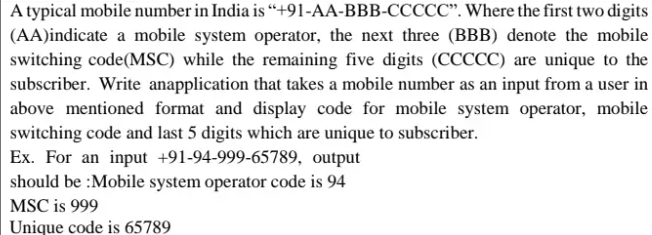
COMPLETE FILE

PART:1

# PRACTICAL :2



**CODE:**

import java.util.Scanner;

public class prac1\_2

{

    public static void main(String[] args)

    {

        try(Scanner sc = new Scanner(System.in))

        {

            String str = sc.nextLine();

            sc.close();

        operatorcode(str);

        }

    }

    public static void operatorcode(String str)

    {

        String [] code = str.split("-");

        System.out.println("Mobile system operator code is " + code[1]);

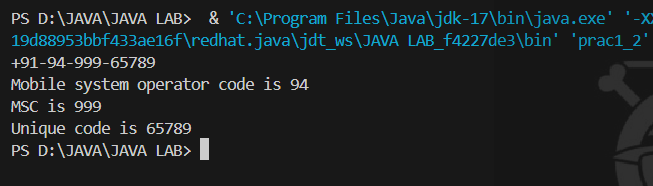
        System.out.println("MSC is " + code[2]);

        System.out.println("Unique code is " + code[3]);

    }

}

# Output :



# PRACTICAL :3

# 

# 

**CODE:**

iimport java.util.Scanner;

public class prac3 {

  public static void PrintCalender(int year, int month)

  {

    int year\_code, month\_code=0, century\_code=0, leap\_year\_code, day;

    int month\_loop=0;

    // year code

    int YY = year % 100;

    year\_code = (YY + (YY / 4));

    year\_code = year\_code % 7;

    // month code

    if (month == 1) month\_code = 0;

    else if (month == 2) month\_code = 3;

    else if (month == 3) month\_code = 3;

    else if (month == 4) month\_code = 6;

    else if (month == 5) month\_code = 1;

    else if (month == 6) month\_code = 4;

    else if (month == 7) month\_code = 6;

    else if (month == 8) month\_code = 2;

    else if (month == 9) month\_code = 5;

    else if (month == 10) month\_code = 0;

    else if (month == 11) month\_code = 3;

    else if (month == 12) month\_code = 5;

    // century code

    if (year / 100 == 17) century\_code = 4;

    else if (year / 100 == 18) century\_code = 2;

    else if (year / 100 == 19) century\_code = 0;

    else if (year / 100 == 20) century\_code = 6;

    else if (year / 100 == 21) century\_code = 4;

    else if (year / 100 == 22) century\_code = 2;

    else if (year / 100 == 23) century\_code = 0;

    // leap year code

    if (year % 400 == 0)

        leap\_year\_code = 1;

    else if (year % 100 == 0)

        leap\_year\_code = 0;

    else if (year % 4 == 0)

        leap\_year\_code = 1;

    else

        leap\_year\_code = 0;

        if (month == 1 || month == 3 || month == 5 || month == 7 || month == 8 || month == 10 || month == 12)

        month\_loop = 31;

        else if (month == 4 || month == 6 || month == 9 || month == 11)

        month\_loop = 30;

        else if (month == 2)

        if (leap\_year\_code == 1)

            month\_loop = 29;

        else

            month\_loop = 28;

    for(int i=1; i<=month\_loop; i++)

    {

        day = (year\_code + month\_code + century\_code + i - leap\_year\_code);

        day = day % 7;

        System.out.print("");

        if(i==1)

        {

            if(day==1) System.out.print("      ");

            else if(day==2) System.out.print("            ");

            else if(day==3) System.out.print("                  ");

            else if(day==4) System.out.print("                        ");

            else if(day==5) System.out.print("                              ");

            else if(day==6) System.out.print("                                    ");

        }

        if(1<=i && i<=9) System.out.print(" "+i+"    ");

        else System.out.print(i+"    ");

        if(day==6) System.out.println();

    }

  }

  public static void main(String[] args) {

    int year, month;

    Scanner scan = new Scanner(System.in);

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

    year = scan.nextInt();

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

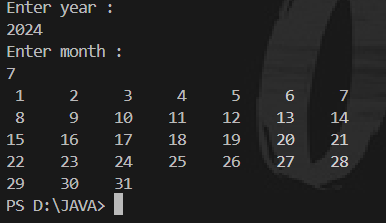
    month = scan.nextInt();

    PrintCalender(year, month);

  }

}

**OUTPUT:**



# PRACTICAL :4

# 

**CODE:**

import java.util.\*;

public class prac1\_4

{

    private static char[] Alphabet =

        {

            'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n',

            'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'

        };

    private static String [] MORSELETTERS =

        {

            ".-", "-...", "-.-.", "-..", ".", "..-.", "--.", "....", "..", ".---",

            "-.-", ".-..", "--", "-.", "---", ".--.", "--.-", ".-.", "...", "-",

            "..-", "...-", ".--", "-..-", "-.--", "--.."

        };

    public static void main(String[] args)

    {

        Scanner sc = new Scanner(System.in);

        int num;

        while(true)

        {

            System.out.println("\n\nEnter 1 for String to Morse code conversation");

            System.out.println("Enter 2 for Morse code to String conversation");

            System.out.println("Enter 3 for exit");

            num = sc.nextInt();

            sc.nextLine();

            switch(num)

            {

                case 1:

                StringtoMorse(sc);

                break;

                case 2:

                MorseTostring(sc);

                break;

                case 3:

                System.out.println("Thank you for using Morse Code Converter!");

                System.exit(0);

            }

        }

    }

    static void StringtoMorse(Scanner sc)

    {

        System.out.print("\nEnter a String : ");

        String str = sc.nextLine();

        str = str.toLowerCase();

        System.out.print("Morse code :");

        for(char c : str.toCharArray())

        {

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

            {

                if(c == Alphabet[i])

                {

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

                    break;

                }

            }

        }

    }

    static void MorseTostring(Scanner sc)

    {

        System.out.print("\nEnter Morse Code : ");

        String str = sc.nextLine();

        String [] morseWords = str.split(" ");

        System.out.print("String :");

        for(String word : morseWords)

        {

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

            {

                if(word.equals(MORSELETTERS[i]))

                {

                    System.out.print(Alphabet[i]);

                    break;

                }

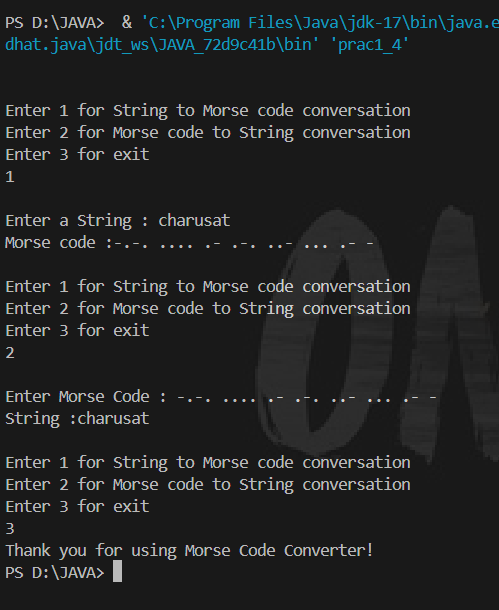
            }

        }

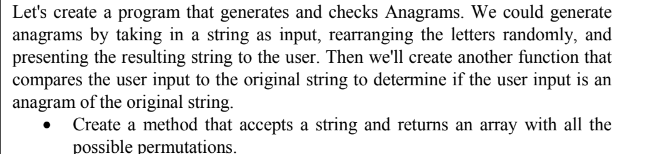
    }

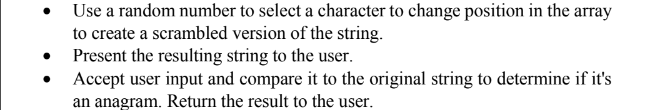
}

**OUTPUT:**



# PRACTICAL :5





**CODE:**

import java.util.\*;

public class prac1\_5

{

    public static void main(String[] args)

    {

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

        try(Scanner scan = new Scanner(System.in))

        {

        String str = scan.nextLine();

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

        System.out.print("Enter your guess : ");

        String userstr = scan.nextLine();

        System.out.print(isAnagram(str, userstr));

        }

    }

    static String Scramble(String str)

    {

        int n;

        Random rand = new Random();

        char[] NewCharArray = str.toCharArray();

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

        {

            n = rand.nextInt(NewCharArray.length);

            char temp = NewCharArray[n];

            NewCharArray[n] = NewCharArray[i];

            NewCharArray[i] = temp;

        }

        return new String(NewCharArray);

    }

    static boolean isAnagram(String OriginalStr, String UserInput)

    {

        int[] CharCount = new int[26];

        if (OriginalStr.length() != UserInput.length())

        {

            return false;

