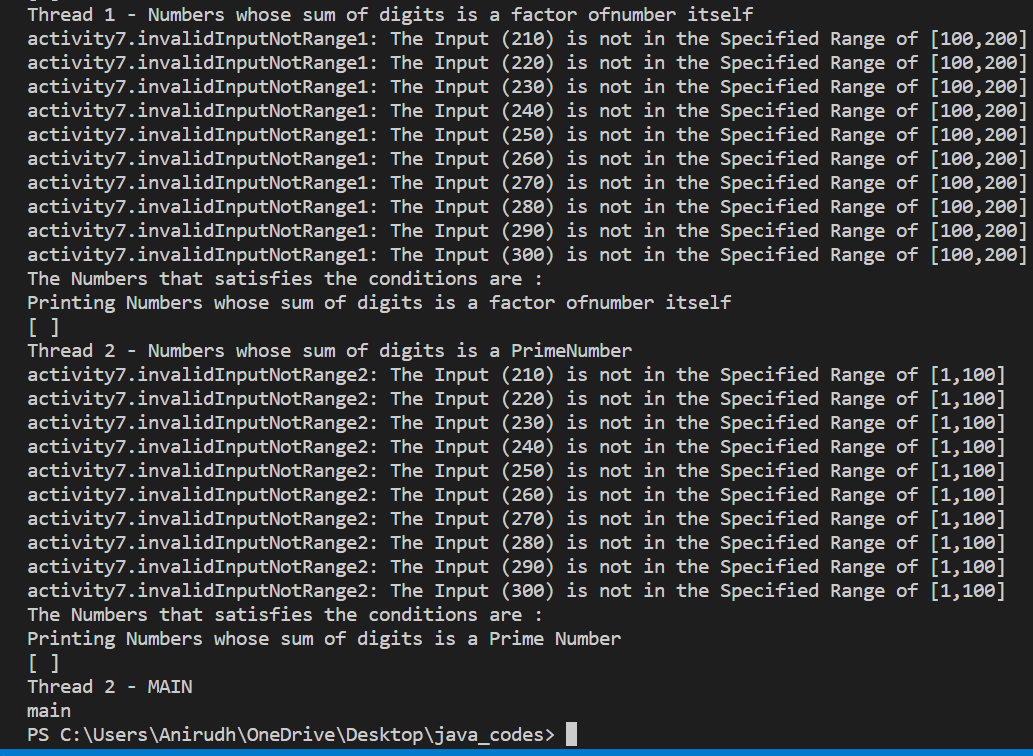
**FOR NUMBER 101-200**

**OUTPUT FOR THREAD 1 AND 2:**

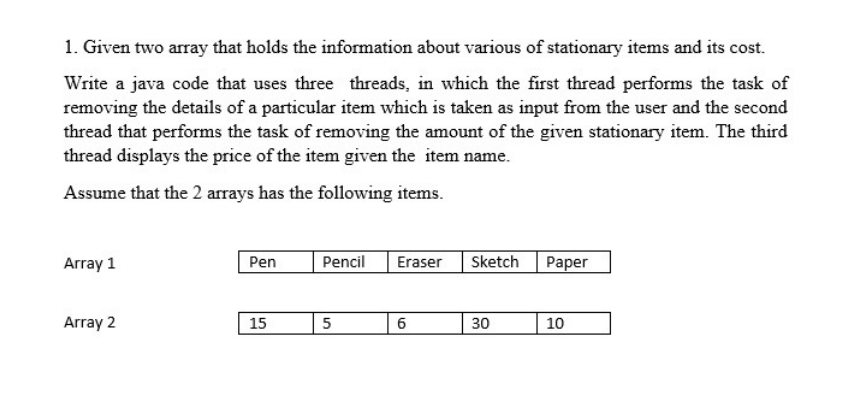
****

**FOR NUMBER 201-300**

**OUTPUT FOR THREAD 1 AND 2:**

****

**ACTIVITY – 8(SYNCHRONIZATION)**

****

**QUESTION 1:**

**CODE:**

package activity8;

import java.util.Scanner;

class notInRange extends Exception {

    public notInRange(String message) {

        super(message);

    }

}

class Items {

    String[] stationary\_items;

    int[] price;

    String[] itemsPriceRemoved = new String[100];

    int counter\_removed = 0;

    public void validate(String input) throws notInRange {

        int flag = 0;

        for (int i = 0; i < itemsPriceRemoved.length; i++) {

            if (input.equals(itemsPriceRemoved[i])) {

                flag = 1;

                break;

            }

        }

        if (flag == 1) {

            throw new notInRange("This Item price is not available in the price List : ");

        }

    }

    Items(String[] stationary\_items, int[] price, int[] link, int end) {

        this.stationary\_items = stationary\_items;

        this.price = price;

    }

    public synchronized void removeDetails(String itemName) {

        int to\_remove = 0;

        for (int i = 0; i < stationary\_items.length; i++) {

            if (stationary\_items[i].equals(itemName)) {

                to\_remove = i;

                break;

            }

        }

        stationary\_items[to\_remove] = "NULL";

        price[to\_remove] = -1;

    }

    public synchronized void removePrice(String itemName) {

        int to\_remove = 0;

        for (int i = 0; i < stationary\_items.length; i++) {

            if (stationary\_items[i].equals(itemName)) {

                to\_remove = i;

                break;

            }

        }

        itemsPriceRemoved[counter\_removed++] = stationary\_items[to\_remove];

        price[to\_remove] = -1;

    }

    public synchronized void Display(String itemName) {

        int to\_display = 0;

        int flag\_t = 0;

        for (int i = 0; i < stationary\_items.length; i++) {

            if (stationary\_items[i].equals(itemName)) {

                to\_display = i;

                flag\_t = 1;

                break;

            }

        }

        if (flag\_t == 0) {

            System.out.println("The item doesnt exist or is deleted from the list : ");

        } else {

            System.out.println("The Item Name is : " + stationary\_items[to\_display]);

            try {

                validate(stationary\_items[to\_display]);

                System.out.println("The Item Price is : " + price[to\_display]);

            } catch (notInRange e) {

                e.printStackTrace();

            }

        }

    }

}

public class activity8q1 {

    public static void main(String[] args) throws InterruptedException {

        Scanner in = new Scanner(System.in);

        System.out.println("ANIRUDH VADERA(20BCE2940)");

        System.out.println("Enter the Number of Items : ");

        int n = in.nextInt();

        String[] stationary\_items = new String[n];

        int[] price = new int[n];

        int[] link = new int[n];

        for (int i = 0; i < n; i++) {

            link[i] = i;

        }

        System.out.println("Enter the initial Item Details : ");

        for (int i = 0; i < n; i++) {

            System.out.println("Item : " + (i + 1));

            System.out.print("Item Name : ");

            in.nextLine();

            stationary\_items[i] = in.nextLine();

            System.out.print("Item Price : ");

            price[i] = in.nextInt();

        }

        in.nextLine();

        System.out.print("Item Name to be removed : ");

        String item1 = in.nextLine();

        Items items = new Items(stationary\_items, price, link, n);

        Thread t1 = new Thread(() -> {

            items.removeDetails(item1);

        });

        System.out.print("Item Price to be removed : ");

        String item2 = in.nextLine();

        Thread t2 = new Thread(() -> {

            items.removePrice(item2);

        });

        System.out.print("Item to be Displayed : ");

        String item3 = in.nextLine();

        Thread t3 = new Thread(() -> {

            try {

                Thread.sleep(500);

            } catch (InterruptedException e) {

                System.out.println(e);

            }

            items.Display(item3);

        });

        t1.start();

        t2.start();

        t3.start();

        t1.join();

        t2.join();

        t3.join();

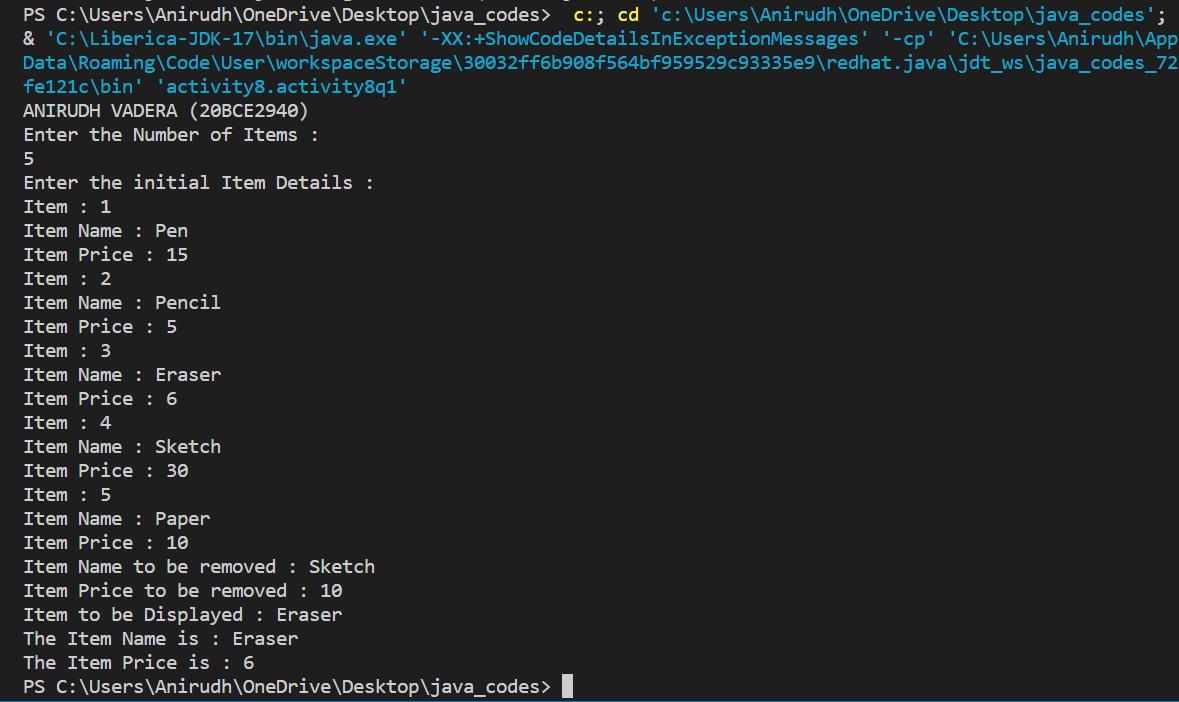
        in.close();

    }

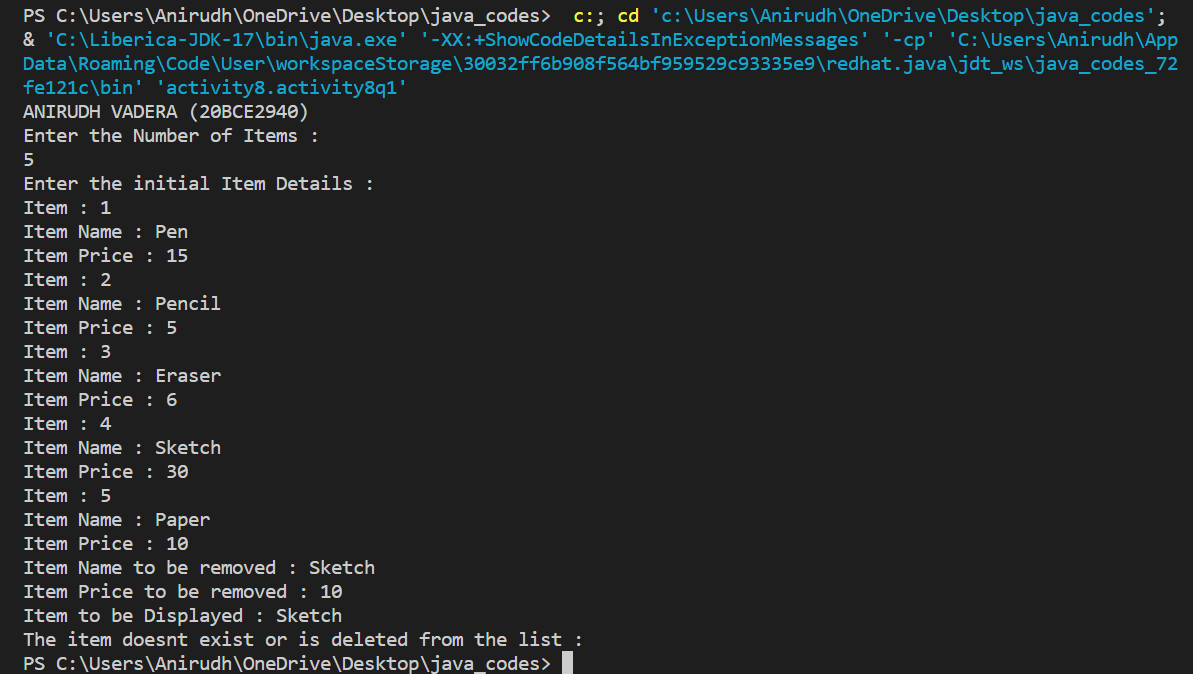
}

**OUTPUT:**

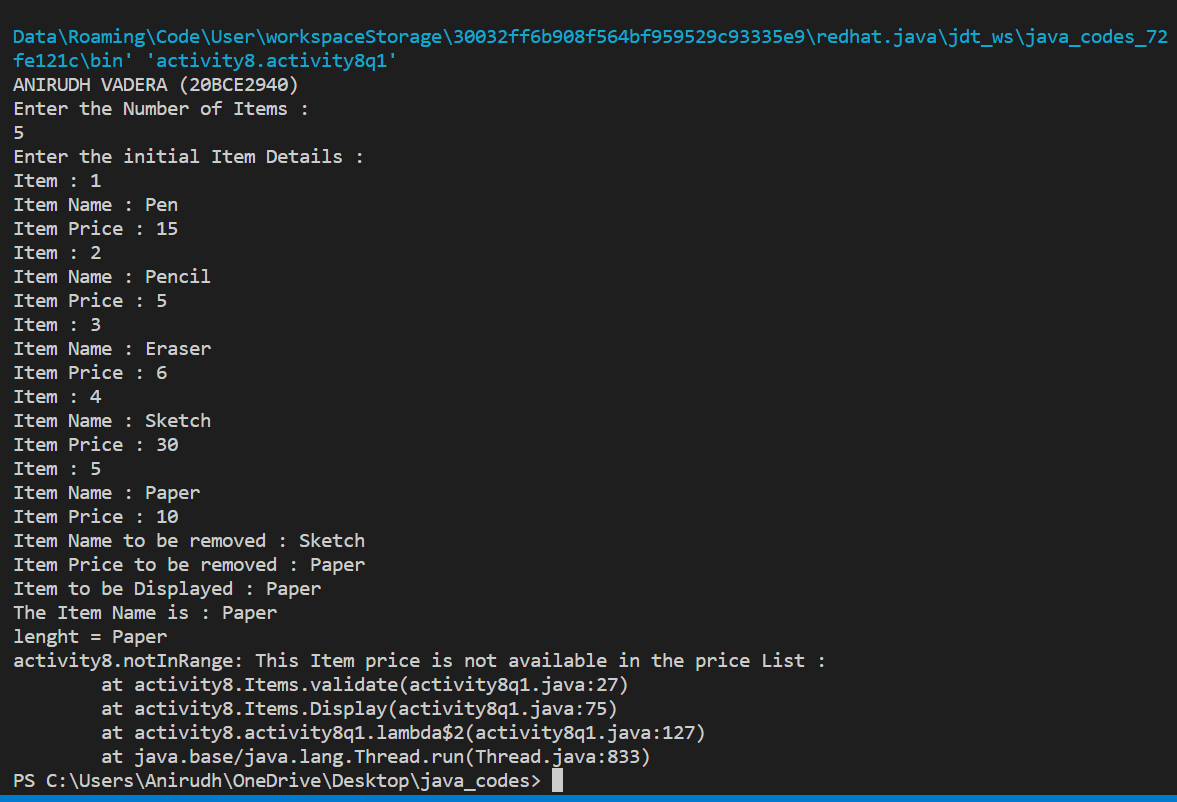
**When item name is available:**

****

**When item name is not available:**

****

**When item price is not available:**

****

**QUESTION 2:**

****

**CODE:**

package activity8;

class BusTicketRegistration {

    String BusName;

    int noOfSeats;

    BusTicketRegistration(String BusName, int noOfSeats) {

        this.BusName = BusName;

        this.noOfSeats = noOfSeats;

    }

    public synchronized void Register\_Seat() {

        while (noOfSeats <= 0) {

            System.out.println("Waiting.... " + Thread.currentThread().getId());

            try {

                wait();

            } catch (Exception e) {

                e.printStackTrace();

            }

        }

        noOfSeats = noOfSeats - 1;

    }

    public synchronized void Allot\_Seat(int seatsToAllocate) {

        System.out.println("ALLOCATING SEATS.....");

        noOfSeats = noOfSeats + seatsToAllocate;

        notifyAll();

    }

}

public class activity8q2 {

    public static void main(String[] args) throws InterruptedException {

        System.out.println("ANIRUDH VADERA (20BCE2940)");

        BusTicketRegistration registration = new BusTicketRegistration("Red Bus", 0);

        Thread t1 = new Thread(() -> {

            registration.Register\_Seat();

        });