        }

        for (char c : OriginalStr.toCharArray())

        {

            CharCount[c - 'a']++;

        }

        for (char c : UserInput.toCharArray())

        {

            if (CharCount[c - 'a'] == 0)

            {

                return false;

            }

            CharCount[c - 'a']++;

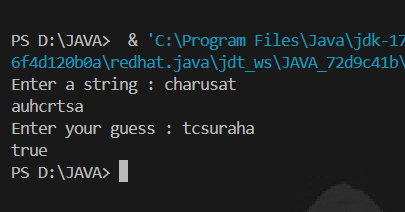
        }

        return true;

    }

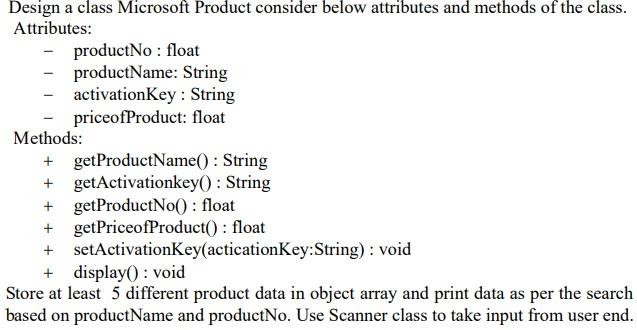
}

**OUTPUT:**



PART 2

# PRACTICAL :1



**CODE:**

import java.util.\*;

class Microsoft

{

    private float productNo;

    private String productName;

    private String activationKey;

    private float priceofProduct;

    public float getProductNo(Scanner sc)

    {

        this.productNo = sc.nextFloat();

        return productNo;

    }

    public String getProductName(Scanner sc)

    {

        sc.nextLine();

        this.productName = sc.nextLine();

        return productName;

    }

    public void setActivationKey(Scanner sc)

    {

        this.activationKey = sc.nextLine();

        getActivationKey();

    }

    public String getActivationKey()

    {

        return activationKey;

    }

    public float getPriceofProduct(Scanner sc)

    {

        this.priceofProduct = sc.nextFloat();

        return priceofProduct;

    }

    public void search(String temp\_name , float temp\_num)

    {

        if(temp\_name.equals(productName) && temp\_num == productNo)

        {

            display();

        }

        else

        {

            System.out.println("\nProduct not found");

        }

    }

    public void display()

    {

        System.out.println("\n\nProduct number : " + productNo);

        System.out.println("Product name : " + productName);

        System.out.println("Product price : " + priceofProduct);

        System.out.println("Product activation key : " + activationKey);

    }

};

public class prac2\_1

{

    public static void main(String[] args)

    {

        try(Scanner sc = new Scanner(System.in))

        {

            Microsoft m = new Microsoft();

            System.out.println("WELCOME TO MICROSOFT STORE ");

            while(true)

            {

                System.out.println("\nEnter 1 to take input of the data ");

                System.out.println("Enter 2 to print the data ");

                System.out.println("Enter 3 to exit ");

                System.out.print("\nEnte your choice : ");

                int a = sc.nextInt();

                switch(a)

                {

                    case 1 :

                    {

                        System.out.print("\n\nEnter the number of times you want to enter the data : ");

                        int p = sc.nextInt();

                        sc.nextLine();

                        for(int i = 0 ; i < p ; i++)

                        {

                            System.out.print("\n\nEnter the product number : ");

                            m.getProductNo(sc);

                            System.out.print("Enter the product name : ");

                            m.getProductName(sc);

                            System.out.print("Enter the product activationkey : ");

                            m.setActivationKey(sc);

                            System.out.print("Enter the product price : ");

                            m.getPriceofProduct(sc);

                            sc.nextLine();

                        }

                        break;

                    }

                    case 2:

                    {

                        System.out.print("Enter the product number : ");

                        float temp\_num = sc.nextFloat();

                        sc.nextLine();

                        System.out.print("Enter the product name : ");

                        String temp\_name = sc.nextLine();

                        m.search(temp\_name,temp\_num);

                        break;

                    }

                    case 3:

                    {

                        System.exit(0);

                    }

                }

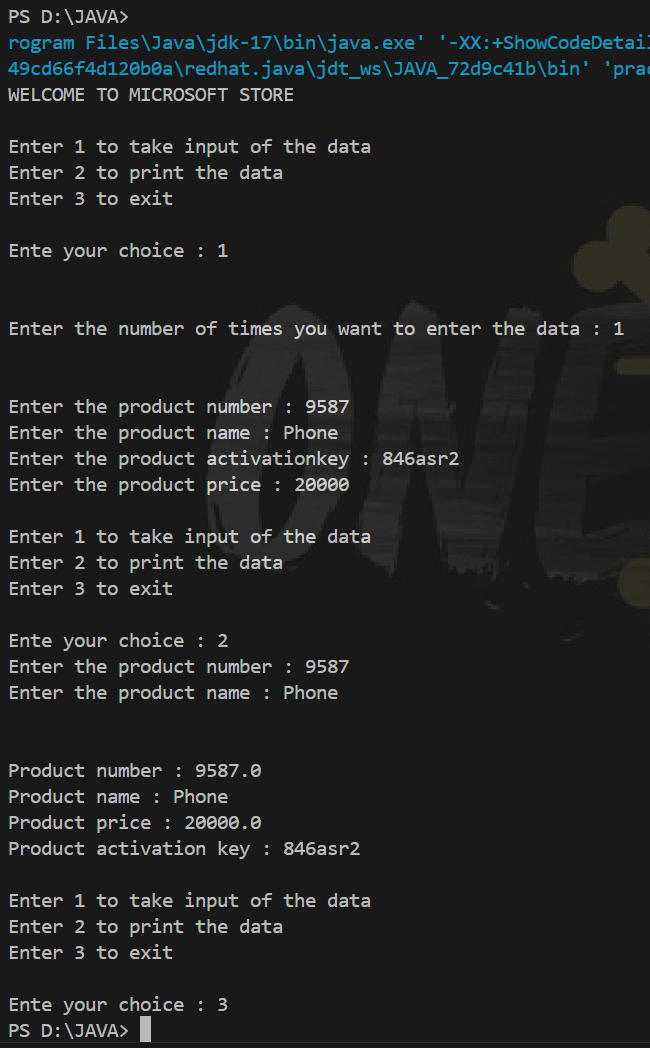
            }

        }

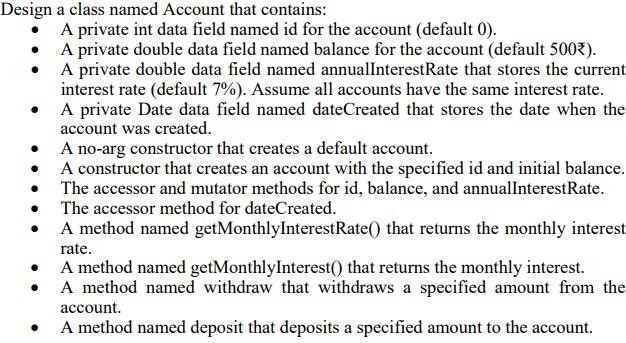
    }

}

**OUTPUT:**



# PRACTICAL :2



**CODE:**

import java.util.\*;

class Account

{

    private int id;

    private double balance;

    private double interest\_rate;

    private Date d = new Date();

    public Account()

    {

        this.id = 0;

        this.balance = 500;

        this.interest\_rate = 7;

    }

    public Account(int id, double balance, double interest\_rate)

    {

        this.id = id;

        this.balance = balance;

        this.interest\_rate = interest\_rate;

    }

    public int getId()

    {

        return id;

    }

    public double getBalance()

    {

        return balance;

    }

    public double getInterest\_rate()

    {

        return interest\_rate;

    }

    public void getMonthlyInterestRate()

    {

        System.out.println("Monthly Interest Rate: " + interest\_rate / 12);

    }

    public void getMonthlyInterest()

    {

        System.out.println("Monthly Interest: " + balance \* interest\_rate / 1200);

    }

    public void withdraw(double amount)

    {

        if (amount > balance)

        {

            System.out.println("Insufficient funds");

        }

        else

        {

            balance = balance - amount;

            System.out.println("Withdrawal successful");

        }

        System.out.println("Current balace " + balance);

    }

    public void deposit(double amount)

    {

        balance = balance + amount;

        System.out.println("Deposit successful");

        System.out.println("Current balace " + balance);

    }

}

public class pra2\_2

{

    public static void main(String[] strg)

    {

        try (Scanner sc = new Scanner(System.in))

        {

            System.out.print("Enter the id of the account : ");

            int id = sc.nextInt();

            System.out.print("Enter the balance of the account : ");

            double balance = sc.nextDouble();

            System.out.print("Enter the interest rate of the account : ");

            double interest\_rate = sc.nextDouble();

            Account a = new Account(id, balance, interest\_rate);

            a.getMonthlyInterestRate();

            a.getMonthlyInterest();

            System.out.println("Enter 1 to withdraw money");

            System.out.println("Enter 2 to deposit money ");

            System.out.println("Enter 3 to exit");