        Thread t2 = new Thread(() -> {

            registration.Register\_Seat();

        });

        Thread t3 = new Thread(() -> {

            registration.Allot\_Seat(60);

        });

        Thread t4 = new Thread(() -> {

            try {

                Thread.sleep(500);

            } catch (InterruptedException e) {

                System.out.println(e);

            }

            System.out.println("The total Seats after registration is : " + registration.noOfSeats);

        });

        t1.start();

        t2.start();

        t3.start();

        t4.start();

        t1.join();

        t2.join();

        t3.join();

        t4.join();

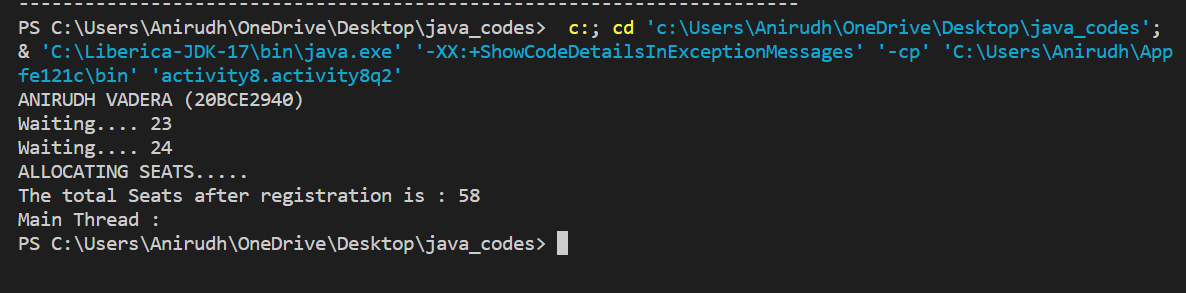
        System.out.println("Main Thread :");

    }

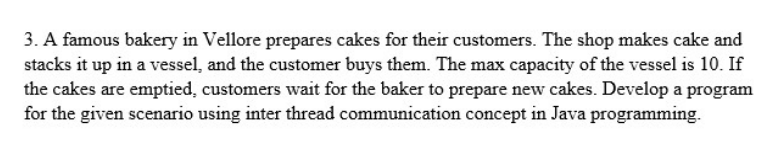
}

**OUTPUT:**

**First the two threads wait for allocation of the seats and then they execute.**

****

**QUESTION 3:**

****

**CODE:**

package activity8;

class Cake {

    int buffer;

    int capicity;

    Cake(int capicity, int buffer) {

        this.capicity = capicity;

        this.buffer = buffer;

    }

    public synchronized void produceCake(int piece) {

        while ((piece + buffer) > capicity) {

            try {

                wait();

            } catch (InterruptedException e) {

                e.printStackTrace();

            }

            System.out.println("Allowing Producer to consume more cakes : ");

        }

        System.out.println("Producing....Number : " + piece);

        buffer = buffer + piece;

        notifyAll();

    };

    public synchronized void consumeCake(int piece) {

        while (piece > buffer) {

            System.out.println("Waiting...To Produce Cakes...");

            notifyAll();

            try {

                wait();

            } catch (Exception e) {

                e.printStackTrace();

            }

        }

        buffer = buffer - piece;

        System.out.println("Consuming.....Number : " + piece);

    };

}

public class activity8q3 {

    public static void main(String[] args) throws InterruptedException {

        System.out.println("ANIRUDH VADERA (20BCE2940)");

        Cake c = new Cake(10, 8);

        Thread t1 = new Thread(() -> {

            for (int i = 0; i < 20; i++) {

                c.consumeCake(1);

            }

        });

        Thread t2 = new Thread(() -> {

            for (int i = 0; i < 10; i++) {

                c.produceCake(2);

            }

        });

        t1.start();

        t2.start();

    }

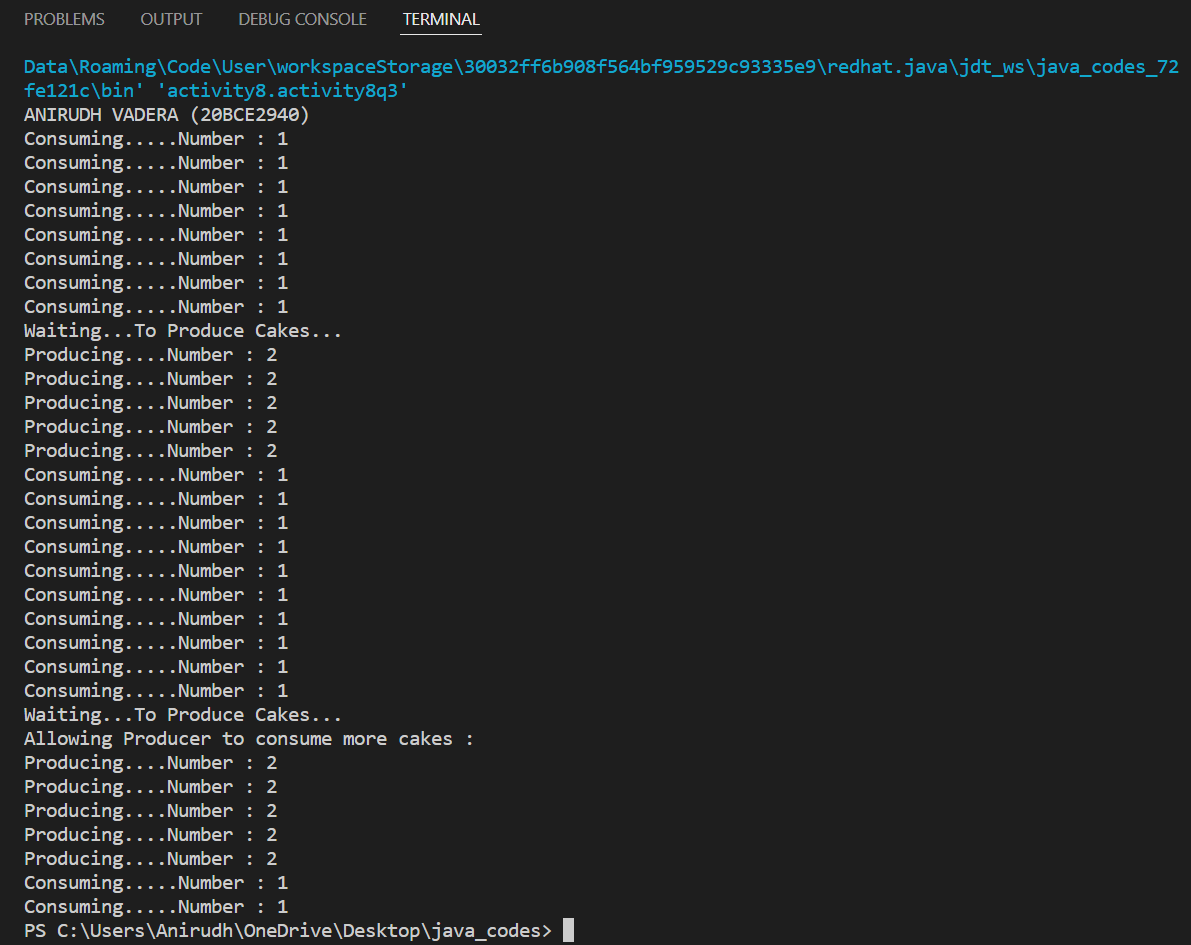
}

**OUTPUT:**

**First the consumer thread consumes the cakes**

**Then after the buffer is 0 it waits for more cakes to be produced**

**After enough cakes has been produced it starts consuming again**

****

**ACTIVITY – 9:(FILES)**

**QUESTION 1:**

**C:\Users\Anirudh\OneDrive\Pictures\Screenshots\Screenshot (3665).png**

**CODE:**

package activity9;

import java.io.FileInputStream;

import java.io.FileOutputStream;

public class activity9q1 {

    public static void main(String[] args) {

        System.out.println("ANIRUDH VADERA (20BCE2940)");

        String Content = new String("");

        int i;

        int[] Content\_in\_integer = new int[10000];

        int end = 0;

        // Reading the initial Content

        try {

            FileInputStream fin = new FileInputStream(

                    "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q1.txt");

            while ((i = fin.read()) != -1) {

                Content\_in\_integer[end++] = i;

                Content = Content + ((char) i);

            }

            System.out.println("The Contents of the File is : " + Content);

            fin.close();

        } catch (Exception e) {

            e.printStackTrace();

        }

        // Writing the reversed Content in the file

        try {

            FileOutputStream fout = new FileOutputStream(

                    "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q1.txt");

            for (int j = (end - 1); j >= 0; j--) {

                fout.write(Content\_in\_integer[j]);

            }

            fout.close();

            System.out.println("Success... Written the reversed Content : ");

        } catch (Exception e) {

            System.out.println(e);

        }

        // Reading the Final Content

        try {

            FileInputStream fin = new FileInputStream(

                    "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q1.txt");

            Content = "";

            end = 0;

            while ((i = fin.read()) != -1) {

                Content\_in\_integer[end++] = i;

                Content = Content + ((char) i);

            }

            System.out.println("The Contents of the File after reversing is : " + Content);

            fin.close();

        } catch (Exception e) {

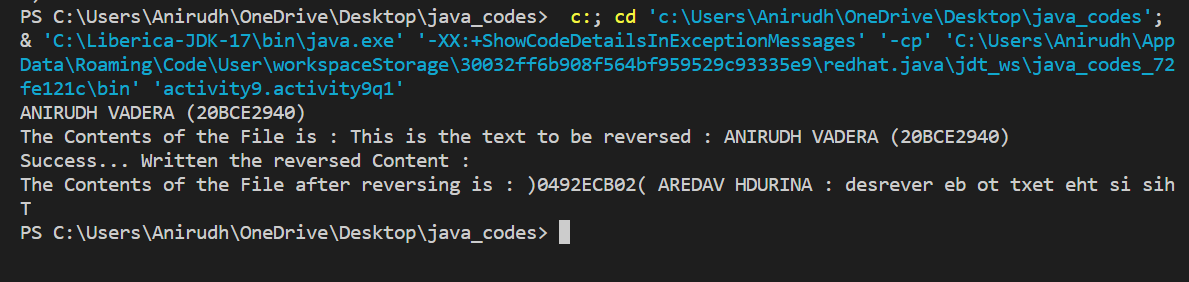
            e.printStackTrace();

        }

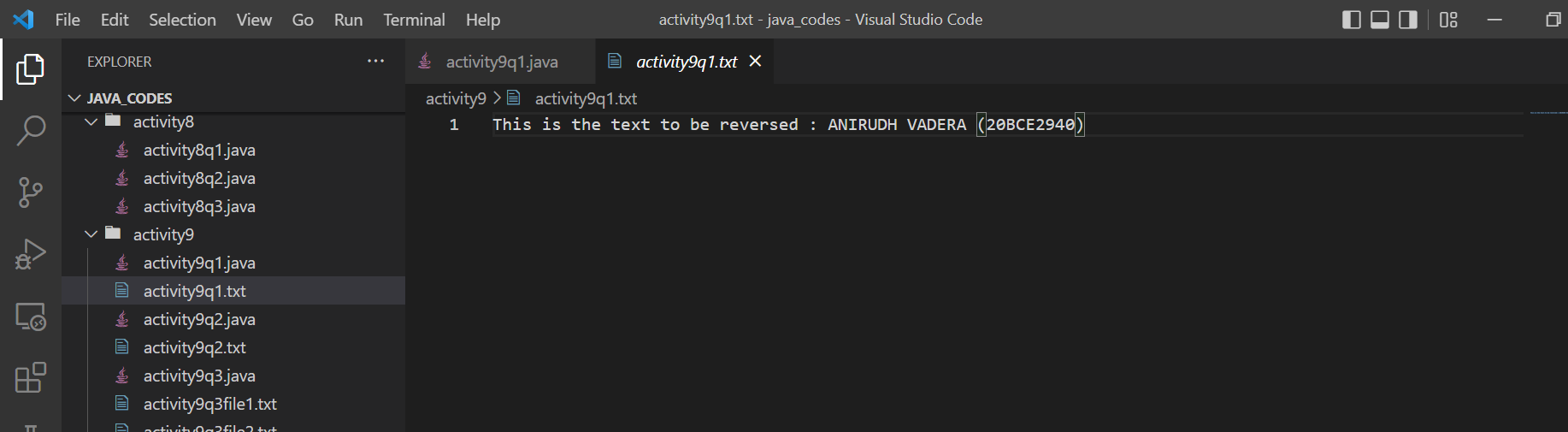
    }

}

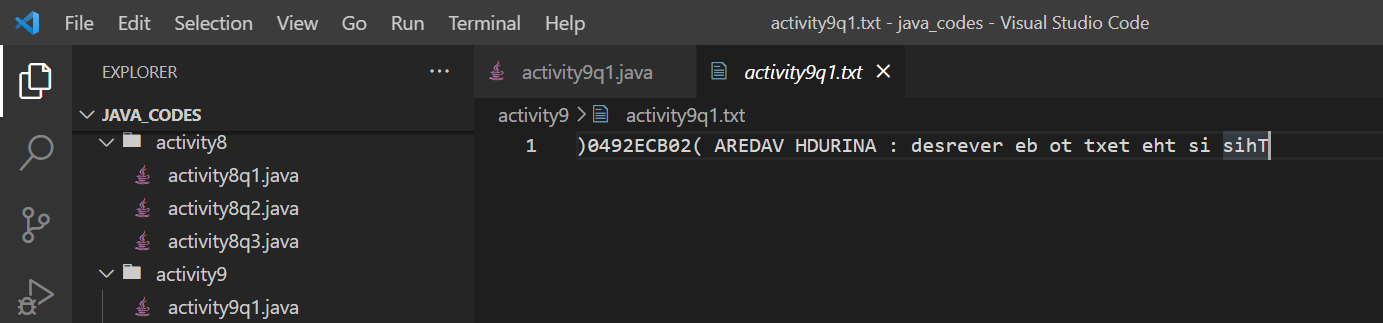
**OUTPUT:**

****

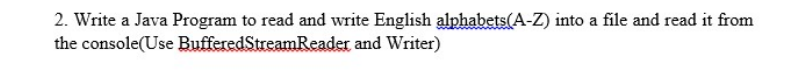
**Text file previously:**

****

**Text file after reversing:**

****

**QUESTION 2:**

****

**CODE:**

package activity9;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileWriter;

import java.io.InputStreamReader;

public class activity9q2 {

    public static void main(String[] args) {

        System.out.println("ANIRUDH VADERA (20BCE2940)");

        try (FileWriter writer = new FileWriter(

                "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q2.txt")) {

            BufferedWriter buffer = new BufferedWriter(writer);

            BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));

            String towrite = reader.readLine();

            buffer.write(towrite);

            System.out.println("Succes in writing the alphabets...");

            buffer.flush();

            buffer.close();

        } catch (

        Exception e) {

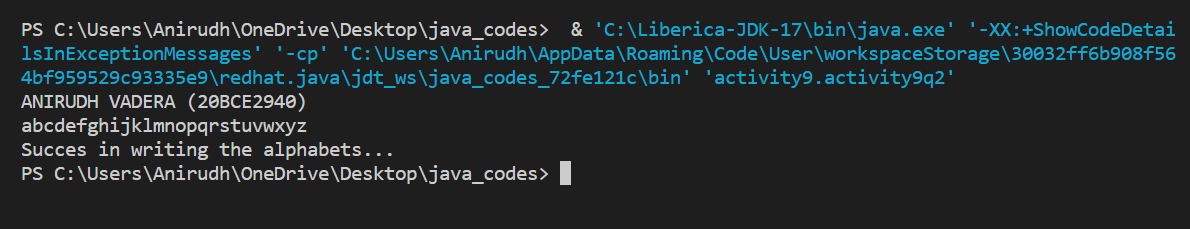
            e.printStackTrace();

        }

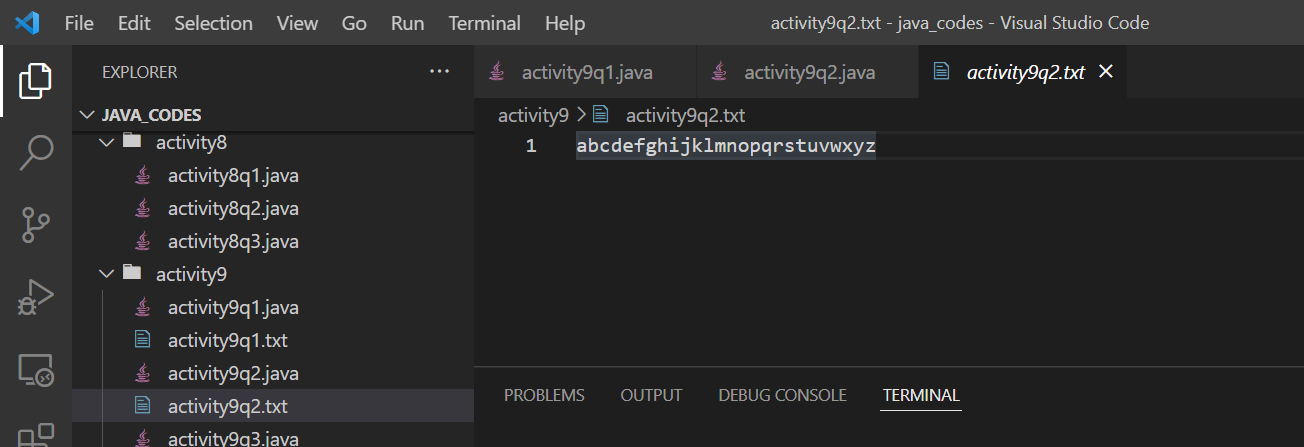
    }

}

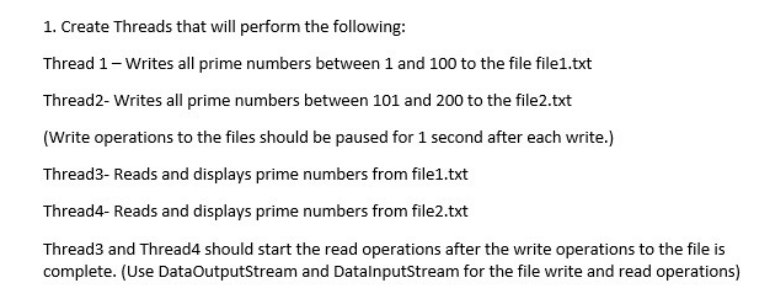
**OUTPUT:**

****

**The output File:**

****

**QUESTION 3:**

****

**CODE:**

package activity9;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

class WriteThread implements Runnable {

    String filePath;

    int flag = 0;

    DataOutputStream dout = null;

    int start;

    int end;

    int file;

    WriteThread(String filePath, int start, int end, int file) {

        this.filePath = filePath;

        this.start = start;

        this.end = end;

        this.file = file;

    }

    public void run() {

        synchronized (WriteThread.class) {

            try {

                dout = new DataOutputStream(new FileOutputStream(filePath));

            } catch (FileNotFoundException e) {

                e.printStackTrace();

            }

            for (int i = start; i < end; i++) {

                flag = 0;

                for (int j = 2; j < i; j++) {

                    if (i % j == 0) {

                        flag = 1;

                        break;

                    }

                }

                if (flag == 0) {

                    if (i != 1) {

                        try {

                            dout.writeInt(i);

                            dout.flush();

                            System.out.println("Waiting 1 sec after writing......prime : " + i);

                            try {

                                Thread.sleep(1000);

                            } catch (InterruptedException e) {

                                System.out.println(e);

                            }

                        } catch (IOException e) {

                            e.printStackTrace();

                        }

                    }

                }

            }

            try {

                dout.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

            if (flag == 1) {

                System.out.println("Successfully written prime numebrs from 1 to 100 in file1...");

            } else {

                System.out.println("Successfully written prime numebrs from 101 to 201 in file2...");

            }

        }

    }

}

class ReadThread implements Runnable {

    String filePath;

    DataInputStream din = null;

    int primes[] = new int[100];

    int end = 0;

    int i;

    int file; // 1 - file1 2 - file2

    ReadThread(String filePath, int file) {

        this.filePath = filePath;

        this.file = file;

    }

    public void run() {

        // Waiting for Write operations to complete before reading

        synchronized (ReadThread.class) {

            try {

                Thread.sleep(1000);

            } catch (InterruptedException e) {

                System.out.println(e);

            }

            try {

                din = new DataInputStream(new FileInputStream(filePath));

            } catch (FileNotFoundException e) {

                e.printStackTrace();

            }

            try {

                while ((din.available() > 0)) {

                    i = din.readInt();

                    primes[end++] = i;

                }

                din.close();

            } catch (IOException e) {

                e.printStackTrace();

            }

            if (file == 1) {

                System.out.println("The Primes are (From File1) : ");

                System.out.print("[ ");

                for (int i = 0; i < end; i++) {

                    System.out.print(primes[i] + " ");

                }

                System.out.println("]");

            } else {

                System.out.println("The Primes are (From File2) : ");

                System.out.print("[ ");

                for (int i = 0; i < end; i++) {

                    System.out.print(primes[i] + " ");

                }

                System.out.println("]");

            }

        }

    }

}

public class activity9q3 {

    public static void main(String[] args) throws InterruptedException {

        System.out.println("ANIRUDH VADERA (20BCE2940)");

        WriteThread writeFile1 = new WriteThread(

                "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q3file1.txt", 1, 101, 1);

        Thread t1 = new Thread(writeFile1);

        WriteThread writeFile2 = new WriteThread(

                "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q3file2.txt", 101, 201, 2);

        Thread t2 = new Thread(writeFile2);

        ReadThread readFile1 = new ReadThread(

                "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q3file1.txt", 1);

        Thread t3 = new Thread(readFile1);

        ReadThread readFile2 = new ReadThread(

                "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q3file2.txt", 2);

        Thread t4 = new Thread(readFile2);

        t1.start();

        t2.start();

        t1.join();

        t2.join();

        t3.start();

        t4.start();

        t3.join();

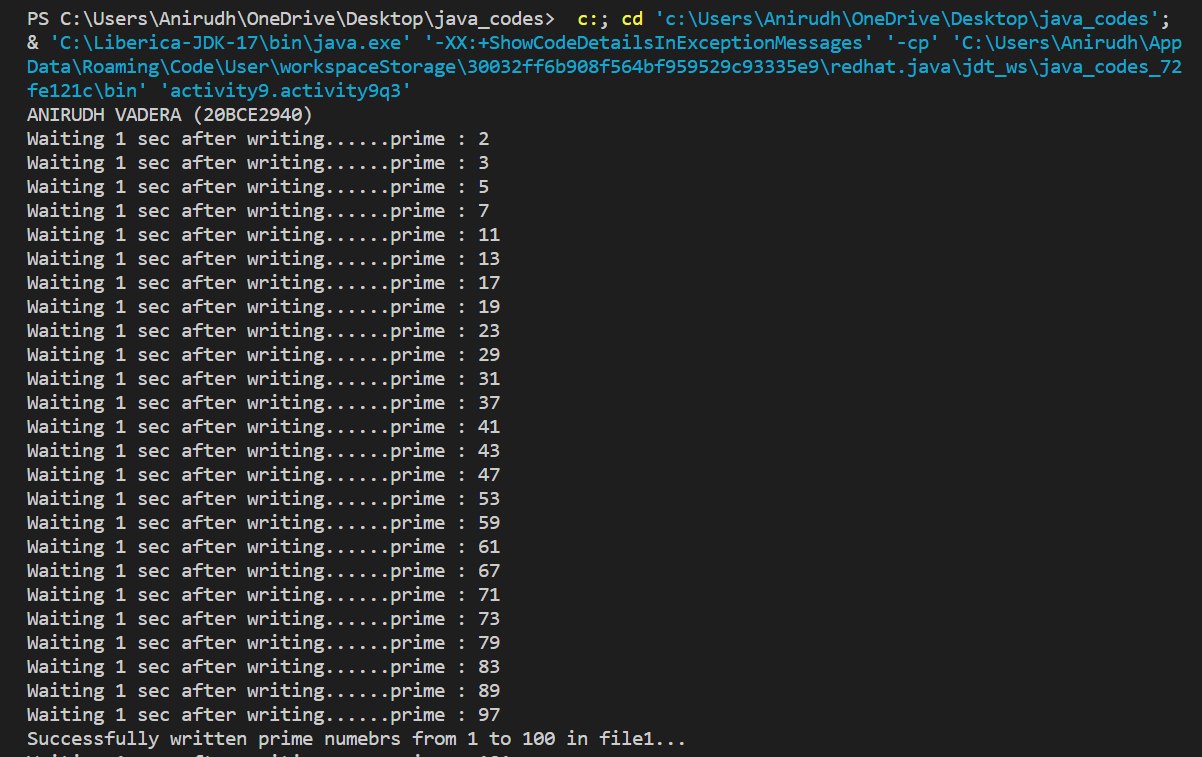
        t4.join();

    }

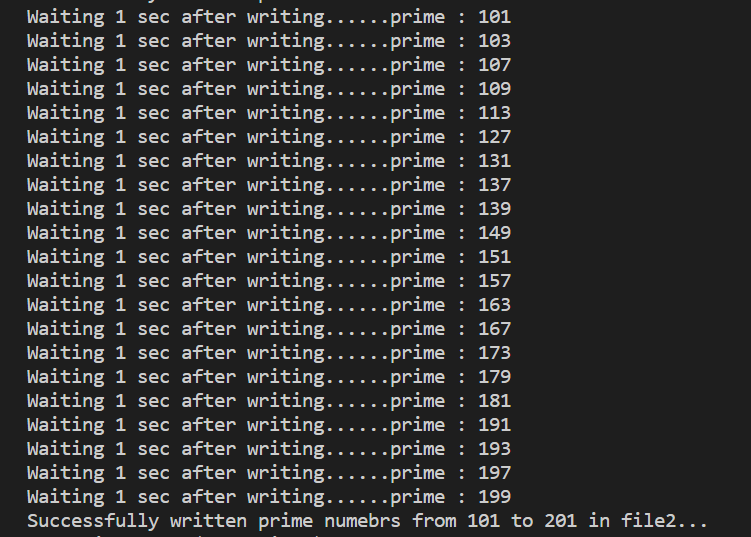
}

**OUTPUT:**

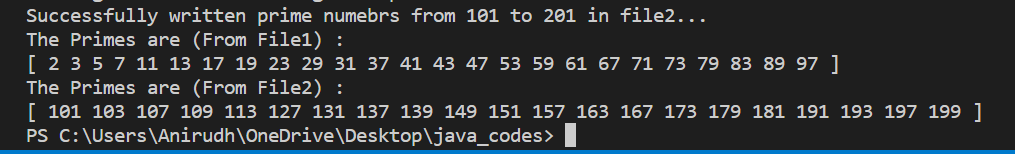
**Writing 1 to 100 in file1:**

****

**Writing 101 to 201 in file1:**

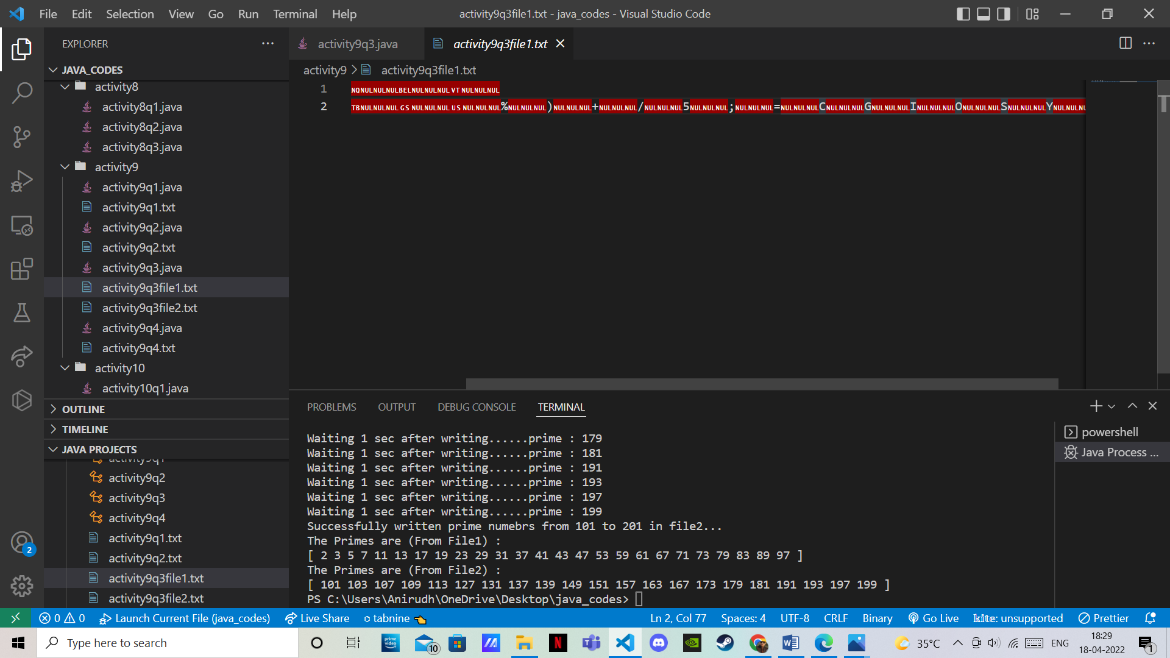
****

**Reading from file:**

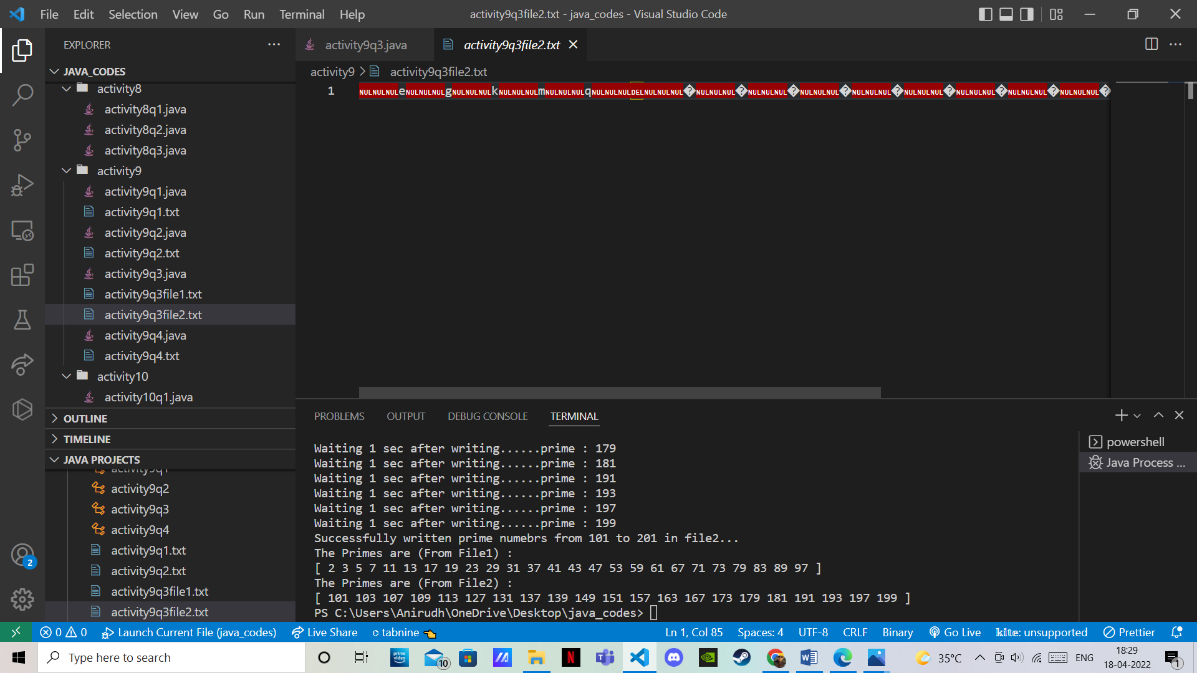
****

**File Contents:**

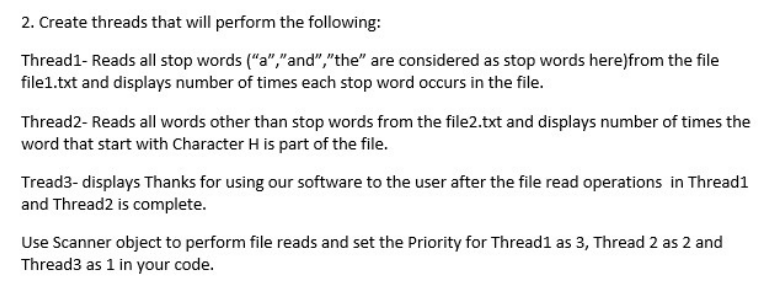
**File 1:**

****

**File 2:**

****

**aQUESTION 4:**

****

**CODE:**

package activity9;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.Scanner;

class readFile implements Runnable {

    int type;

    String filePath;

    readFile(int type, String filePath) {

        this.type = type;

        this.filePath = filePath;

    }

    public void run() {

        synchronized (readFile.class) {

            File file = new File(filePath);

            Scanner in = null;

            try {

                in = new Scanner(file);

                in.useDelimiter(" ");

                if (type == 1) {

                    String[] stop = { "a", "and", "the" };

                    int[] times = new int[3];

                    while (in.hasNext()) {

                        String word = in.next();

                        if (word.equals(stop[0])) {

                            times[0] = times[0] + 1;

                        }

                        if (word.equals(stop[1])) {

                            times[1] = times[1] + 1;

                        }

                        if (word.equals(stop[2])) {

                            times[2] = times[2] + 1;

                        }

                    }

                    System.out.println("Displaying the Number of Stop Words : ");

                    for (int i = 0; i < stop.length; i++) {

                        System.out.println("Stop Word : " + stop[i] + " :: Number of times appeared : " + times[i]);

                    }

                } else {

                    String[] stop = { "a", "and", "the" };

                    int times = 0;

                    while (in.hasNext()) {

                        String word = in.next();

                        if (word.equals(stop[0]) || word.equals(stop[1]) || word.equals(stop[2])) {

                        } else {

                            if (word.startsWith("H")) {

                                times++;

                            }

                        }

                    }

                    System.out.println("Number of times Non Stop words starting with 'H' Appeared : " + times);

                }

            } catch (FileNotFoundException e) {

                e.printStackTrace();

            } finally {

                in.close();