            System.out.print("Enter your choice : ");

            int choice = sc.nextInt();

            switch (choice)

            {

                case 1:

                    System.out.print("Enter the amount to withdraw : ");

                    double amount = sc.nextDouble();

                    a.withdraw(amount);

                    break;

                case 2:

                    System.out.print("Enter the amount to deposit : ");

                    double amount1 = sc.nextDouble();

                    a.deposit(amount1);

                    break;

                case 3:

                    System.out.println("Thank you");

                    System.exit(0);

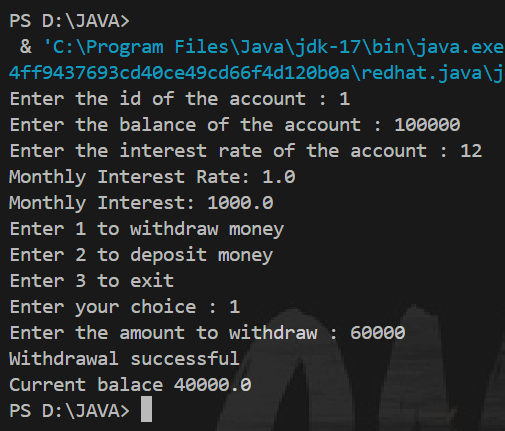
                    break;

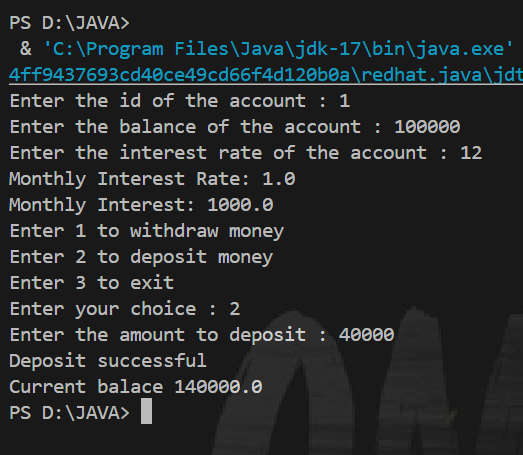
            }

        }

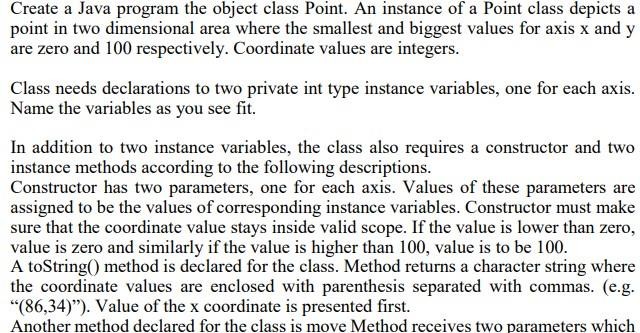
    }

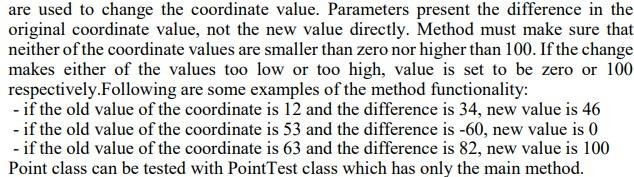
**OUTPUT:**





# PRACTICAL :3





**CODE:**

import java.util.Scanner;

class point

{

    private int x;

    private int y;

    public point(int x, int y)

    {

        if(x > 100)

        {

            this.x = 100;

        }

        else if(x < 0)

        {

            this.x = 0;

        }

        else

        {

            this.x = x;

        }

        if(y > 100)

        {

            this.y = 100;

        }

        else if(y < 0)

        {

            this.y = 0;

        }

        else

        {

            this.y = y;

        }

    }

    String tostring()

    {

        return "(" + x + "," + y + ")";

    }

    void move(int x, int y)

    {

        this.x += x;

        this.y += y;

        if(x > 100)

        {

            this.x = 100;

        }

        else if(x < 0)

        {

            this.x = 0;

        }

        if(y > 100)

        {

            this.y = 100;

        }

        else if(y < 0)

        {

            this.y = 0;

        }

    }

}

public class prac2\_3

{

    public static void main(String[] args)

    {

        try(Scanner sc = new Scanner (System.in))

        {

            int x, y;

            System.out.print("Enter the x coordinate of the point : ");

            x = sc.nextInt();

            System.out.print("Enter the y coordinate of the point : ");

            y = sc.nextInt();

            point p = new point(x,y);

            System.out.println("Coordinates are : " + p.tostring());

            System.out.print("Enter the difference between the x coordinate : ");

            x = sc.nextInt();

            System.out.print("Enter the difference between the y coordinate : ");

            y = sc.nextInt();

            sc.close();

            p.move(x, y);

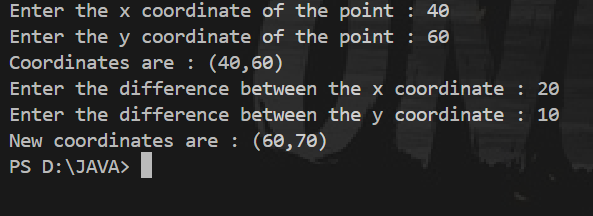
            System.out.print("New coordinates are : " + p.tostring());

        }

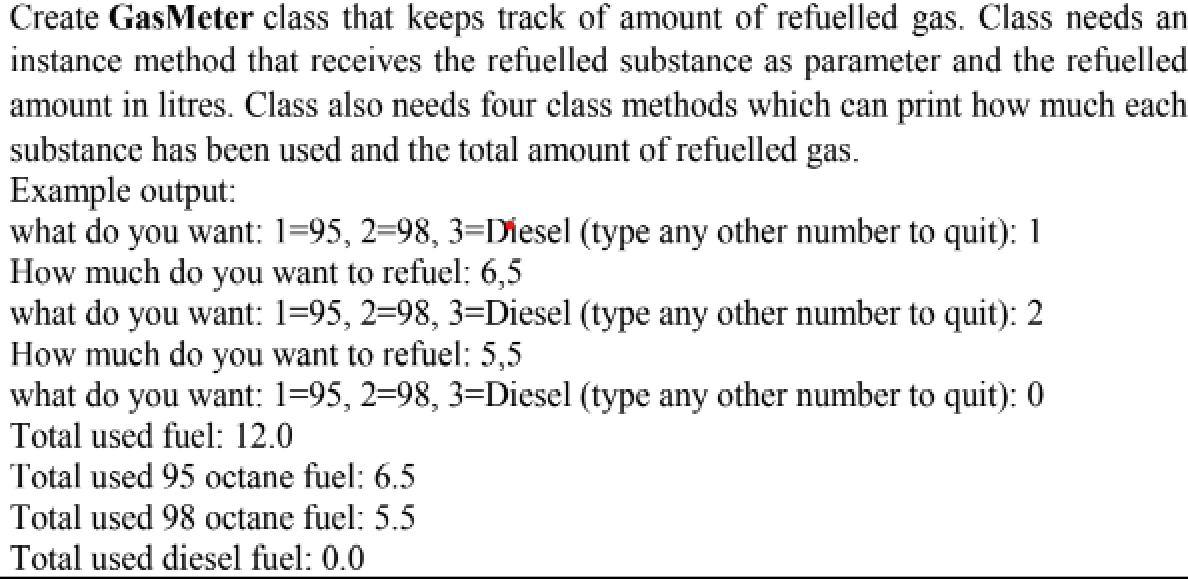
    }

}

**OUTPUT:**



# PRACTICAL :4



**CODE:**

import java.util.Scanner;

class GasMeter

{

    private float Fuel\_95;

    private float Fuel\_98;

    private float Fuel\_diesel;

    private float Fuel\_total;

    void calculate\_95(float get\_95)

    {

        Fuel\_95 = get\_95;

        Fuel\_total += get\_95;

    }

    void calculate\_98(float get\_98)

    {

        Fuel\_98 = get\_98;

        Fuel\_total += get\_98;

    }

    void calculate\_diesel(float get\_diesel)

    {

        Fuel\_diesel = get\_diesel;

        Fuel\_total += get\_diesel;

    }

    void display()

    {

        System.out.println("Total used fuel : " + Fuel\_total);

        System.out.println("Total used 95 octane fuel : " + Fuel\_95);

        System.out.println("Total used 98 octane fuel : " + Fuel\_98);

        System.out.println("Total used diesel fuel : " + Fuel\_diesel);

    }

}

public class prac2\_4

{

    public static void main(String[] args)

    {

        try(Scanner scanner = new Scanner(System.in))

        {

            GasMeter gas = new GasMeter();

            int choice;

            float get\_95, get\_98, get\_diesel;

            while(true)

            {

                System.out.print("what do you want: 1=95, 2=98, 3=Diesel (type any other number to quit) : ");

                choice = scanner.nextInt();

                switch(choice)

                {

                    case 1 :

                    {

                        System.out.print("How much do you want to refuel : ");

                        get\_95 = scanner.nextFloat();

                        gas.calculate\_95(get\_95);

                        break;

                    }

                    case 2 :

                    {

                        System.out.print("How much do you want to refuel : ");