            }

        }

    }

}

public class activity9q4 {

    public static void main(String[] args) throws InterruptedException {

        System.out.println("ANIRUDH VADERA (20BCE2940)");

        readFile readThread1 = new readFile(1,

                "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q4file1.txt");

        Thread t1 = new Thread(readThread1);

        readFile readThread2 = new readFile(2,

                "C:/Users/Anirudh/OneDrive/Desktop/java\_codes/activity9/activity9q4file2.txt");

        Thread t2 = new Thread(readThread2);

        Thread t3 = new Thread(() -> {

            System.out.println("Waiting for Thread 1 and 2 to complete ....");

            try {

                Thread.sleep(2000);

            } catch (InterruptedException e) {

                System.out.println(e);

            }

            System.out.println("Thank you for using our Software ...");

        });

        t1.setPriority(3);

        t2.setPriority(2);

        t3.setPriority(1);

        t1.start();

        t2.start();

        t3.start();

        t1.join();

        t2.join();

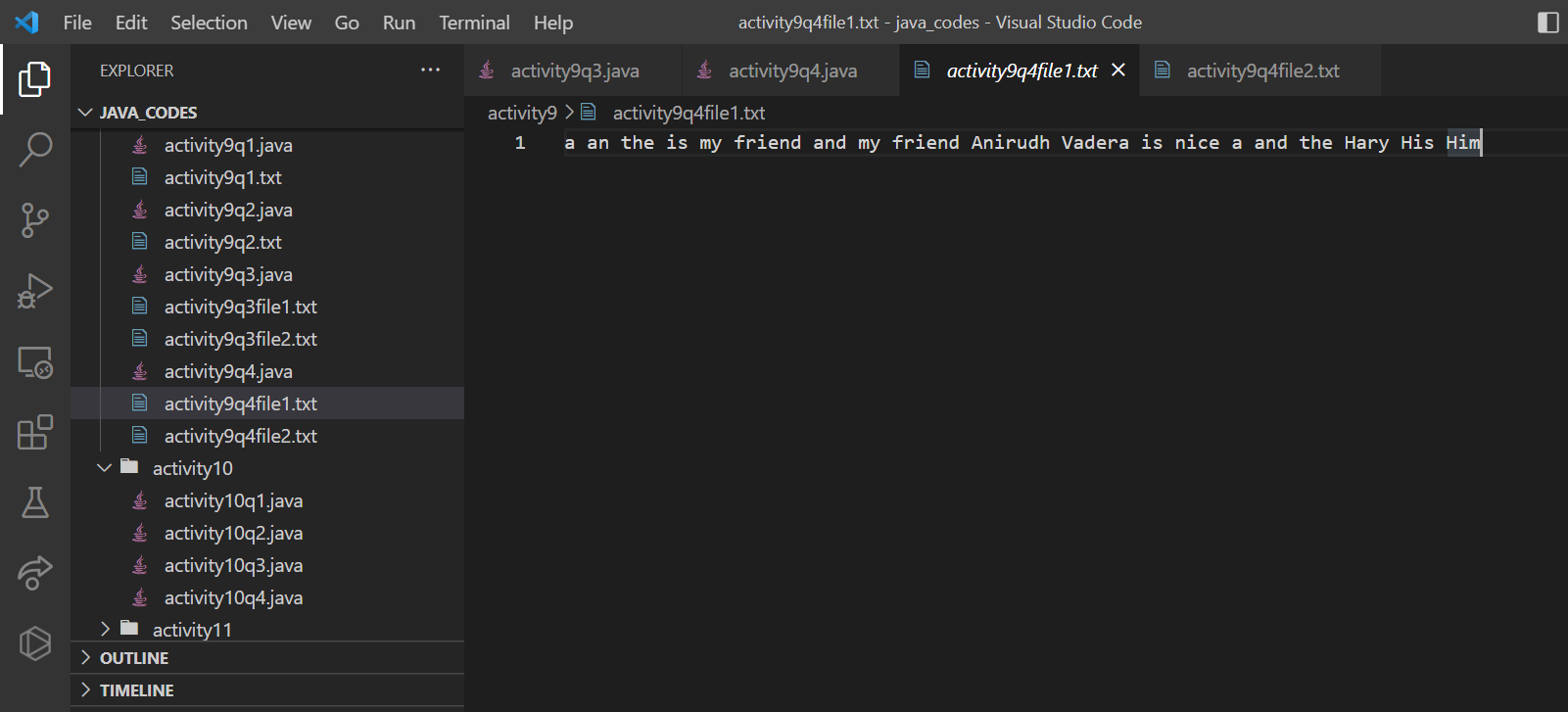
        t3.join();

    }

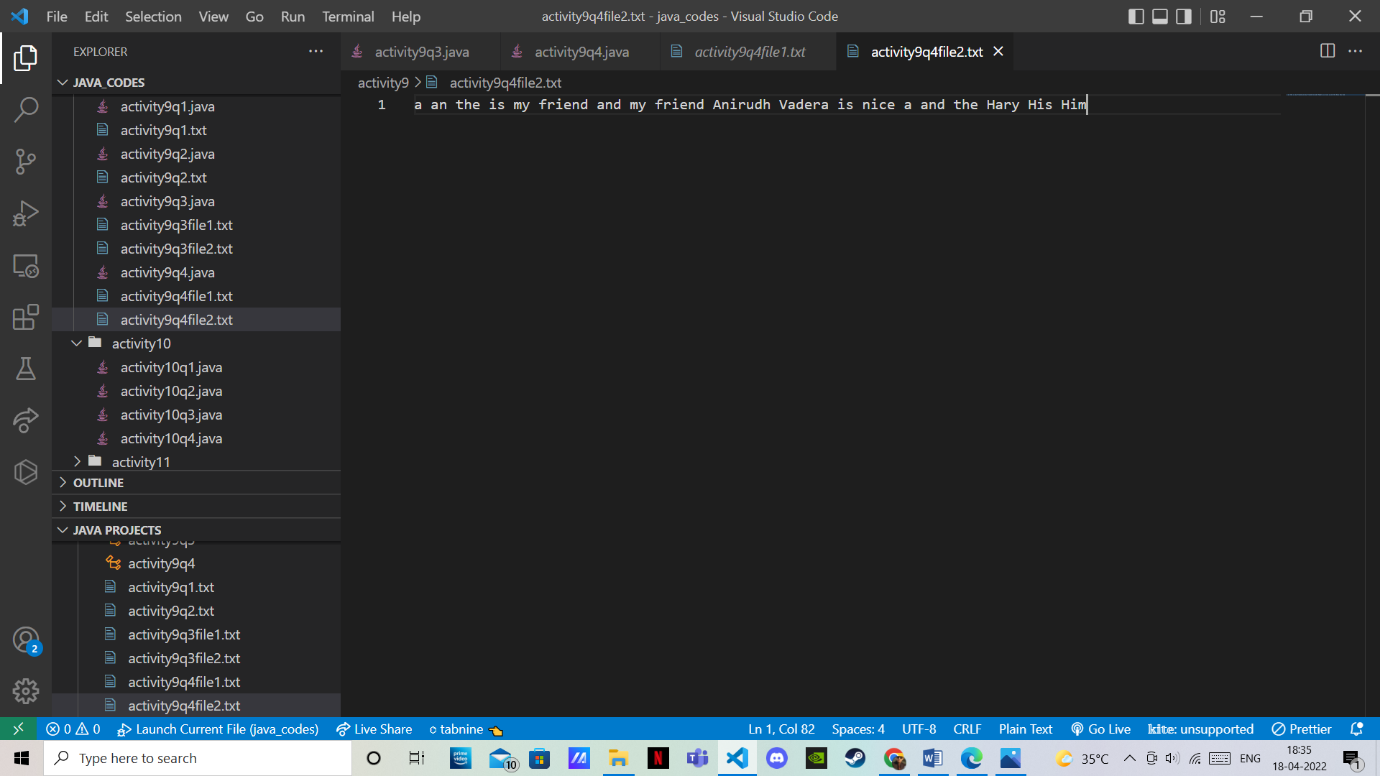
}

**OUTPUT:**

**File 1 content:**

****

**File 2 content:**

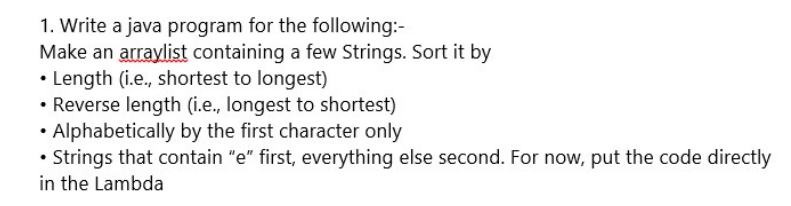
****

**Output:**

****

**ACTIVITY – 10:(Collection)**

**QUESTION 1:**

****

**CODE:**

package activity10;

import java.util.ArrayList;

import java.util.Comparator;

public class activity10q1 {

    public static void main(String[] args) {

        ArrayList<String> list = new ArrayList<String>();

        System.out.println("ANIRUDH VADERA (20BCE2940)");

        list.add("Volkswagen");

        list.add("Toyota");

        list.add("Porsche");

        list.add("Ferrari");

        list.add("Mercedes-Benz");

        list.add("Audi");

        list.add("Rolls-Royce");

        list.add("BMW");

        list.add("E\_Coe");

        System.out.println("Before Sorting: " + list);

        list.sort((p1, p2) -> p1.length() - p2.length());

        System.out.println("After Sorting(Length): " + list);

        list.sort((p1, p2) -> p2.length() - p1.length());

        System.out.println("After Sorting(Reverse Length): " + list);

        list.sort((str1, str2) -> str1.charAt(0) - str2.charAt(0));

        System.out.println("After Sorting(First Character): " + list);

        list.sort(Comparator.comparingInt(a -> (a.startsWith("E") || a.startsWith("E") ? 0 : 1)));

        System.out.println("After Sorting(First e): " + list);

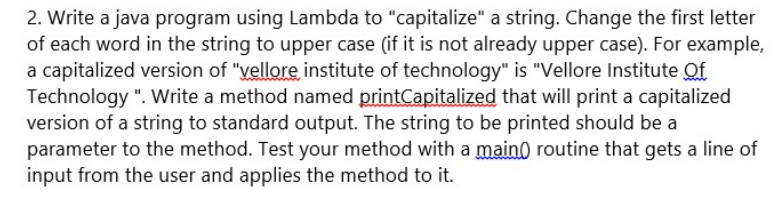
    }

}

**OUTPUT:**

****

**QUESTION 2:**

****

**CODE:**

package activity10;

import java.util.Scanner;

interface Line {

    public void printCapitalized(String s);

}

class Worker implements Line {

    public void printCapitalized(String s) {

        StringBuffer str = new StringBuffer(s);

        if (str.charAt(0) >= 97 && str.charAt(0) <= 122)

            str.setCharAt(0, (char) ((int) str.charAt(0) - 32));

        for (int i = 0; i < str.length(); i++) {

            if (str.charAt(i) == ' ') {

                if (str.charAt(i + 1) >= 97 && str.charAt(i + 1) <= 122)

                    str.setCharAt(i + 1, (char) ((int) str.charAt(i + 1) - 32));

            }

        }

        System.out.println(str.toString());

    }

}

public class activity10q2 {

    static void doWork(Line w) {

        Scanner sc = new Scanner(System.in);

        w.printCapitalized(sc.nextLine());

    }

    public static void main(String[] args) {

        doWork(

                (String s) -> {

                    StringBuffer str = new StringBuffer(s);

                    if (str.charAt(0) >= 97 && str.charAt(0) <= 122)

                        str.setCharAt(0, (char) ((int) str.charAt(0) - 32));

                    for (int i = 0; i < str.length(); i++) {

                        if (str.charAt(i) == ' ') {

                            if (str.charAt(i + 1) >= 97 && str.charAt(i + 1) <= 122)

                                str.setCharAt(i + 1, (char) ((int) str.charAt(i + 1) - 32));

                        }

                    }

                    System.out.println(str.toString());

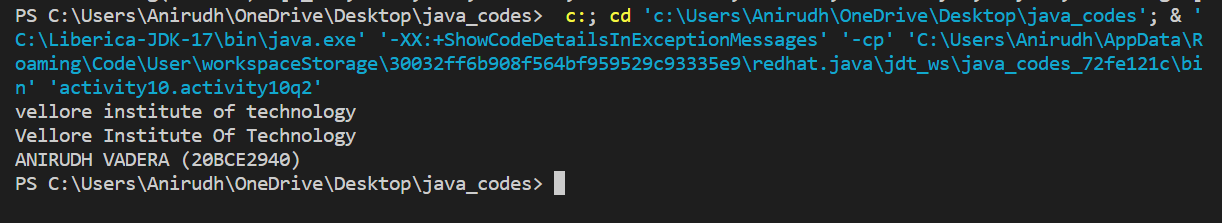
                });

        System.out.println("ANIRUDH VADERA (20BCE2940)");

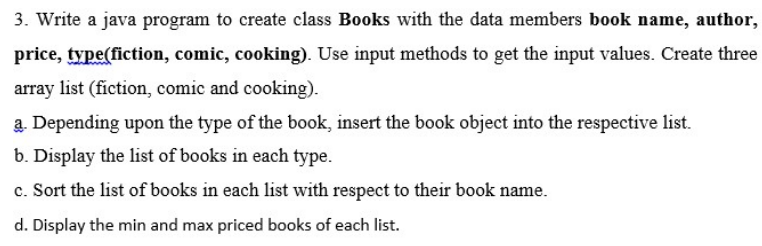
    }

}

**OUTPUT:**

****

**QUESTION 3:**

****

**CODE:**

package activity10;

import java.util.List;

import java.util.ArrayList;

import java.util.Scanner;

class Book {

    String bookName;

    String author;

    int price;

    String type;

    Book() {

        bookName = "";

        author = "";

        price = 0;

        type = "";

    }

    Book(Book b) {

        this.bookName = b.bookName;

        this.type = b.type;

        this.price = b.price;

        this.author = b.author;

    }

    void input(String bookName, String author, int price, String type) {

        this.author = author;

        this.bookName = bookName;

        this.price = price;

        this.type = type;

    }

}

public class activity10q3 {

    static List<Book> fiction = new ArrayList<>();

    static List<Book> comic = new ArrayList<>();

    static List<Book> cooking = new ArrayList<>();

    static int book\_count;

    static void sort\_by\_Bookname(List<Book> list) {

        for (int i = 0; i < list.size() - 1; i++) {

            int min = i;

            for (int j = i + 1; j < list.size(); j++) {

                if ((list.get(j).bookName).compareTo((list.get(min).bookName)) < 0)

                    min = j;

            }

            if (min != i) {

                Book temp = list.get(i);

                list.set(i, list.get(min));

                list.set(min, temp);

            }

        }

    }

    static Book maxPriced(List<Book> list, boolean flag) {

        Book max = list.get(0);

        for (Book i : list) {

            if (flag == true && i.price > max.price)

                max = i;

            else if (flag == false && i.price < max.price)

                max = i;

        }

        return max;

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the number of books : ");

        book\_count = sc.nextInt();

        sc.nextLine();

        Book temp = new Book();

        for (int i = 0; i < book\_count; i++) {

            System.out.println("Enter book name : ");

            temp.bookName = sc.nextLine();

            System.out.println("Enter author name : ");

            temp.author = sc.nextLine();

            System.out.println("Enter price : ");

            temp.price = sc.nextInt();

            sc.nextLine();

            System.out.println("Enter book type (fiction, comic, cooking) : ");

            temp.type = sc.nextLine();

            Book insert = new Book(temp);

            if (temp.type.equals("comic"))

                comic.add(insert);

            else if (temp.type.equals("fiction"))

                fiction.add(insert);

            else if (temp.type.equals("cooking"))

                cooking.add(insert);

            else {

                System.out.println("Wrong type entered!");

                System.exit(0);

            }

        }

        System.out.println("Comic books : ");

        for (Book i : comic)

            System.out.print(i.bookName + "\t");

        System.out.println();

        System.out.println("Fiction books : ");

        for (Book i : fiction)

            System.out.print(i.bookName + "\t");

        System.out.println();

        System.out.println("Cooking books : ");

        for (Book i : cooking)

            System.out.print(i.bookName + "\t");

        System.out.println();

        sort\_by\_Bookname(comic);

        System.out.println("(After Sorting)Comic books : ");

        for (Book i : comic)

            System.out.print(i.bookName + "\t");

        System.out.println();

        sort\_by\_Bookname(fiction);

        System.out.println("(After Sorting)Fiction books : ");

        for (Book i : fiction)

            System.out.print(i.bookName + "\t");

        System.out.println();

        sort\_by\_Bookname(cooking);

        System.out.println("(After Sorting)Cooking books : ");

        for (Book i : cooking)

            System.out.print(i.bookName + "\t");

        System.out.println();

        System.out.println("Max priced comic book : " + maxPriced(comic, true).bookName + "Price : "

                + maxPriced(comic, true).price);

        System.out.println("Min priced comic book : " + maxPriced(comic, false).bookName + "Price : "

                + maxPriced(comic, false).price);

        System.out.println("Max priced fiction book : " + maxPriced(fiction, true).bookName + "Price : "

                + maxPriced(fiction, true).price);