                        get\_98 = scanner.nextFloat();

                        gas.calculate\_98(get\_98);

                        break;

                    }

                    case 3 :

                    {

                            System.out.print("How much do you want to refuel : ");

                        get\_diesel = scanner.nextFloat();

                        gas.calculate\_diesel(get\_diesel);

                        break;

                    }

                }

                if (choice < 1 || choice > 3)

                {

                    break;

                }

            }

        gas.display();

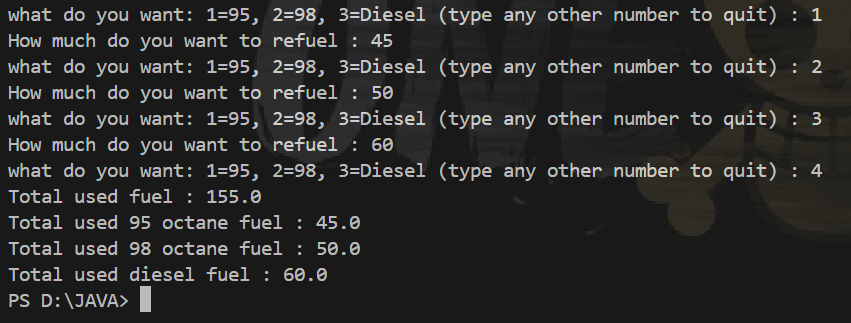
        scanner.close();

        }

    }

}

**OUTPUT:**



# PRACTICAL :5

**CODE:**

import java.util.\*;

class Overloading

{

    public String stringM(String str)

    {

        str = str.toUpperCase();

        str = str.replace('A', 'Z');

        System.out.println(str);

        return str;

    }

    public String stringM(String str, int a)

    {

        if(str.contains(" ") && a<=10)

        {

            String[] str1 = str.split(" ");

            String str2 = str1[0];

            String str3 = str1[1];

            str = str2 + str3.replace( str3, "CHARUSAT");

            System.out.println(str);

            return str;

        }

        else if(str.contains(" ") && a > 10)

        {

            str = str.toLowerCase();

            System.out.println(str);

            return str;

        }

        else

        {

            return "Invalid output";

        }

    }

}

public class prac2\_5

{

    public static void main(String[] args)

    {

        try(Scanner sc = new Scanner(System.in))

        {

            Overloading o = new Overloading();

            while(true)

            {

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

                String str = sc.nextLine();

                int a = str.length();

                if(!str.contains(" "))

                {

                    o.stringM(str);

                    System.out.println("Length of string is : " + a);

                }

                else

                {

                    o.stringM(str , a);

                }

                System.out.print("Do you want to continue (y/n) ? ");

                char ch = sc.next().charAt(0);

                sc.nextLine();

                if(ch == 'Y' || ch =='y') {}

                else break;

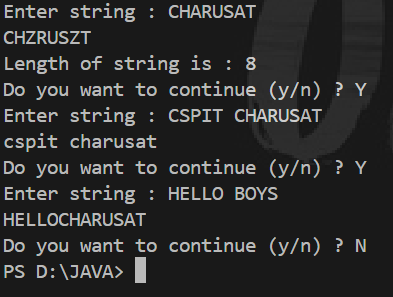
            }

        }

    }

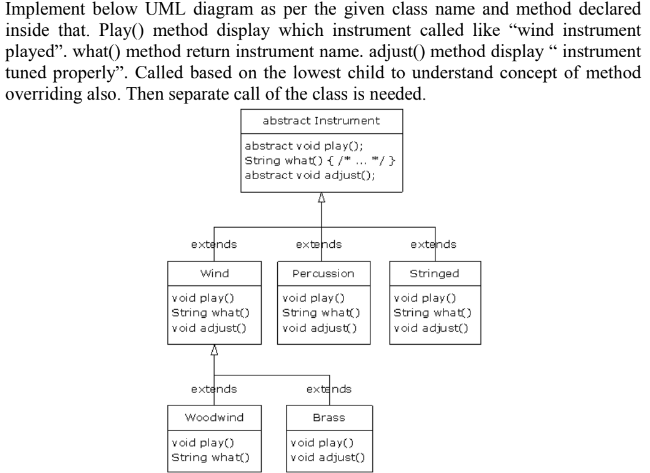
}

**OUTPUT:**



PART:3

# PRACTICAL :1



**CODE:**

abstract class instrument

{

    abstract void play();

    String what()

    {

        return "instrument";

    }

    abstract void adjust();

}

class Wind extends instrument

{

    @Override

    void play()

    {

        System.out.println("Wind instrument played");

    }

    @Override

    String what()

    {

        return "Wind";

    }

    @Override

    void adjust()

    {

        System.out.println("Instrument tuned properly");

    }

}

class Percusion extends instrument

{

    @Override

    void play()

    {

        System.out.println("Percusion instrument played");

    }

    @Override

    String what()

    {

        return "Percusion";

    }

    @Override

    void adjust()

    {

        System.out.println("Instrument tuned properly");

    }

}

class Stringed extends instrument

{

    @Override

    void play()

    {

        System.out.println("Stringed instrument played");

    }

    @Override

    String what()

    {

        return "Stringed";

    }

    @Override

    void adjust()

    {

        System.out.println("Instrument tuned properly");

    }

}

class Woodwind extends Wind

{

    @Override

    void play()

    {

        System.out.println("WoodWind instrument played");

    }

    @Override

    String what()

    {

        return "WoodWind";

    }

}

class Brass extends Wind

{

    @Override

    void play()

    {

        System.out.println("Brass instrument played");

    }

    @Override

    void adjust()

    {

        System.out.println("Instrument tuned properly");

**}**

}

public class prac3\_1

{

    public static void main(String[] args)

    {

        Woodwind w = new Woodwind();

        Brass b = new Brass();

        w.play();

        w.what();

        System.out.println(w.what());

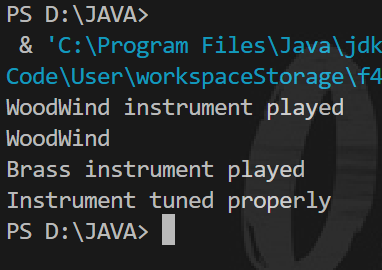
        b.play();

        b.adjust();

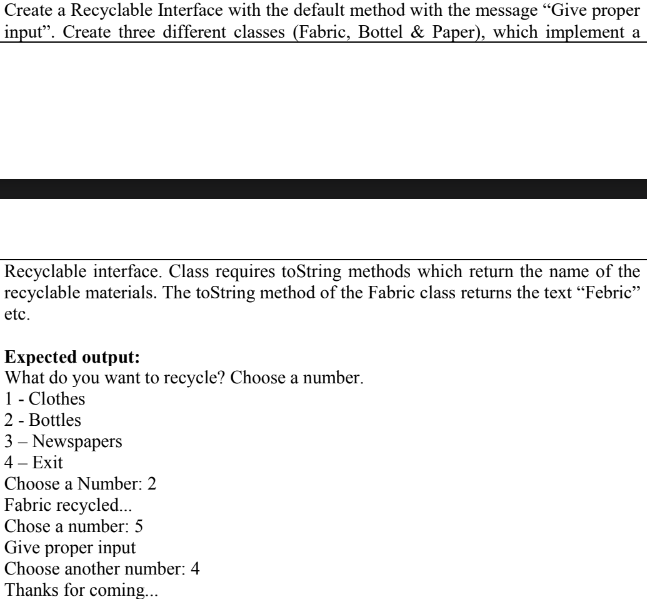
    }

}

**OUTPUT:**



# PRACTICAL :2



**CODE:**

import java.util.Scanner;

interface Recyclable

{

    default void display()

    {

        System.out.println("Give proper input");

    }

}

class Fabric implements Recyclable

{

@Override

    public String toString()

    {

        return "Fabric recycled...";

    }

}

class Bottel implements Recyclable

{

@Override

    public String toString()

    {

        return "Bottel recycled...";

    }

}

class Paper implements Recyclable

{

@Override

    public String toString()

    {

        return "Paper recycled...";

    }

}

public class prac3\_2

{

    public static void main(String[] args)

    {

        try(Scanner sc = new Scanner(System.in))

        {

            Fabric f = new Fabric();

            Bottel b = new Bottel();

            Paper p = new Paper();

            int a;

            while(true)

            {

                System.out.println("What do you want to recycle? Choose a number.");

                System.out.println("1 - Clothes");

                System.out.println("2 - Bottles");

                System.out.println("3 - Newspapers");

                System.out.println("4 - Exit");

                System.out.print("Choose a Number : ");

                a = sc.nextInt();

                switch(a)

                {

                    case 1:

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

                        break;

                    case 2:

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

                        break;

                    case 3:

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

                        break;

                    case 4:

                        System.out.println("Thanks for coming...");

                        System.exit(0);

                    default:

                        f.display();

                        System.out.print("Choose another number : ");

                        a = sc.nextInt();

                        if (a < 1 || a > 3)

                        {

                            break;