        System.out.println("Min priced fiction book : " + maxPriced(fiction, false).bookName + "Price : "

                + maxPriced(fiction, false).price);

        System.out.println("Max priced cooking book : " + maxPriced(cooking, true).bookName + "Price : "

                + maxPriced(cooking, true).price);

        System.out.println("Min priced cooking book : " + maxPriced(cooking, false).bookName + "Price : "

                + maxPriced(cooking, false).price);

        System.out.println("ANIRUDH VADERA (20BCE2940)");

        sc.close();

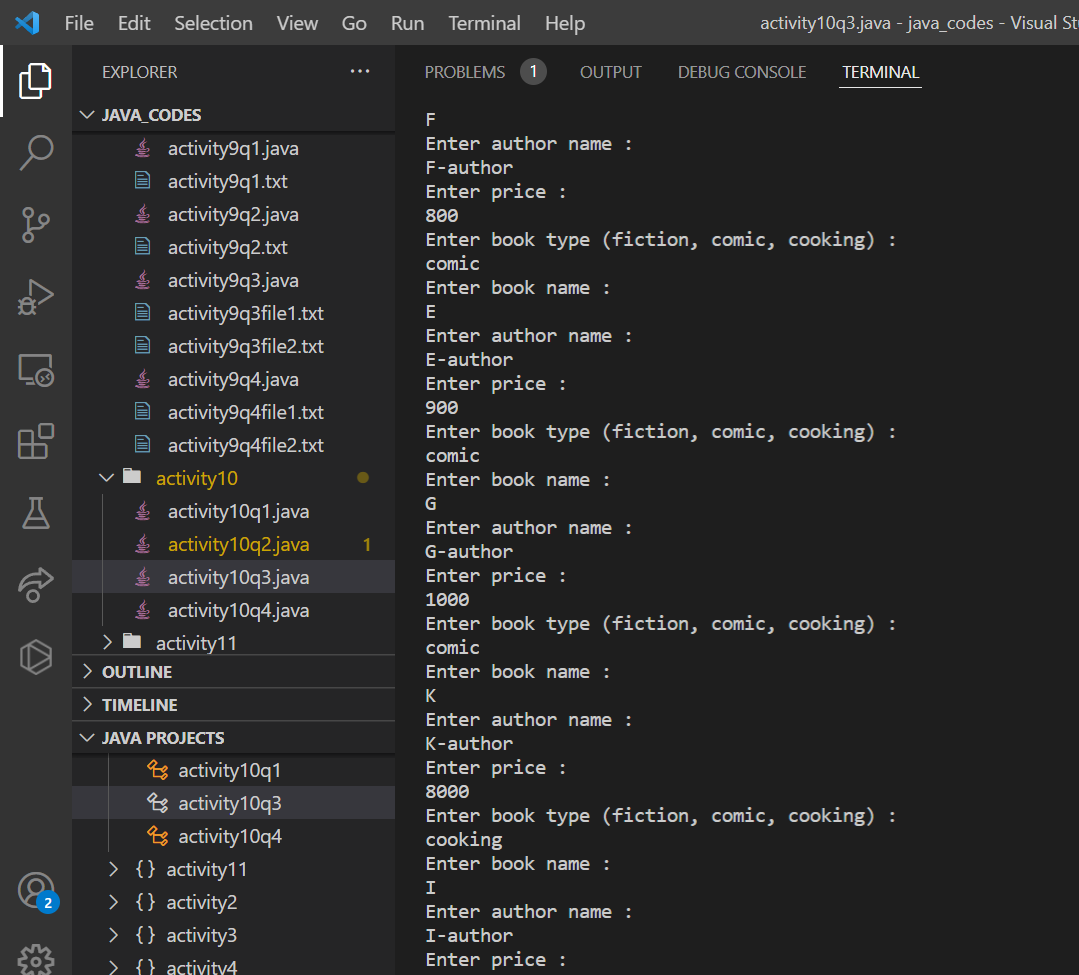
    }

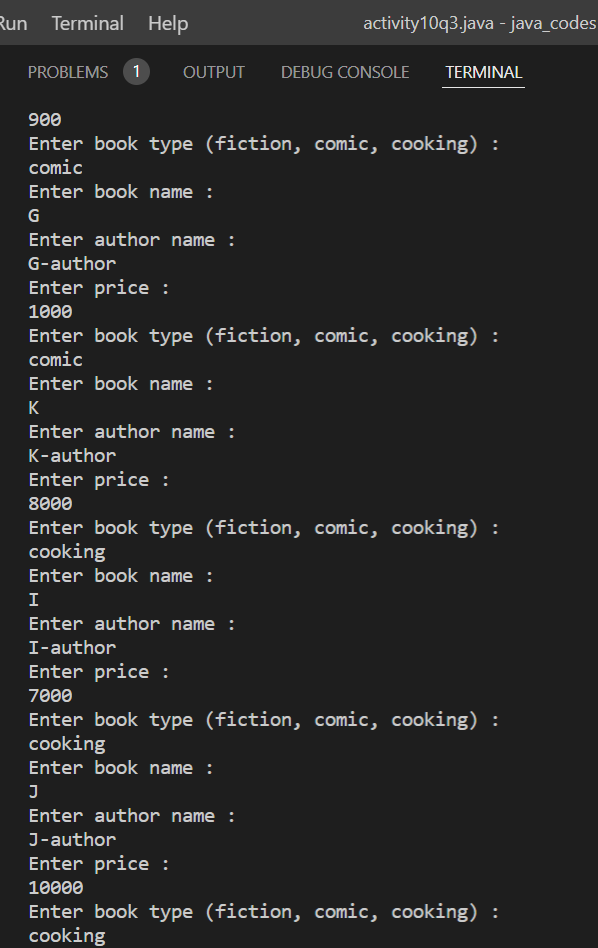
}

**OUTPUT:**

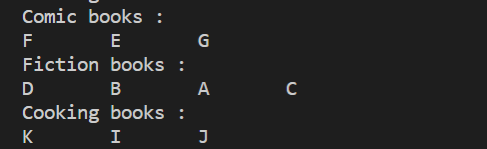
**Entering the details:**

****

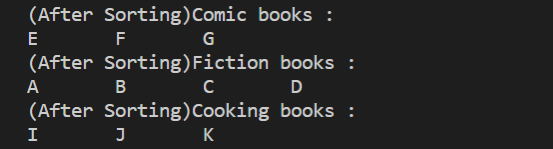
****

****

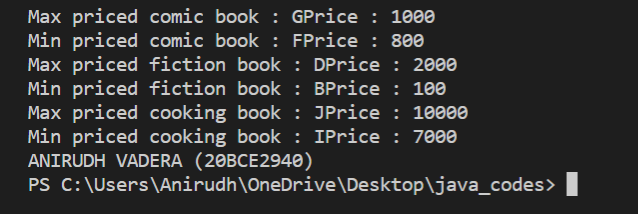
**List of Book in Each type:**

****

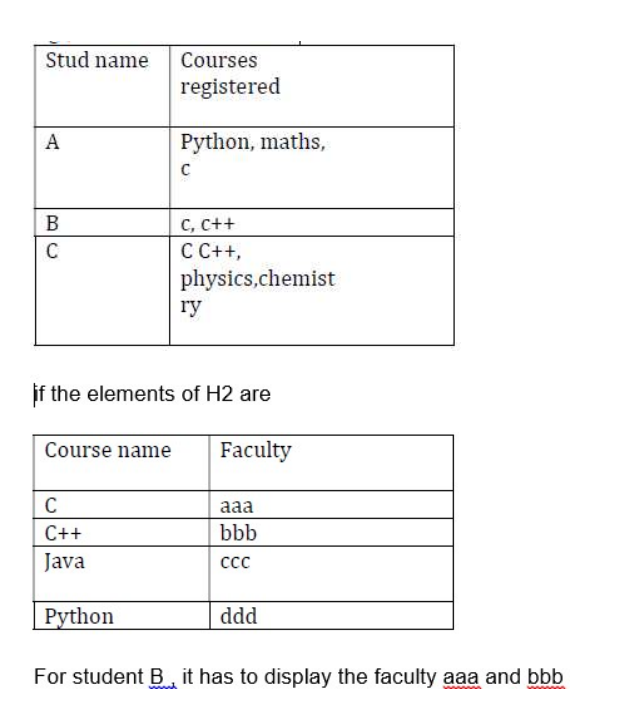
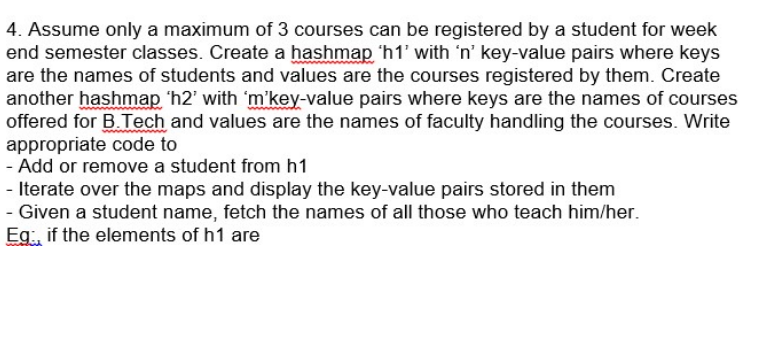
**After sorting List of Book in each type:**

****

**Min and Max priced book of each type:**

****

**QUESTION 4:**

****

**CODE:**

package activity10;

import java.util.\*;

public class activity10q4 {

    Map<String, List<String>> h1 = new HashMap<>();

    Map<String, String> h2 = new HashMap<>();

    int n, m;

    void insert\_to\_studentList(String student, String course) {

        if (h1.containsKey(student)) {

            List<String> courses = new ArrayList<>(h1.get(student));

            courses.add(course);

            h1.put(student, courses);

        } else {

            List<String> courseTemp = new ArrayList<>();

            courseTemp.add(course);

            h1.put(student, courseTemp);

        }

    }

    void removeStudent(String student) {

        if (h1.containsKey(student))

            h1.remove(student);

    }

    void showMaps() {

        System.out.println("Student Name\tCourses");

        Set<Map.Entry<String, List<String>>> keySet = h1.entrySet();

        for (Map.Entry<String, List<String>> set : keySet)

            System.out.println(set.getKey() + "\t" + set.getValue());

        System.out.println("Courses\tTeachers");

        Set<Map.Entry<String, String>> newSet = h2.entrySet();

        for (Map.Entry<String, String> set : newSet)

            System.out.println(set.getKey() + "\t" + set.getValue());

    }

    void showFaculty(String student) {

        System.out.println("Teachers for " + student);

        if (h1.containsKey(student))

            for (String course : h1.get(student))

                System.out.print(h2.get(course) + "\t");

    }

    public static void main(String[] args) {

        activity10q4 q = new activity10q4();

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter n : ");

        q.n = sc.nextInt();

        System.out.println("Enter m : ");

        q.m = sc.nextInt();

        sc.nextLine();

        System.out.println("Enter student details");

        for (int i = 0; i < q.n; i++) {

            System.out.println("Enter student name : ");

            String name = sc.nextLine();

            System.out.println("Enter " + name + "'s courses : (Press e to stop)");

            while (true) {

                String course = sc.nextLine();

                if (course.length() == 1 && course.charAt(0) == 'e')

                    break;

                q.insert\_to\_studentList(name, course);

            }

        }

        System.out.println("Enter faculty details");

        for (int i = 0; i < q.m; i++) {

            System.out.println("Enter course name : ");

            String name = sc.nextLine();

            System.out.println("Enter " + name + " faculty name : ");

            q.h2.put(name, sc.nextLine());

        }

        q.showMaps();

        System.out.println("Enter the student name whose faculty names you want to view : ");

        String name = sc.nextLine();

        q.showFaculty(name);

        System.out.println("ANIRUDH VADERA (20BCE2940)");

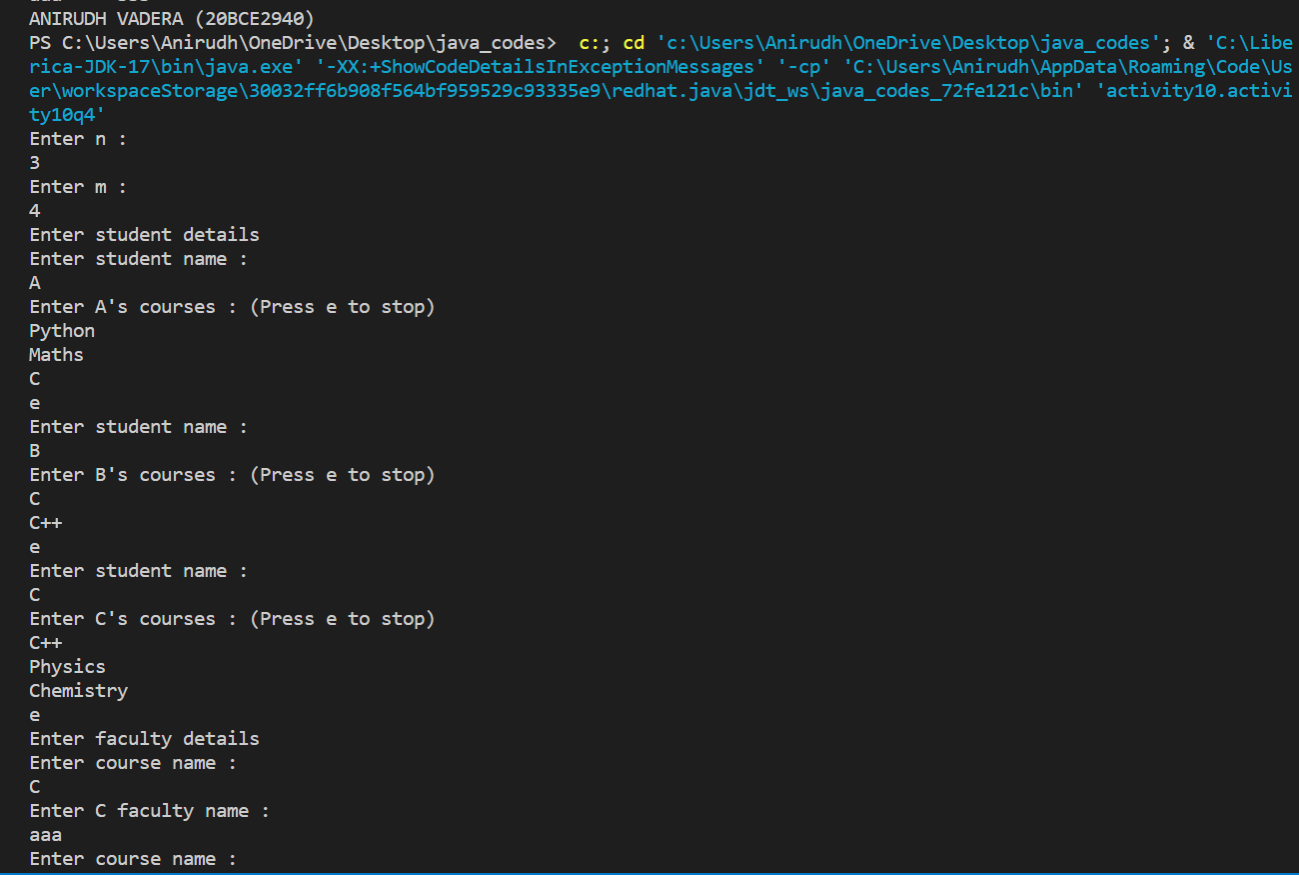
        sc.close();

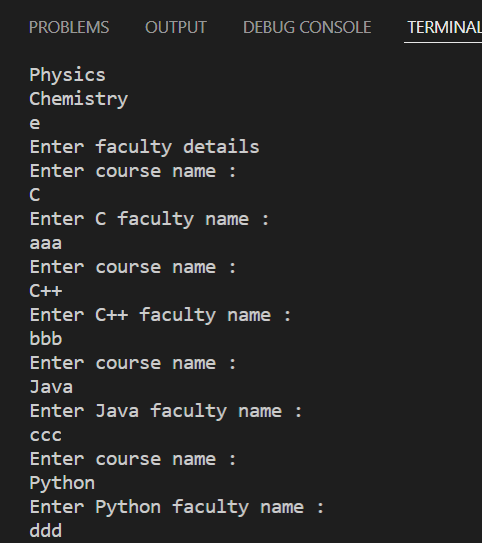
    }

}

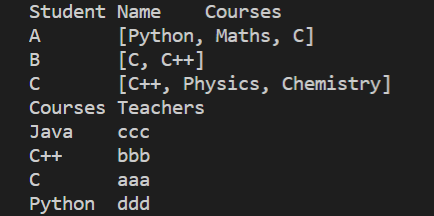
**OUTPUT:**

**Entering the details of the student:**

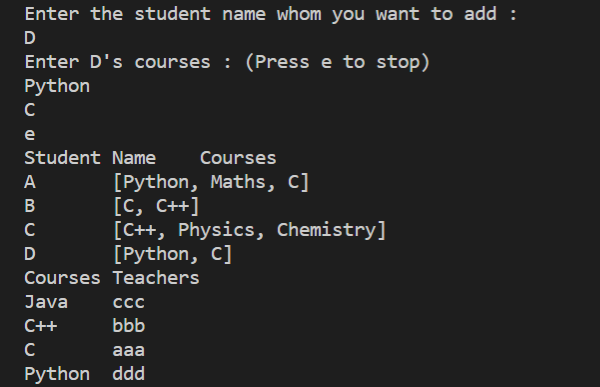
****

****

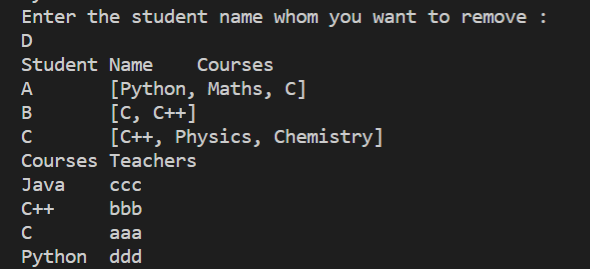
**Iterating through maps and showing the stored details:**