                        }

                }

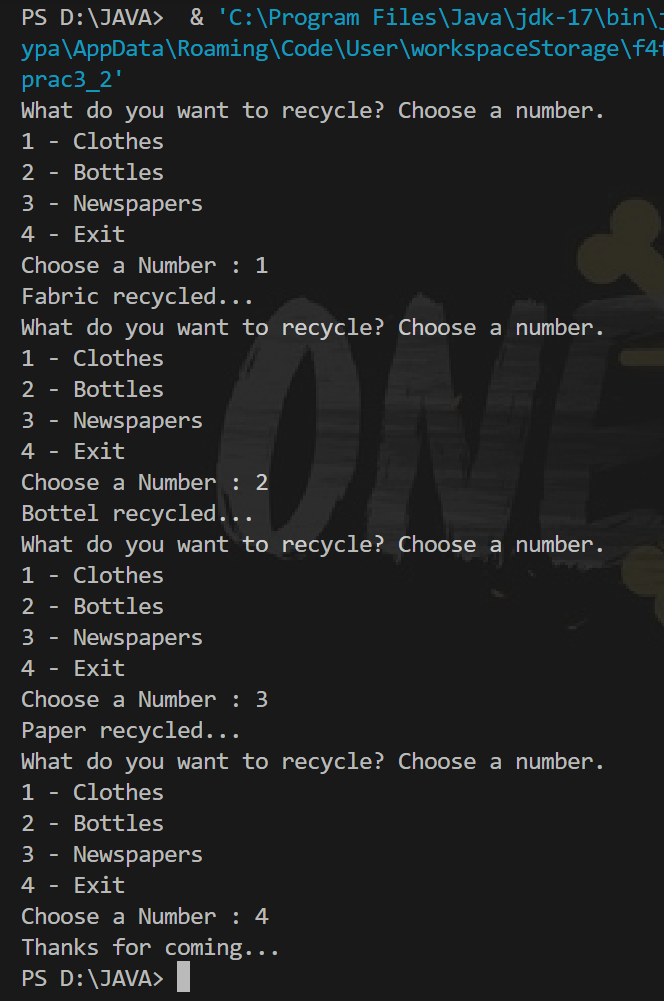
            }

        }

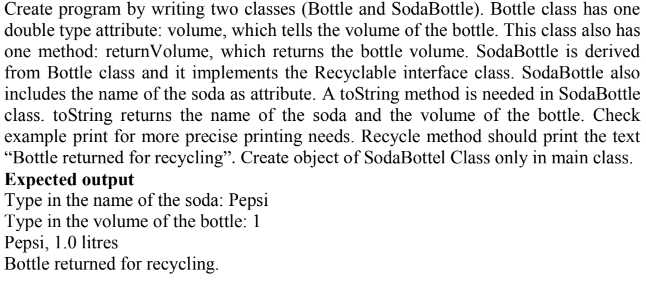
    }

}

# OUTPUT:



**PRACTICAL :3**



**CODE:**

import java.util.Scanner;

interface Recylable

{

    default void display()

    {

        System.out.println("Bottle returned for recycling");

    }

}

class Bottle

{

    private double volume;

    public double returnVolume()

    {

        return volume;

    }

}

class SodaBottle extends Bottle implements Recylable

{

    private String soda\_name;

@Override

    public String toString()

    {

        return soda\_name + returnVolume();

    }

}

public class prac3\_3

{

    public static void main(String[] args)

    {

        try(Scanner sc = new Scanner(System.in))

        {

            SodaBottle sb = new SodaBottle();

            double volume;

            String soda\_name;

            System.out.print("Type in the name of the soda : ");

            soda\_name = sc.nextLine();

            System.out.print("Type in the volume of the bottle : ");

            volume = sc.nextDouble();

            sb.toString();

            System.out.println(soda\_name + "," + volume +"liters" );

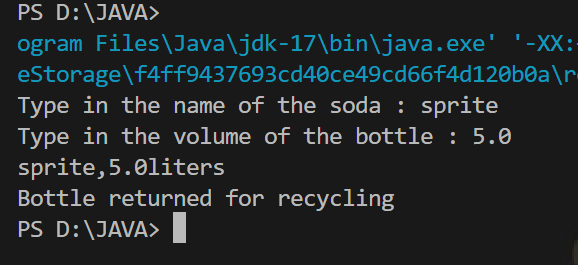
            sb.display();

        }

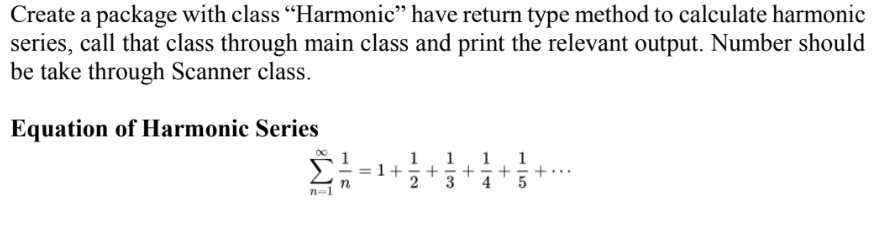
    }

}

**OUTPUT:**



# PRACTICAL: 4



**Package definition :**

package harmonic;

public class harmonic

{

    public double harmonic1(int n)

    {

        double sum=0;

        for(int i=1;i<=n;i++)

        {

            sum=sum+(1.0/i);

        }

        return sum;

    }

}

# MAIN CODE:

import harmonic.harmonic;

import java.util.Scanner;

public class prac3\_4

{

    public static void main(String[] args)

    {

        try (Scanner sc = new Scanner(System.in))

        {

            System.out.print("Enter the number of terms : ");

            int n = sc.nextInt();

            harmonic h =new harmonic();

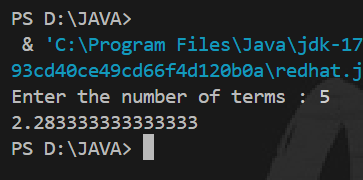
            System.out.println(h.harmonic1(n));

        }

    }

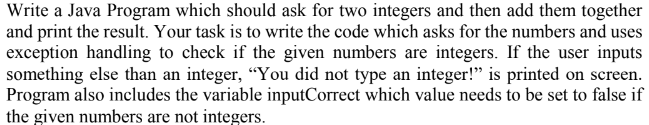
}

# OUTPUT:



PART:4

# PRACTICAL :1



**CODE:**

import java.util.Scanner;

public class prac4\_1

{

    public static void main(String[] args)

    {

        try(Scanner sc = new Scanner(System.in))

        {

            int a = 0, b = 2;

            try

            {

                System.out.print("Enter the first number : ");

                a = sc.nextInt();

                System.out.print("Enter the second number : ");

                b = sc.nextInt();

            }catch(Exception e)

            {

                System.out.println("Invalid unput");

            }

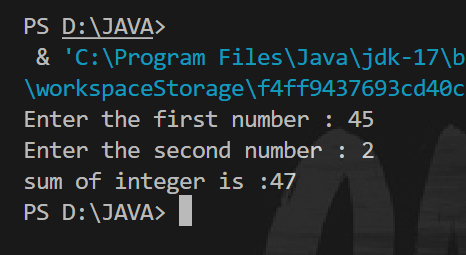
            System.out.println("sum of integer is :" + (a + b));

        }

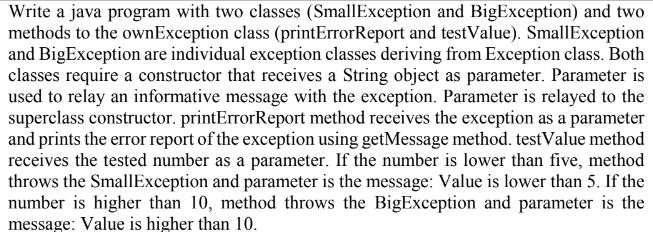
    }

}

**OUTPUT:**



# PRACTICAL :2



**CODE:**

import java.util.Scanner;

class SmallException extends Exception

{

    SmallException(String str)

    {

        super(str);

    }

}

class BigException extends Exception

{

    BigException(String str)

    {

        super(str);

    }

}

class OwnException

{

    public void printErrorReport(Exception e)

    {

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

    }

    public void testValue(int x) throws SmallException, BigException

    {

        if (x < 5)

        {

            throw new SmallException("Value is smaller than 5.");

        }

        else if (x > 10)

        {

            throw new BigException("Value is bigger than 10.");

        }

        else

        {

            System.out.println("The value is between 5 and 10");

        }

    }

}

public class prac4\_2

{

    public static void main(String[] args)

    {

        OwnException obj = new OwnException();

        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number : ");

        int a = sc.nextInt();

        try

        {

            obj.testValue(a);

        }

        catch (SmallException | BigException e)

        {

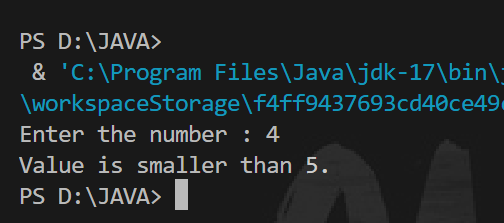
            obj.printErrorReport(e);

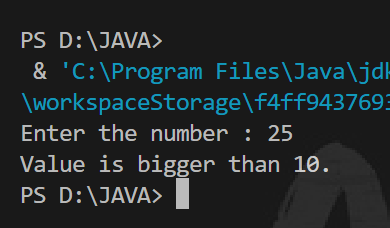
        }

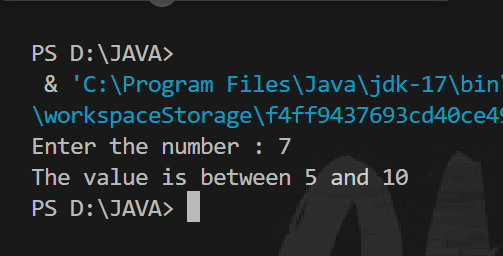
    }

}

# OUTPUT:

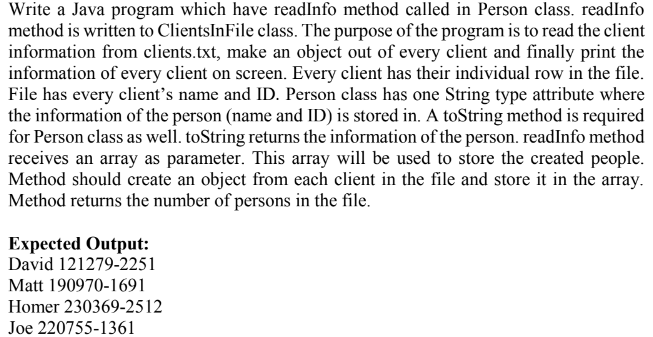






PART:5

# PRACTICAL :1



**CODE:**

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

class Person

{

    private final String info;

    public Person(String info)

    {

        this.info = info;

    }

    @Override

    public String toString()

    {

        return info;

    }

}

class ClientsInFile

{

    int readInfo(Person[] persons)

    {

        int count = 0;

        String filename = "D:\\JAVA\\JAVA LAB\\clients.txt";

        try (BufferedReader br = new BufferedReader(new FileReader(filename)))

        {

            String line;

            while ((line = br.readLine()) != null && count < persons.length)

            {

                persons[count] = new Person(line);

                count++;

            }

        } catch (IOException e)

        {

            System.out.println("Error handeling file " + e.getMessage());

        }

        return count;

    }

}

public class prac5\_1

{

    public static void main(String[] args)

    {

        ClientsInFile clf = new ClientsInFile();

        Person p[] = new Person[5];

        int clients = clf.readInfo(p);

        for (int i = 0; i < clients; i++)

        {

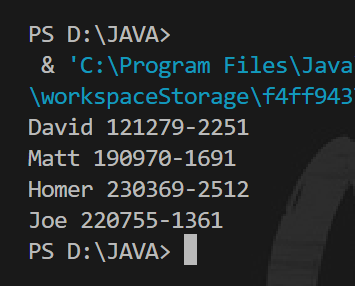
            System.out.println(p[i].toString());

        }

    }

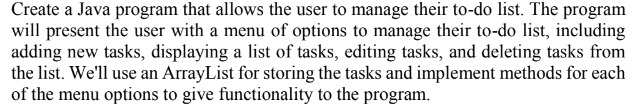
}

**OUTPUT:**



PART:7

# PRACTICAL :1



**CODE:**

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Task

{

    String task;

    boolean isDone;

    public Task(String task)

    {

        this.task = task;

        this.isDone = false;

    }

}

public class pract7\_1

{

    static Scanner sc = new Scanner(System.in);

    static List<Task> tasks = new ArrayList<>();

    public static void main(String[] args)

    {

        while (true)

        {

            printMenu();

            int choice = getChoice();

            todoList(choice);

        }

    }

    private static int getChoice()

    {

        System.out.print("Choose an option: ");

        return sc.nextInt();

    }

    private static void printMenu()

    {

        System.out.println("\nTo-Do List Menu:");

        System.out.println("1. Add Task");

        System.out.println("2. Display Tasks");

        System.out.println("3. Edit Task");

        System.out.println("4. Delete Task");

        System.out.println("5. Exit");

    }

    private static void todoList(int choice)

    {

        switch (choice)

        {

            case 1 -> addTask();

            case 2 -> displayTasks();

            case 3 -> editTask();

            case 4 -> deleteTask();

            case 5 -> System.exit(0);

            default -> System.out.println("Invalid option. Please choose again.");

        }

    }

    private static void addTask()

    {

        System.out.print("Enter task name: ");

        sc.nextLine(); // Consume newline left-over

        tasks.add(new Task(sc.nextLine()));

        System.out.println("Task added successfully!");

    }

    private static void displayTasks()

    {

        if (tasks.isEmpty())

        {

            System.out.println("No task available.");

            return;

        }

        System.out.println("\nYour Tasks: ");

        for (int i = 0; i < tasks.size(); i++)

        {

            Task task = tasks.get(i);

            System.out.println((i + 1) + ". " + task.task + " [" + (task.isDone ? "Completed" : "NotCompleted") + "]");

        }

    }

    private static void editTask()

    {

        displayTasks();

        System.out.print("Enter Task number to edit: ");

        int n = sc.nextInt();

        sc.nextLine();

        if (n < 1 || n > tasks.size())

        {

            System.out.println("Enter Valid Number.");

            return;

        }

        System.out.print("Enter new Task Name: ");

        tasks.get(n - 1).task = sc.nextLine();

        System.out.println("Task updated successfully!");

    }

    private static void deleteTask()

    {

        displayTasks();

        System.out.print("Enter Task number to delete: ");

        int n = sc.nextInt();

        sc.nextLine();

        if (n < 1 || n > tasks.size())

        {

            System.out.println("Enter Valid Number.");

            return;

        }

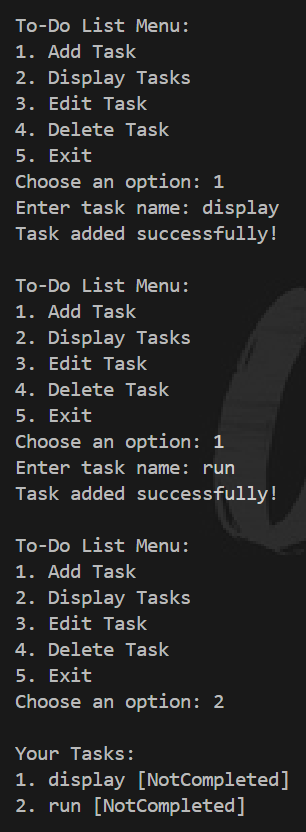
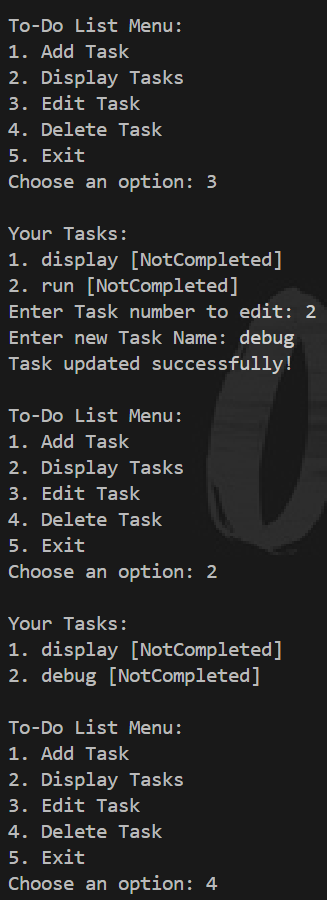
        tasks.remove(n - 1);

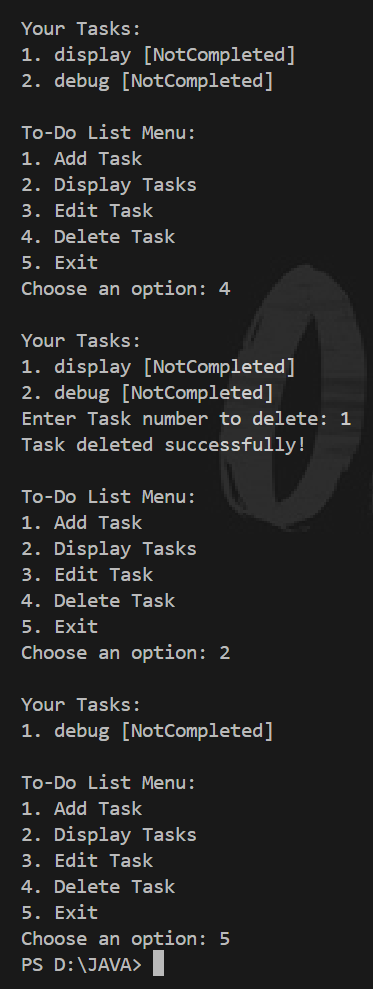
        System.out.println("Task deleted successfully!");