****

**Adding a student:**

****

**Removing a student:**

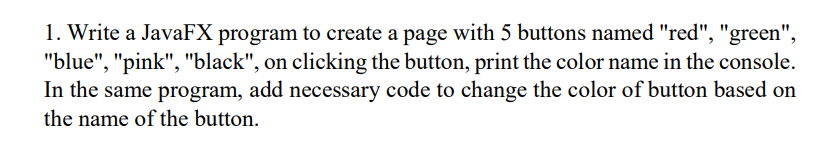
****

**Given a student name fetching details of the teacher:**

****

**ACTIVITY – 11:(JAVAFX)**

**QUESTION 1:**

****

**CODE:**

package activity11;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.HPos;

import javafx.geometry.Pos;

import javafx.geometry.VPos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.layout.GridPane;

import javafx.stage.Stage;

public class activity11q1 extends Application {

    public void start(Stage primaryStage) {

        int[] flag = { 0, 0, 0, 0, 0 };

        Button btn1 = new Button();

        btn1.setText("Red");

        Button btn2 = new Button();

        btn2.setText("Green");

        Button btn3 = new Button();

        btn3.setText("Blue");

        Button btn4 = new Button();

        btn4.setText("Pink");

        Button btn5 = new Button();

        btn5.setText("Black");

        btn1.setOnAction(new EventHandler<ActionEvent>() {

            @Override

            public void handle(ActionEvent event) {

                if (flag[0] == 0) {

                    flag[0] = 1;

                    btn2.setStyle("");

                    btn3.setStyle("");

                    btn4.setStyle("");

                    btn5.setStyle("");

                    flag[1] = 0;

                    flag[2] = 0;

                    flag[3] = 0;

                    flag[4] = 0;

                } else {

                    flag[0] = 0;

                }

                System.out.println("Red");

                if (flag[0] == 1) {

                    btn1.setStyle("-fx-background-color: red;-fx-text-fill: white; ");

                } else {

                    btn1.setStyle("");

                }

            }

        });

        btn2.setOnAction(new EventHandler<ActionEvent>() {

            public void handle(ActionEvent event) {

                if (flag[1] == 0) {

                    flag[1] = 1;

                    btn1.setStyle("");

                    btn3.setStyle("");

                    btn4.setStyle("");

                    btn5.setStyle("");

                    flag[0] = 0;

                    flag[2] = 0;

                    flag[3] = 0;

                    flag[4] = 0;

                } else {

                    flag[1] = 0;

                }

                System.out.println("Green");

                if (flag[1] == 1) {

                    btn2.setStyle("-fx-background-color: green;-fx-text-fill: white; ");

                } else {

                    btn2.setStyle("");

                }

            }

        });

        btn3.setOnAction(new EventHandler<ActionEvent>() {

            public void handle(ActionEvent event) {

                if (flag[2] == 0) {

                    flag[2] = 1;

                    btn2.setStyle("");

                    btn1.setStyle("");

                    btn4.setStyle("");

                    btn5.setStyle("");

                    flag[1] = 0;

                    flag[0] = 0;

                    flag[3] = 0;

                    flag[4] = 0;

                } else {

                    flag[2] = 0;

                }

                System.out.println("Blue");

                if (flag[2] == 1) {

                    btn3.setStyle("-fx-background-color: blue;-fx-text-fill: white; ");

                } else {

                    btn3.setStyle("");

                }

            }

        });

        btn4.setOnAction(new EventHandler<ActionEvent>() {

            public void handle(ActionEvent event) {

                if (flag[3] == 0) {

                    flag[3] = 1;

                    btn2.setStyle("");

                    btn3.setStyle("");

                    btn1.setStyle("");

                    btn5.setStyle("");

                    flag[1] = 0;

                    flag[2] = 0;

                    flag[0] = 0;

                    flag[4] = 0;

                } else {

                    flag[3] = 0;

                }

                System.out.println("Pink");

                if (flag[3] == 1) {

                    btn4.setStyle("-fx-background-color: pink;-fx-text-fill: white; ");

                } else {

                    btn4.setStyle("");

                }

            }

        });

        btn5.setOnAction(new EventHandler<ActionEvent>() {

            public void handle(ActionEvent event) {

                if (flag[4] == 0) {

                    flag[4] = 1;

                    btn2.setStyle("");

                    btn3.setStyle("");

                    btn4.setStyle("");

                    btn1.setStyle("");

                    flag[1] = 0;

                    flag[2] = 0;

                    flag[3] = 0;

                    flag[0] = 0;

                } else {

                    flag[4] = 0;

                }

                System.out.println("Black");

                if (flag[4] == 1) {

                    btn5.setStyle("-fx-background-color: black;-fx-text-fill: white; ");

                } else {

                    btn5.setStyle("");

                }

            }

        });

        btn1.setMaxSize(100, 200);

        btn2.setMaxSize(100, 200);

        btn3.setMaxSize(100, 200);

        btn4.setMaxSize(100, 200);

        btn5.setMaxSize(100, 200);

        GridPane root = new GridPane();

        root.add(btn1, 0, 0);

        root.add(btn2, 0, 1);

        root.add(btn3, 1, 0);

        root.add(btn4, 1, 1);

        root.add(btn5, 0, 2);

        Scene scene = new Scene(root, 300, 250);

        root.setVgap(10);

        root.setHgap(10);

        root.setAlignment(Pos.CENTER);

        GridPane.setHalignment(btn1, HPos.CENTER);

        GridPane.setValignment(btn1, VPos.CENTER);

        GridPane.setHalignment(btn2, HPos.CENTER);

        GridPane.setValignment(btn2, VPos.CENTER);

        GridPane.setHalignment(btn3, HPos.CENTER);

        GridPane.setValignment(btn3, VPos.CENTER);

        GridPane.setHalignment(btn4, HPos.CENTER);

        GridPane.setValignment(btn4, VPos.CENTER);

        GridPane.setHalignment(btn5, HPos.CENTER);

        GridPane.setValignment(btn5, VPos.CENTER);

        primaryStage.setTitle("ANIRUDH VADERA (20BCE2940)");

        primaryStage.setScene(scene);

        primaryStage.show();

    }

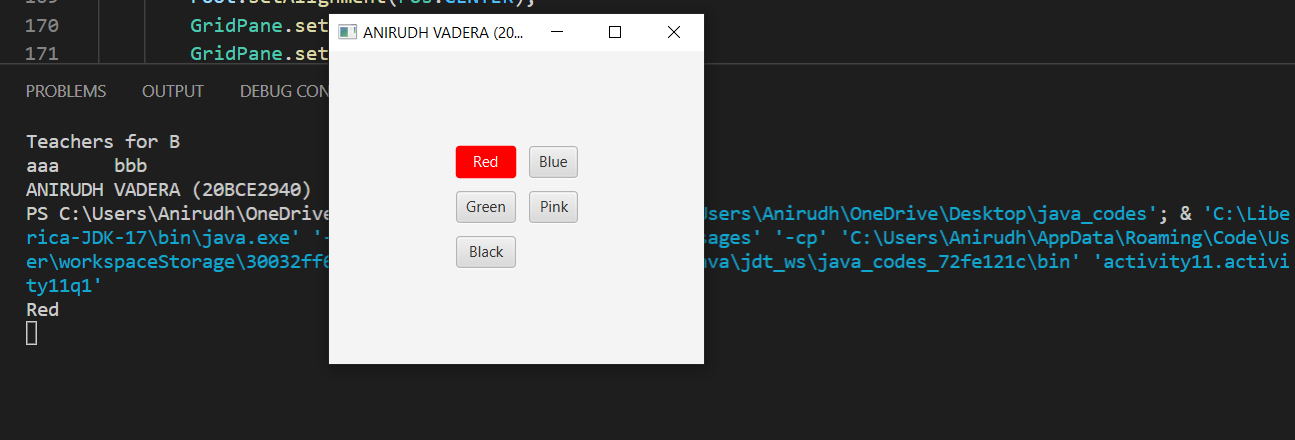
    public static void main(String[] args) {

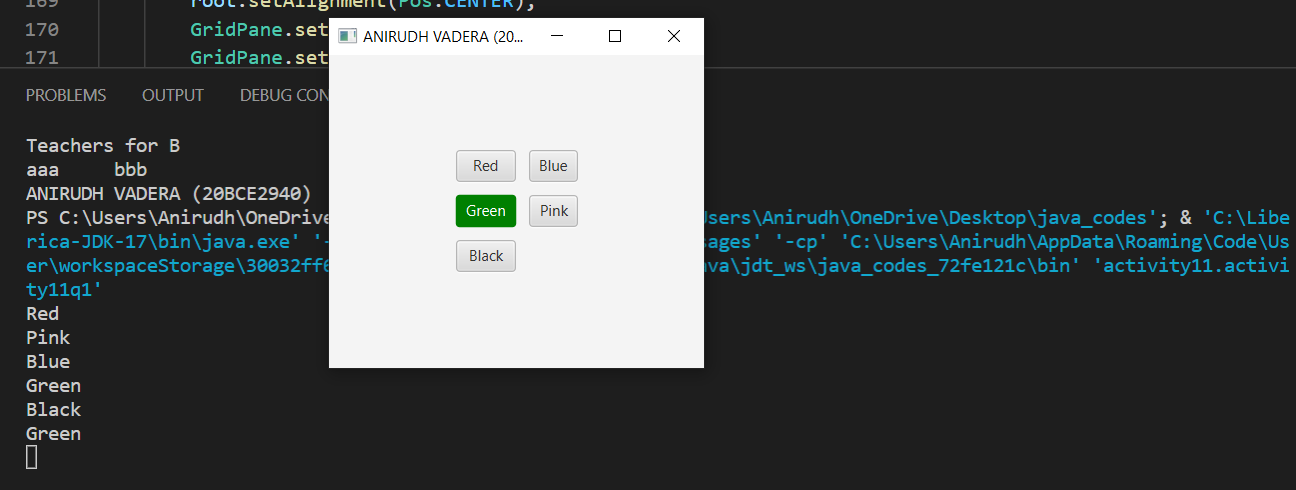
        launch(args);

    }

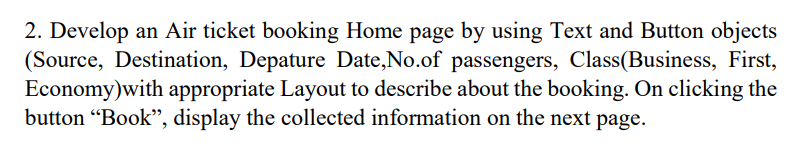
}

**OUTPUT:**

****

****

**QUESTION 2:**

****

**CODE:**

package activity11;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.HPos;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.GridPane;

import javafx.stage.Stage;

import javafx.scene.text.Text;

public class activity11q2 extends Application {

    public void start(Stage primaryStage) throws Exception {

        GridPane rootNode = new GridPane();

        rootNode.setHgap(5);

        rootNode.setVgap(5);

        rootNode.setAlignment(Pos.CENTER);

        Scene myScene = new Scene(rootNode, 700, 500);

        rootNode.add(new Label("Enter Source:"), 0, 0);

        TextField source = new TextField();

        rootNode.add(source, 1, 0);

        rootNode.add(new Label("Enter Destination:"), 0, 1);

        TextField dest = new TextField();

        rootNode.add(dest, 1, 1);

        rootNode.add(new Label("Enter Date of departure:"), 0, 2);

        TextField dod = new TextField();

        rootNode.add(dod, 1, 2);

        rootNode.add(new Label("Enter Number of passengers:"), 0, 3);

        TextField nop = new TextField();

        rootNode.add(nop, 1, 3);

        rootNode.add(new Label("Enter class: "), 0, 4);

        ComboBox comboBox = new ComboBox();

        comboBox.getItems().add("Business");

        comboBox.getItems().add("First");

        comboBox.getItems().add("Economy");

        rootNode.add(comboBox, 1, 4);

        Button btn = new Button("Book");

        rootNode.add(btn, 1, 5);

        GridPane.setHalignment(btn, HPos.LEFT);

        GridPane rootNode2 = new GridPane();

        Scene myScene2 = new Scene(rootNode2, 700, 500);

        rootNode2.setHgap(5);

        rootNode2.setVgap(5);

        rootNode2.setAlignment(Pos.CENTER);

        Button btn2 = new Button("Edit details");

        int flag = 0;

        btn.setOnAction(new EventHandler<ActionEvent>() {

            public void handle(ActionEvent event) {

                String s = "";

                if (flag == 0) {

                    try {

                        String src = source.getText();

                        String dest1 = dest.getText();

                        String dod1 = dod.getText();

                        String nop1 = nop.getText();

                        String cls1 = (String) comboBox.getValue();

                        if (src.equals("") || dest1.equals("") || dod1.equals("") || nop1.equals("")

                                || cls1.equals("")) {

                            s = "Please fill all the fields";

                        } else {

                            s = "Source: " + src + "\nDestination: " + dest1 + "\nDate of departure: " +

                                    dod1

                                    + "\nNumber of passengers: " + nop1 + "\nClass: " + cls1;

                        }

                        Text t = new Text(s);

                        rootNode2.add(t, 1, 1);

                        rootNode2.add(btn2, 1, 2);

                    } catch (Exception e) {

                        s = "";

                    }

                    rootNode2.add(new Label("Details of Booking:"), 1, 0);

                    primaryStage.setScene(myScene2);

                    s = "";

                }

            }

        });

        btn2.setOnAction(new EventHandler<ActionEvent>() {

            public void handle(ActionEvent event) {

                primaryStage.setScene(myScene);

                rootNode2.getChildren().clear();

            }

        });

        primaryStage.setScene(myScene);

        primaryStage.setTitle("Air ticket booking system ANIRUDH VADERA (20BCE2940)");

        primaryStage.setScene(myScene);

        primaryStage.show();

    }

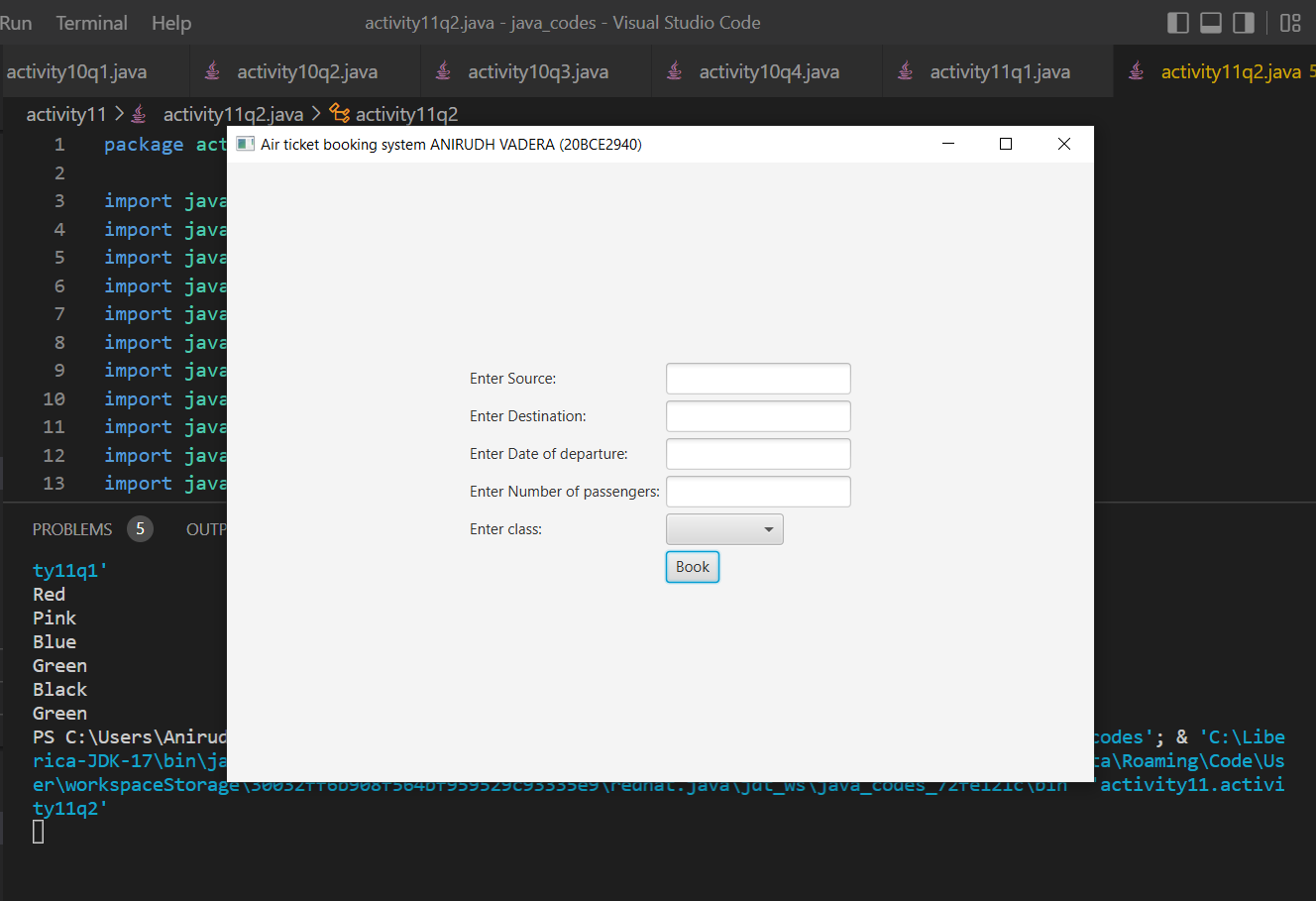
    public static void main(String[] args) {

        launch(args);

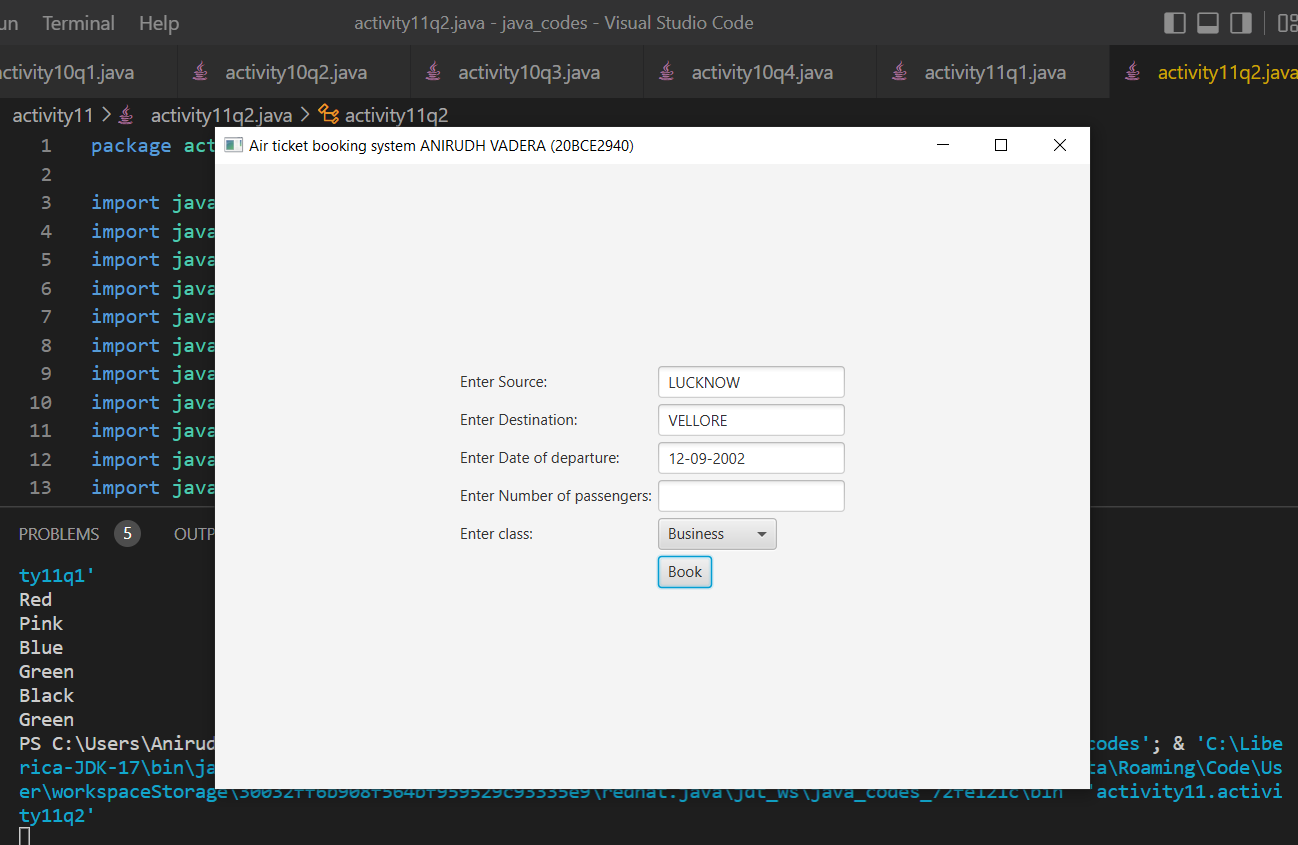
    }

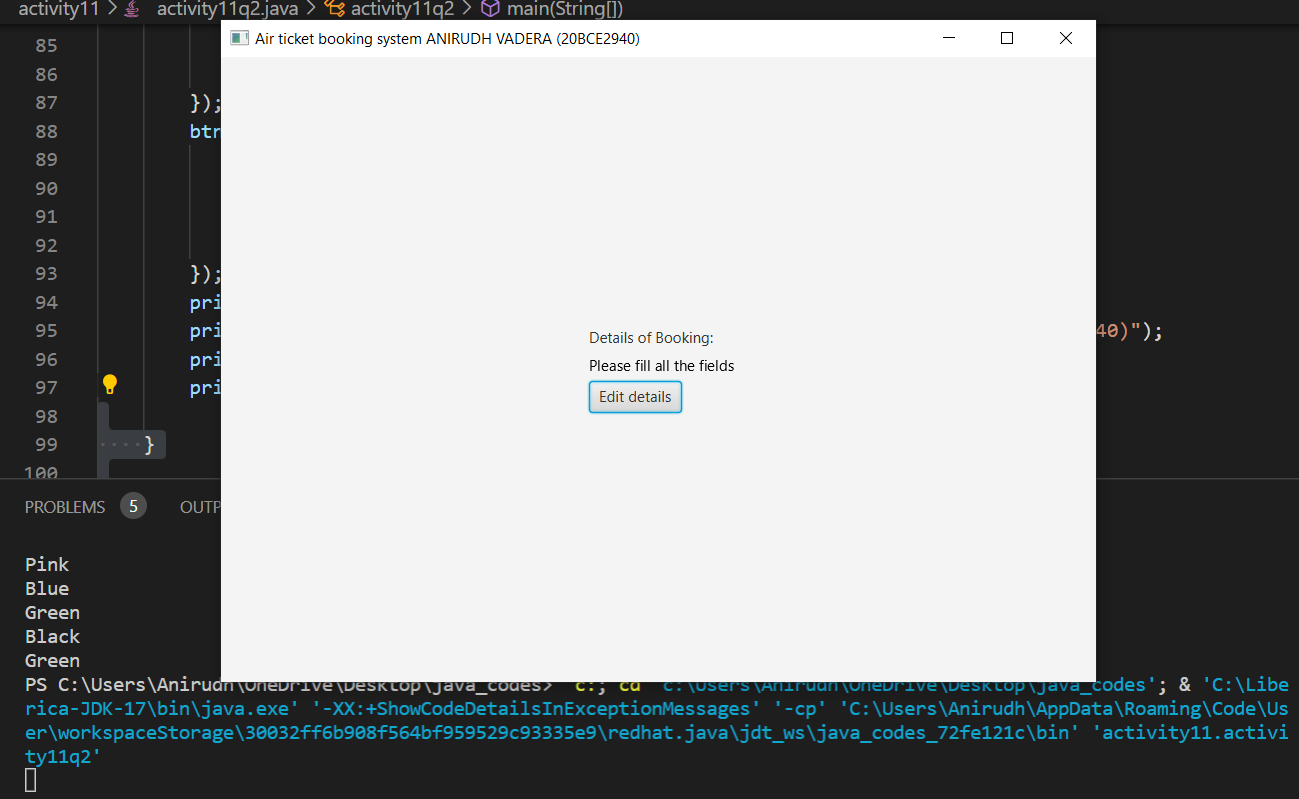
}

**OUTPUT:**

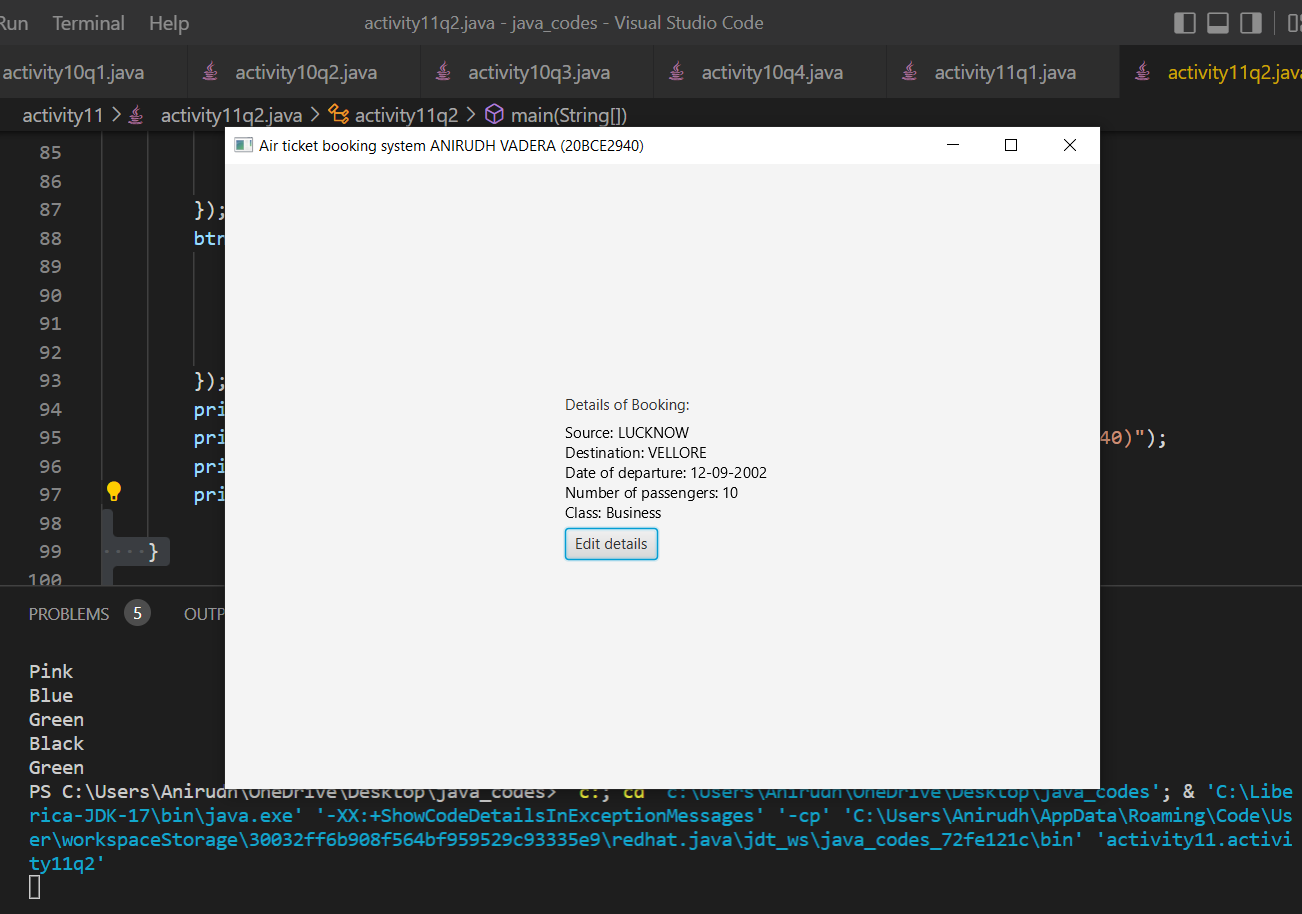
****

**IF SOME DETAILS ARE MISSING:**

****

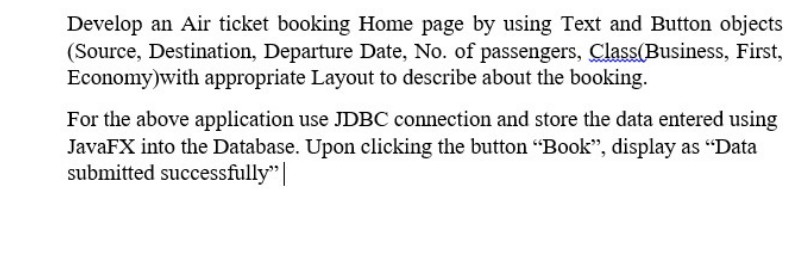
****

**ALL DETAILS SUCCESFULY FILLED:**

****

**ACTIVITY 12:(JDBC)**

**QUESTION 1:**

****

**CODE:**

package javaapplication1;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.SQLException;

import java.sql.Statement;

import javafx.application.Application;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.HPos;

import javafx.geometry.Pos;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.ComboBox;

import javafx.scene.control.Label;

import javafx.scene.control.TextField;

import javafx.scene.layout.GridPane;

import javafx.stage.Stage;

import javafx.scene.text.Text;

import javafx.scene.control.DatePicker;

public class activity11q3 extends Application {

    public void start(Stage primaryStage)

            throws Exception, ClassNotFoundException, InstantiationException, IllegalAccessException {

        GridPane rootNode = new GridPane();

        rootNode.setHgap(5);

        rootNode.setVgap(5);

        rootNode.setAlignment(Pos.CENTER);

        Scene myScene = new Scene(rootNode, 700, 500);

        rootNode.add(new Label("Enter Name:"), 0, 0);

        TextField name = new TextField();

        rootNode.add(name, 1, 0);

        rootNode.add(new Label("Enter phone number:"), 0, 1);

        TextField phno = new TextField();

        rootNode.add(phno, 1, 1);

        rootNode.add(new Label("Enter Source:"), 0, 2);

        TextField source = new TextField();

        rootNode.add(source, 1, 2);

        rootNode.add(new Label("Enter Destination:"), 0, 3);

        TextField dest = new TextField();

        rootNode.add(dest, 1, 3);

        rootNode.add(new Label("Enter Date of departure:"), 0, 4);

        DatePicker dod = new DatePicker();

        rootNode.add(dod, 1, 4);

        rootNode.add(new Label("Enter Number of passengers:"), 0, 5);

        TextField nop = new TextField();

        rootNode.add(nop, 1, 5);

        rootNode.add(new Label("Enter class: "), 0, 6);

        ComboBox comboBox = new ComboBox();

        comboBox.getItems().add("Business");

        comboBox.getItems().add("First");

        comboBox.getItems().add("Economy");

        rootNode.add(comboBox, 1, 6);

        Button btn = new Button("Book");

        rootNode.add(btn, 1, 7);

        GridPane.setHalignment(btn, HPos.LEFT);

        GridPane rootNode2 = new GridPane();

        Scene myScene2 = new Scene(rootNode2, 700, 500);

        rootNode2.setHgap(5);

        rootNode2.setVgap(5);

        rootNode2.setAlignment(Pos.CENTER);

        Button btn2 = new Button("Book more tickets");

        btn.setOnAction(new EventHandler<ActionEvent>() {

            public void handle(ActionEvent event) {

                String s = "";

                int flag = 2;

                try {

                    String names = name.getText();

                    String phone = phno.getText();

                    String src = source.getText();

                    String dest1 = dest.getText();

                    String dod1 = dod.getValue().toString();

                    String nop1 = nop.getText();

                    String cls1 = (String) comboBox.getValue();

                    if (names.equals("") || phone.equals("") || src.equals("") || dest1.equals("") || dod1.equals("")

                            || nop1.equals("")

                            || cls1.equals("")) {

                        s = "Please fill all the fields";

                        s += "\nName: " + names + "\nPhone number: " + phone + "\nSource: " + src + "\nDestination: "

                                + dest1 + "\nDate of departure: " + dod1

                                + "\nNumber of passengers: " + nop1 + "\nClass: " + cls1;

                    } else {

                        s = "Name: " + names + "\nPhone number: " + phone + "\nSource: " + src + "\nDestination: "

                                + dest1 + "\nDate of departure: " + dod1

                                + "\nNumber of passengers: " + nop1 + "\nClass: " + cls1;

                        try {

                            Connection con = null;

                            Class.forName("com.mysql.cj.jdbc.Driver").newInstance();

                            con = DriverManager.getConnection("jdbc:mysql://localhost:3307/javacodes", "root",

                                    "");

                            System.out.println("Connection Successful");

                            Statement sqlStatements = con.createStatement();

                            String querryString = "INSERT INTO airticketbooking VALUES('"

                                    + src + "','" + dest1 + "','" + dod1 + "'," + nop1 + ",'" + cls1 + "','" + names

                                    + "'," + phone + ");";

                            System.out.println(querryString);

                            sqlStatements.executeUpdate(querryString);

                            System.out.println("Record inserted");

                            sqlStatements.close();

                            con.close();

                            flag = 0;

                        } catch (SQLException e) {

                            System.out.println(e.getMessage());

                            flag = 1;

                        }

                    }

                    Text t = new Text(s);

                    rootNode2.add(t, 1, 1);

                    rootNode2.add(btn2, 1, 2);

                } catch (Exception e) {

                    s = "";

                }

                Label heading;

                if (flag == 0) {

                    heading = new Label("Booking Successful");

                } else {

                    heading = new Label("Booking Unsuccessful");

                }

                heading.setStyle("-fx-font-size: 20px;");

                rootNode2.add(heading, 1, 0);

                primaryStage.setScene(myScene2);

                s = "";

            }

        });

        btn2.setOnAction(new EventHandler<ActionEvent>() {

            @Override

            public void handle(ActionEvent event) {

                primaryStage.setScene(myScene);

                rootNode2.getChildren().clear();

            }

        });

        primaryStage.setScene(myScene);

        primaryStage.setTitle("Air ticket booking system ANIRUDH VADERA (20BCE2940)");

        primaryStage.setScene(myScene);

        primaryStage.show();

    }

    public static void main(String[] args) {

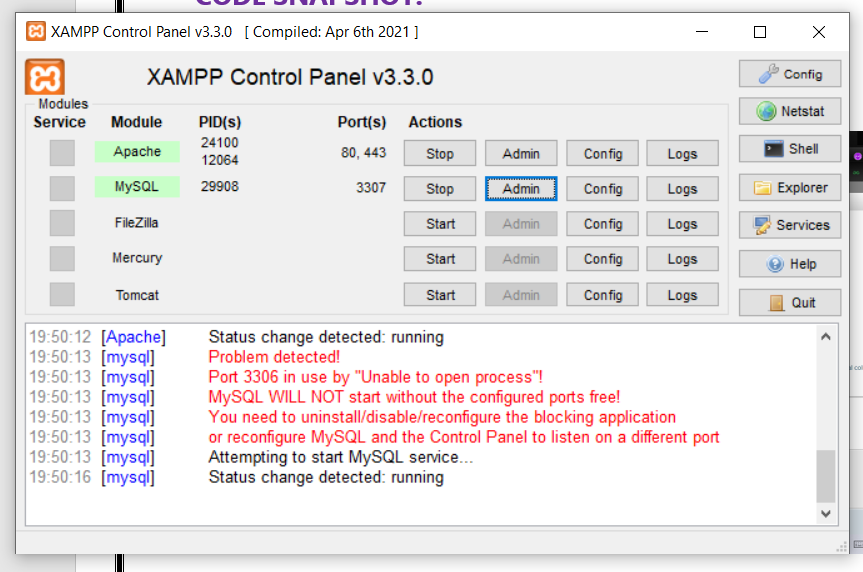
        launch(args);

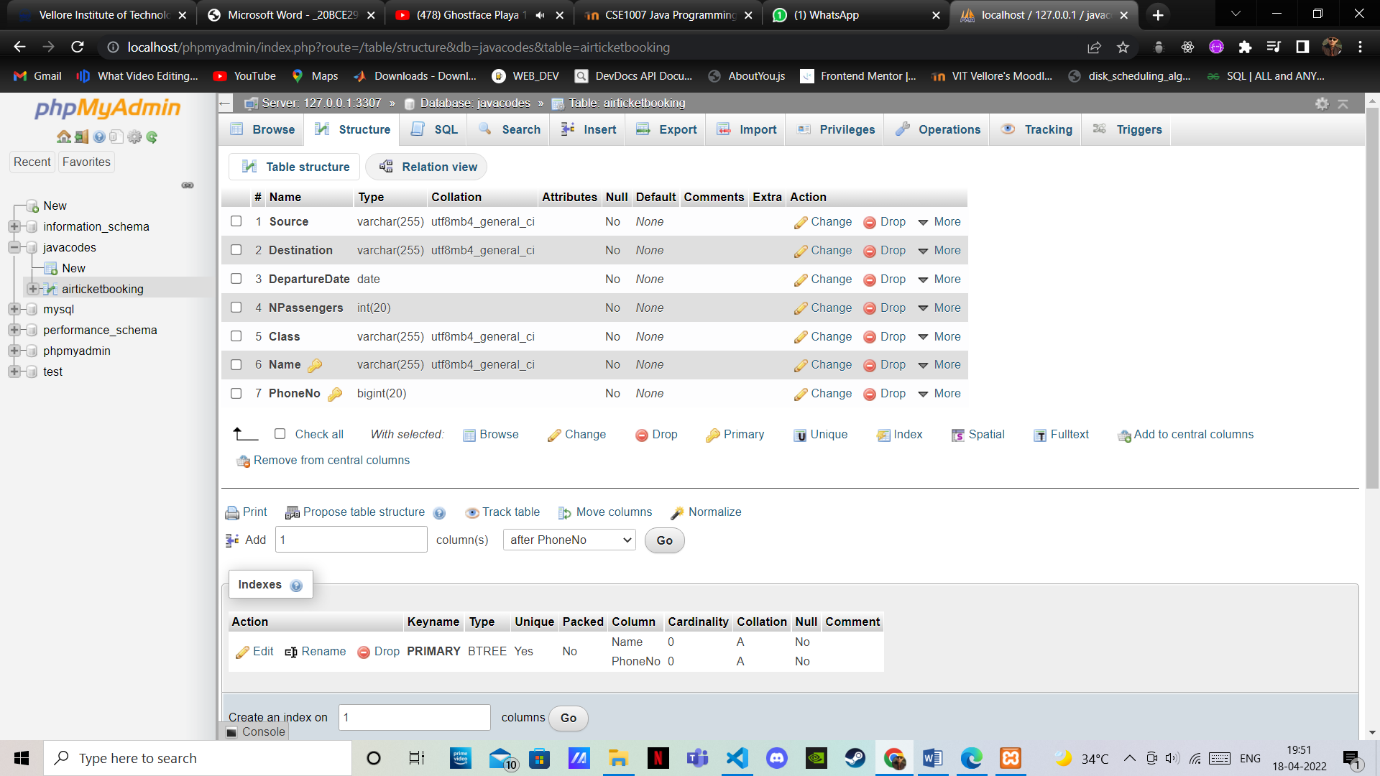
    }

}

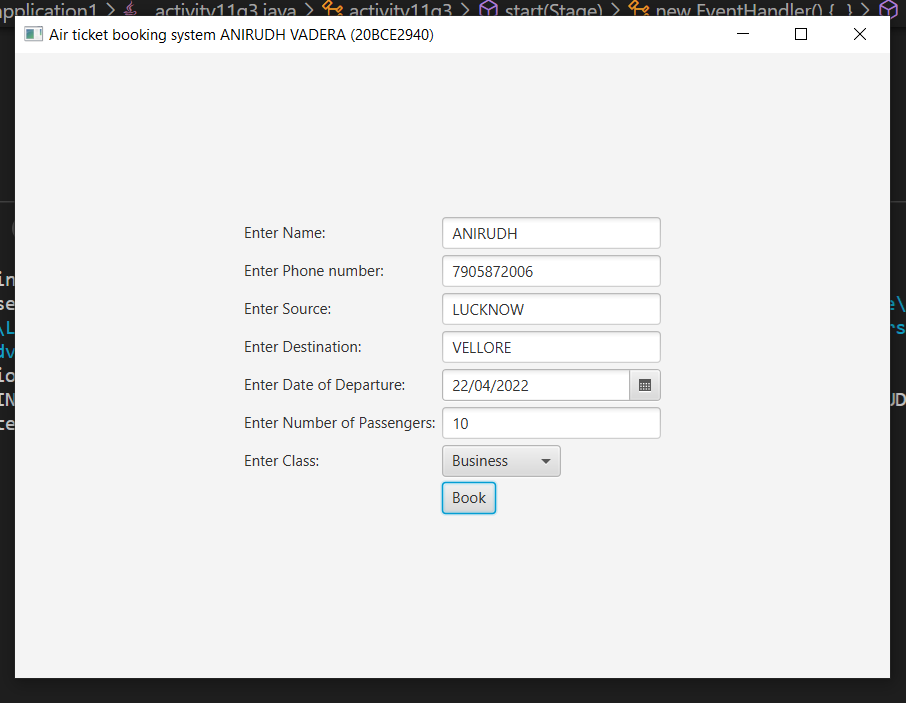
**OUTPUT:**

**Php my admin table Structure:**

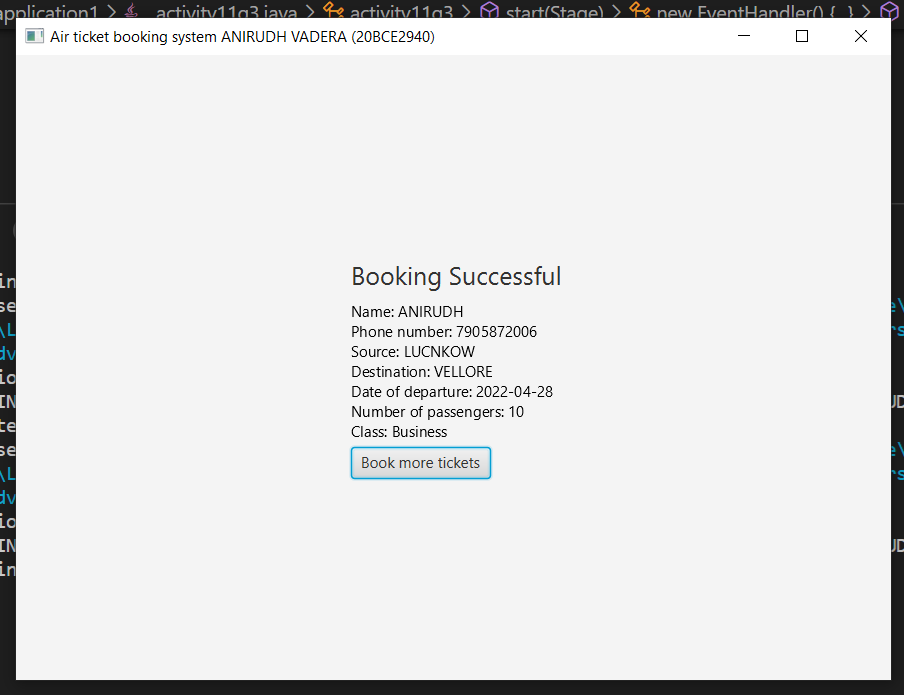
****

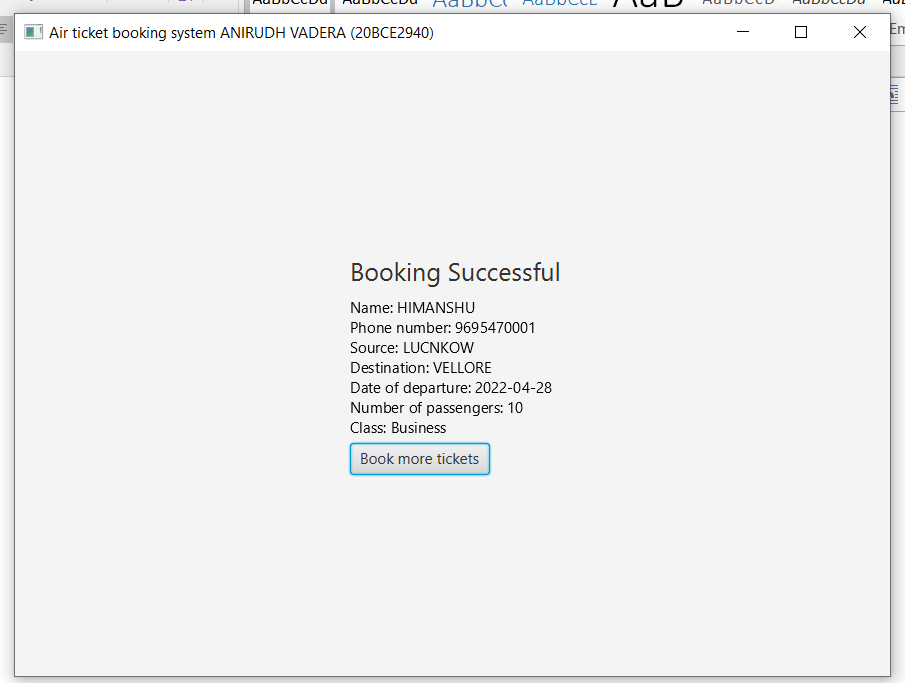
****

**Booking the tickets:**

****

**SUCCESFULL:**

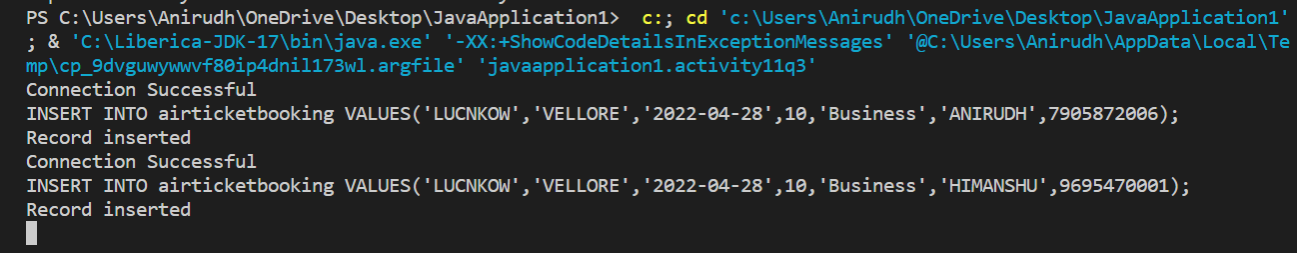
****

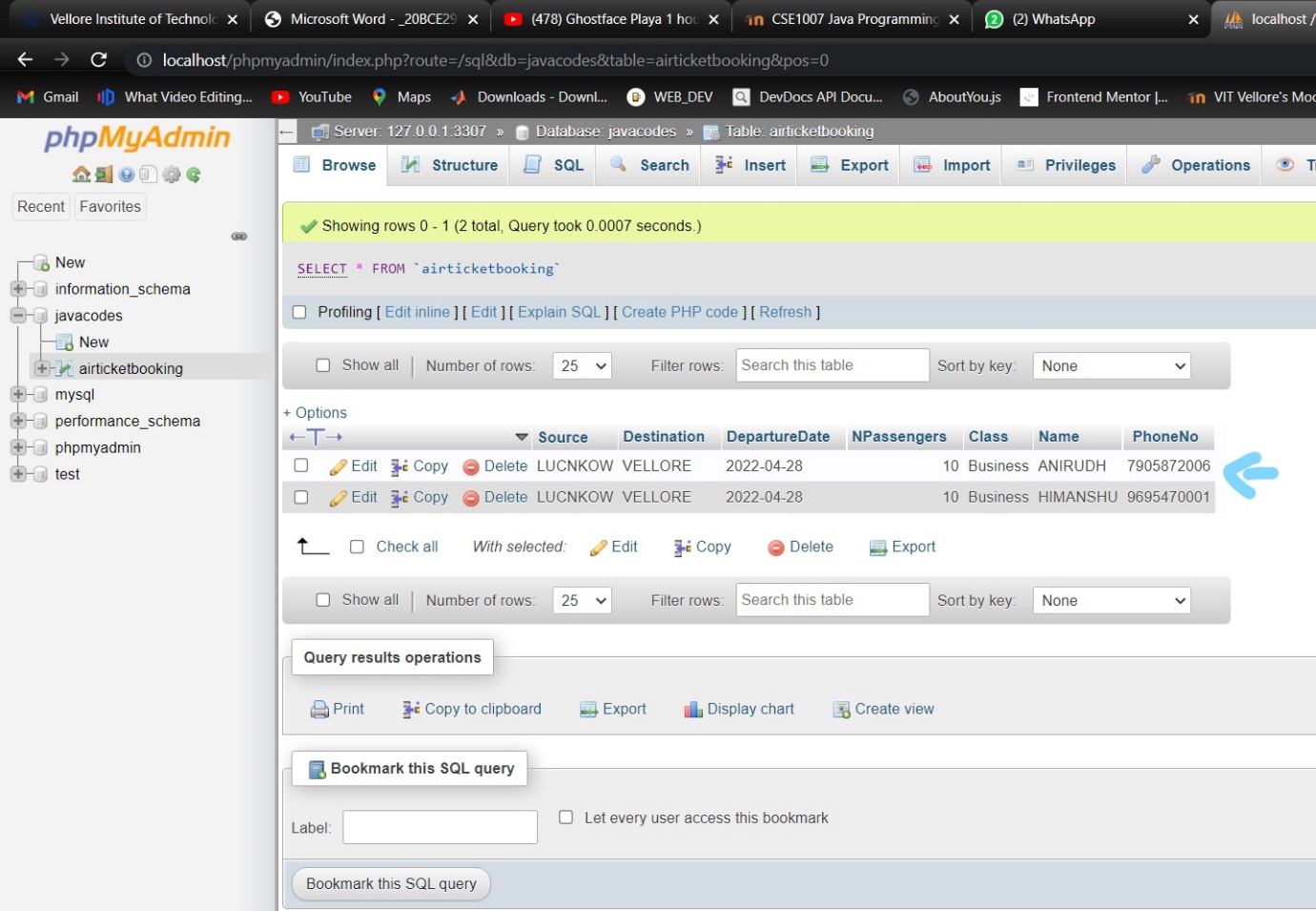
****

**UNSUCCESFULL:**

****

**After adding the tickets: the table is as follows:**

****

****