    }

}

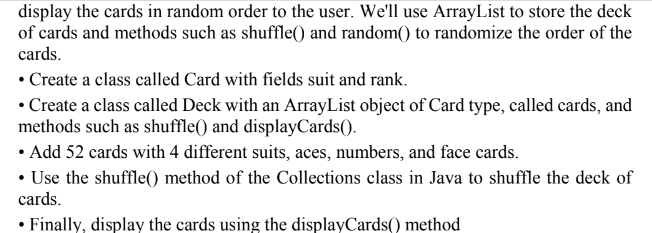
**OUTPUT:**



# PRACTICAL :2





**CODE:**

import java.util.ArrayList;

import java.util.Collections;

class Card

{

    private final String suit;

    private final String rank;

    public Card(String suit, String rank)

    {

        this.suit = suit;

        this.rank = rank;

    }

    @Override

    public String toString()

    {

        return rank + " of " + suit;

    }

}

class Deck

{

    private final ArrayList<Card> cards;

    public Deck()

    {

        this.cards = new ArrayList<>();

        createDeck();

    }

    private void createDeck()

    {

        String[] suits = { "Hearts", "Diamonds", "Clubs", "Spades" };

        String[] ranks = { "Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King" };

        for (String suit : suits)

        {

            for (String rank : ranks)

            {

                cards.add(new Card(suit, rank));

            }

        }

    }

    public void shuffle()

    {

        Collections.shuffle(cards);

    }

    public void displayCards()

    {

        for (int i = 0; i < cards.size(); i++)

        {

            System.out.println((i + 1) + ". " + cards.get(i));

        }

    }

    public int getDeckSize()

    {

        return cards.size();

    }

}

public class prac7\_2

{

    public static void main(String[] args)

    {

        Deck deck = new Deck();

        System.out.println("Initial deck size: " + deck.getDeckSize());

        System.out.println("Before shuffling:");

        deck.displayCards();

        deck.shuffle();

        System.out.println("\nAfter shuffling:");

        deck.displayCards();

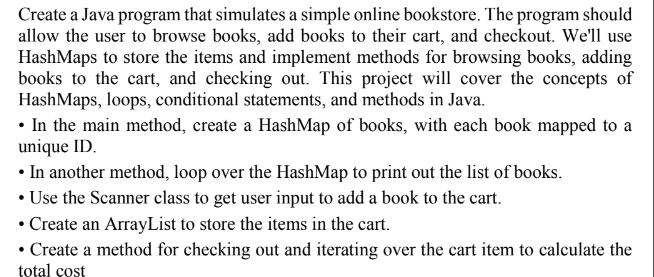
    }

}

# OUTPUT:

# 

**PRACTICAL :3**



**CODE:**

import java.util.ArrayList;

import java.util.HashMap;

import java.util.Scanner;

class Book

{

    String title;

    double price;

    public Book(String title, double price)

    {

        this.title = title;

        this.price = price;

    }

}

public class prac7\_3

{

    static HashMap<String, Book> books = new HashMap<>();

    static ArrayList<Book> cart = new ArrayList<>();

    static Scanner scanner = new Scanner(System.in);

    public static void main(String[] args)

    {

        initializeBooks();

        while (true)

        {

            System.out.println("\nOnline Bookstore Menu:");

            System.out.println("1. Browse Books");

            System.out.println("2. Add Book to Cart");

            System.out.println("3. Checkout");

            System.out.println("4. Exit");

            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();

            scanner.nextLine(); // Consume newline left-over

            switch (choice)

            {

                case 1 -> browseBooks();

                case 2 -> addBookToCart();

                case 3 -> checkout();

                case 4 -> System.exit(0);

                default -> System.out.println("Invalid option. Please choose again.");

            }

        }

    }

    private static void initializeBooks()

    {

        books.put("B001", new Book("Book 1", 49.99));

        books.put("B002", new Book("Book 2", 39.99));

        books.put("B003", new Book("Book 3", 29.99));

        books.put("B004", new Book("Book 4", 19.99));

        books.put("B005", new Book("Book 5", 9.99));

    }

    private static void browseBooks()

    {

        System.out.println("\nAvailable Books:");

        for (String id : books.keySet())

        {

            Book book = books.get(id);

            System.out.println(id + ": " + book.title + " ($" + book.price + ")");

        }

    }

    private static void addBookToCart()

    {

        System.out.print("\nEnter book ID: ");

        String bookId = scanner.nextLine();

        if (books.containsKey(bookId))

        {

            cart.add(books.get(bookId));

            System.out.println("Book added to cart successfully!");

        }

        else

        {

            System.out.println("Invalid book ID. Please try again.");

        }

    }

    private static void checkout()

    {

        double totalCost = 0;

        if (cart.isEmpty())

        {

            System.out.println("Your cart is empty.");

            return;

        }

        System.out.println("\nCart Items:");

        for (int i = 0; i < cart.size(); i++)

        {

            Book book = cart.get(i);

            System.out.println((i + 1) + ". " + book.title + " ($" + book.price + ")");

            totalCost += book.price;

        }

        System.out.println("\nTotal Cost: $" + totalCost);

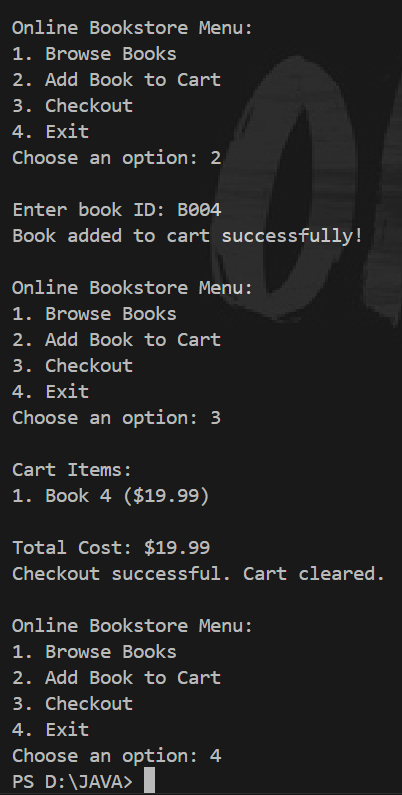
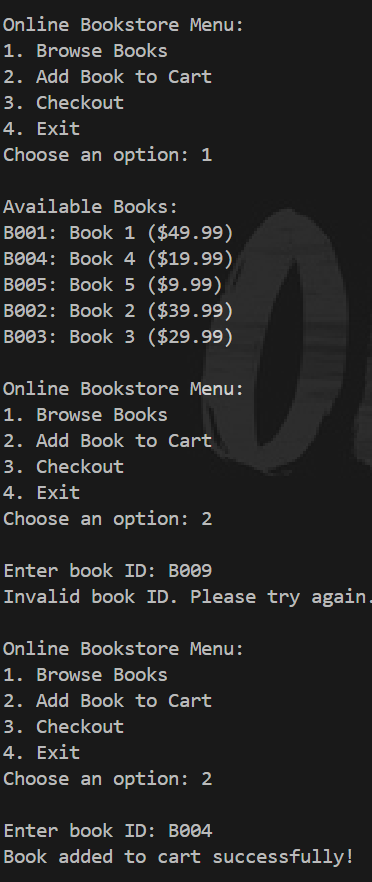
        cart.clear();

        System.out.println("Checkout successful. Cart cleared.");

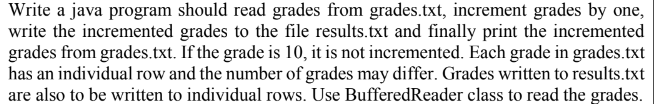
    }

}

**OUTPUT:**



# PRACTICAL :2



**CODE:**

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

public class prac5\_2

{

    public static void main(String[] args)

    {

        try (BufferedReader br = new BufferedReader(new FileReader("D:\\JAVA\\JAVA LAB\\grades.txt")))

        {

            try (BufferedWriter fw = new BufferedWriter(new FileWriter("D:\\JAVA\\JAVA LAB\\result.txt")))

            {

                String line;

                while ((line = br.readLine()) != null)

                {

                    int grade = Integer.parseInt(line.trim());

                    if (grade < 10)

                    {

                        grade++;

                    }

                    fw.write(String.valueOf(grade));

                    fw.newLine();

                }

                fw.close();

                br.close();

            }

        } catch (IOException e)

        {

            System.out.println("Error handeling file " + e.getMessage());

        }

        try (BufferedReader br = new BufferedReader(new FileReader("D:\\\\JAVA\\\\JAVA LAB\\\\result.txt")))

        {

            String line;

            while ((line = br.readLine()) != null)

            {

                System.out.println(line);

            }

            br.close();

        } catch (IOException e)

        {

            System.out.println("Error handeling file " + e.getMessage());

        }

    }

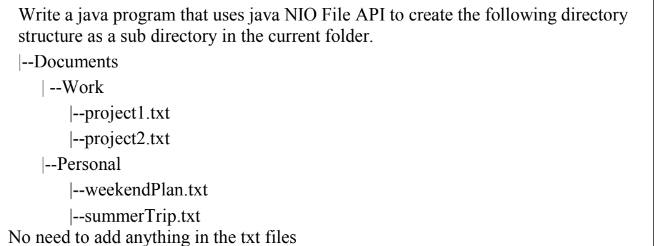
}

# OUTPUT:

**INPUT: OUTPUT:**

# 

**PRACTICAL :3**



**CODE:**

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

public class prac5\_3

{

    public static void main(String[] args) throws IOException

    {

        Path dir = Paths.get("Documents");

        Path work = dir.resolve("Work");

        Path project\_1 = work.resolve("project\_1.txt");

        Path project\_2 = work.resolve("project\_2.txt");

        Path personal = dir.resolve("Personal");

        Path weekendPlan = personal.resolve("weekendPlan.txt");

        Path summerTrip = personal.resolve("summerTrip.txt");

        Files.createDirectories(work);

        Files.createDirectories(personal);

        Files.createFile(project\_1);

        Files.createFile(project\_2);

        Files.createFile(weekendPlan);

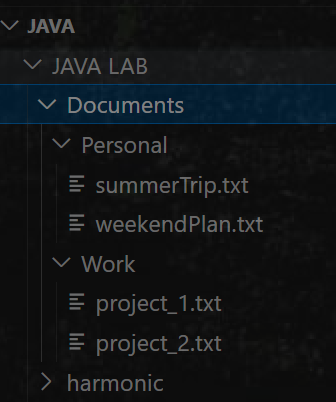
        Files.createFile(summerTrip);

        System.out.println(dir.toAbsolutePath());

    }

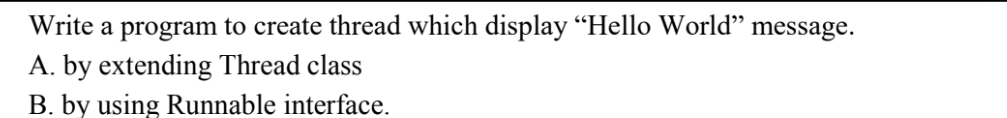
}

**OUTPUT:**



PART:6

# PRACTICAL :1



**CODE:**

**1:**

public class prac6\_1\_1 implements Runnable

{

    @Override

    public void run()

    {

        System.out.println("Hello World");

    }

    public static void main(String[] args)

    {

        prac6\_1\_1 runnable = new prac6\_1\_1();

        Thread thread = new Thread(runnable);

        thread.start();

    }

}

**2:**

public class prac6\_1\_2 implements Runnable

{

    @Override

    public void run()

    {

        System.out.println("Hello World");

    }

    public static void main(String[] args)

    {

        prac6\_1\_1 obj = new prac6\_1\_1();

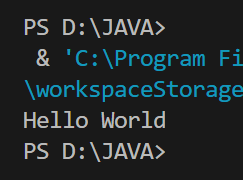
        Thread t = new Thread(obj);

        t.start();

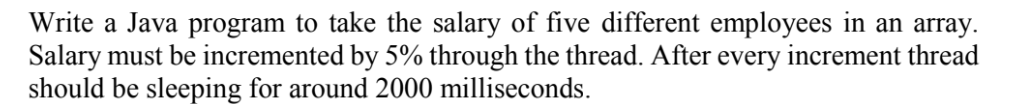
    }

}

**OUTPUT:**



# PRACTICAL :2



**CODE:**

public class prac6\_2 extends Thread

{

    int[] salary = new int[5];

    @Override

    public void run()

    {

        for (int j = 0; j < salary.length; j++)

        {

            double temp = salary[j];

            temp = temp + (temp \* 0.05);

            salary[j] = (int) temp;

            System.out.println("Incremented salary : " + salary[j]);

            try

            {

                Thread.sleep(2000);

            } catch (InterruptedException e)

            {

                System.out.println("interrupt occur ! ");

            }

        }

    }

    public static void main(String[] args)

    {

        prac6\_2 obj = new prac6\_2();

        for (int j = 0; j < 5; j++)

        {

            obj.salary[j] = 50000 + (j\*1000);

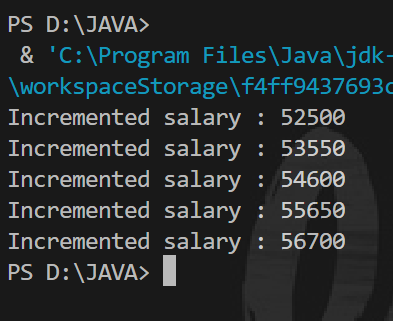
        }

        obj.start();

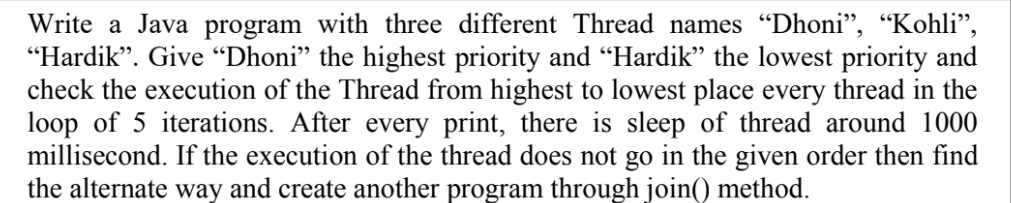
    }

}

# OUTPUT:



**PRACTICAL :3**



**CODE:**

class MyThread extends Thread

{

    @Override

    public void run()

    {

        for (int i = 0; i < 5; i++)

        {

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

            try

            {

                Thread.sleep(1000);

            } catch (InterruptedException e)

            {

                e.printStackTrace();

            }

        }

    }

}

public class prac6\_3

{

    public static void main(String[] args)

    {

        MyThread dhoni = new MyThread();

        dhoni.setName("Dhoni");

        dhoni.start();

        try

        {

            dhoni.join();

        } catch (InterruptedException e)

        {

            e.printStackTrace();

        }

        MyThread kohli = new MyThread();

        kohli.setName("Kohli");

        kohli.start();

        try

        {

            kohli.join();

        } catch (InterruptedException e)

        {

            e.printStackTrace();

        }

        MyThread hardik = new MyThread();

        hardik.setPriority(Thread.MIN\_PRIORITY);

        hardik.setName("Hardik");

        hardik.start();

        try

        {

            hardik.join();

        } catch (InterruptedException e)

        {

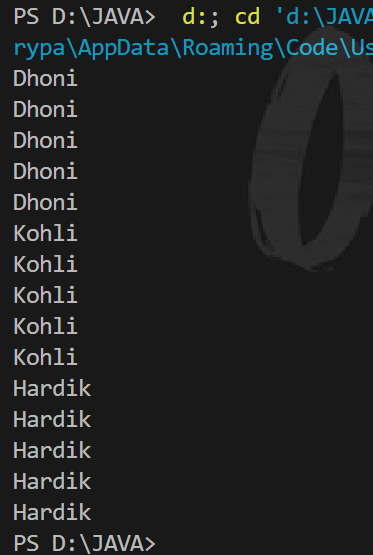
            e.printStackTrace();

        }

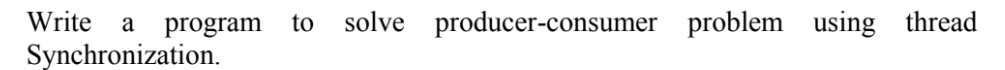
    }

}

**OUTPUT:**



# PRACTICAL: 4



**CODE:**

class Stack

{

    private final int maxSize;

    private int top;

    private final int[] stackArray;

    public Stack(int size)

    {

        maxSize = size;

        stackArray = new int[maxSize];

        top = -1;

    }

    public synchronized void push(int value)

    {

        while (top >= maxSize - 1)

        {

            try

            {

                wait();

            } catch (InterruptedException e)

            {

                System.out.println("Error: " + e.getMessage());

            }

        }

        stackArray[++top] = value;

        notify();

    }

    public synchronized int pop()

    {

        while (top < 0)

        {

            try

            {

                wait();

            } catch (InterruptedException e)

            {

                System.out.println("Error: " + e.getMessage());

            }

        }

        notify();

        return stackArray[top--];

    }

}

class Producer extends Thread

{

    private final Stack stack;

    private final int numItems;

    public Producer(Stack stack, int numItems)

    {

        this.stack = stack;

        this.numItems = numItems;

    }

    @Override

    public void run()

    {

        for (int i = 1; i <= numItems; i++)

        {

            stack.push(i);

            System.out.println("Produced: " + i);

            try

            {

                sleep(1000); // simulate production time

            } catch (InterruptedException e)

            {

                System.out.println("Error: " + e.getMessage());

            }

        }

    }

}

class Consumer extends Thread

{

    private final Stack stack;

    private final int numItems;

    public Consumer(Stack stack, int numItems)

    {

        this.stack = stack;

        this.numItems = numItems;

    }

    @Override

    public void run()

    {

        for (int i = 1; i <= numItems; i++)

        {

            int item = stack.pop();

            System.out.println("Consumed: " + item);

            try

            {

                sleep(1000); // simulate consumption time

            } catch (InterruptedException e)

            {

                System.out.println("Error: " + e.getMessage());

            }

        }

    }

}

public class prac6\_4

{

    public static void main(String[] args)

    {

        Stack stack = new Stack(5);

        Producer producer = new Producer(stack, 10);

        Consumer consumer = new Consumer(stack, 10);

        producer.start();

        consumer.start();

    }

}

# OUTPUT:

