CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY & RESEARCH**

Department of Information Technology

**PRACTICALS**

1. **AIM:** In a peaceful town, a budding programmer named Sam was tasked by her mentor, Ms. Java, to declare an integer variable named age, assign it the value of 25, and display it in a sentence. Sam quickly took to her computer and, with focus,wrote a program that would show "25 is the age of Sam." on the screen. Pleased with her work, Sam proudly presented her solution to Ms. Java, who commended her for her precise and clear coding skills.

**CODE:**

public class first {

    public static void main(String[] args) {

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

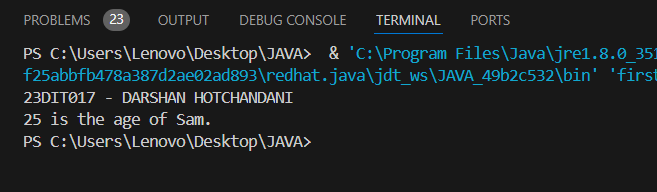
        int age=25;

        System.out.println(age+" is the age of Sam.");

    }

}

**OUTPUT:**

****

**CONCLUSION :**

So , from this practical I learnt the basic knowledge about the JAVA programming. I learnt about how to print something in java using System.out.println() . Also learnt how to declare the class then inside it the main function and then code inside it.

1. **AIM :** Java program that converts a string entered by the user to Morsecode or vice versa. It will require the implementation of data structures, including arrays, loops, and conditional statements.

• Create two arrays - one to contain the strings of letters to be converted and

one to contain the Morse codes.

• In the program's main method, prompt the user for input to choose

between the string or Morse.

• For Morse code conversion, read a string from the user; use conditional

statements, looping, and array methods to convert the string to Morse-code.

• For string conversion, read in a Morse-coded string from the user; use

arrays, conditional statements, and looping to convert

Morse code to a string

**CODE:**

import java.util.\*;

class second{

    public static void main(String[] args){

        System.out.println("23DIT017-DARSHAN HOTCHANDANI");

        char[] a={'A','B','C','D'};

        String[] b={"...-","..-.",".-..","-..."};

        int n;

        do {

            System.out.println("CHOOSE \n FROM 1. ENG TO MORSE \t 2. MORSE TO ENGLISH");

            Scanner obj = new Scanner(System.in);

            int x=obj.nextInt();

            if(x==1){

            System.out.println("CHAR TO MORSE::::: \n");

            System.out.println("ENTER YOUR CHARACTER: ");

            Scanner obj1 = new Scanner(System.in);

            String line=obj1.nextLine();

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

                char s = line.charAt(i);

                if(s==a[i]){

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

                }

            }

            }

            else {

            System.out.println("MORSE TO CHAR::::: \n");

            System.out.println("ENTER YOUR MORSE CODE (use spaces to separate each morse character): ");

            Scanner obj2 = new Scanner(System.in);

            String line = obj2.nextLine();

            String[] s2 = line.split(" ");

            String cod = "";

            for (int i = 0; i < s2.length; i++) {

            int index = 0;

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

                if (b[j].equals(s2[i])) {

                    index = j;

                }

            }

            cod = cod + a[index];

            }

            System.out.println("String : " + cod);

            }

            System.out.println("\n Press 1 to continue further and 2 to exit: ");

            Scanner obj5= new Scanner(System.in);

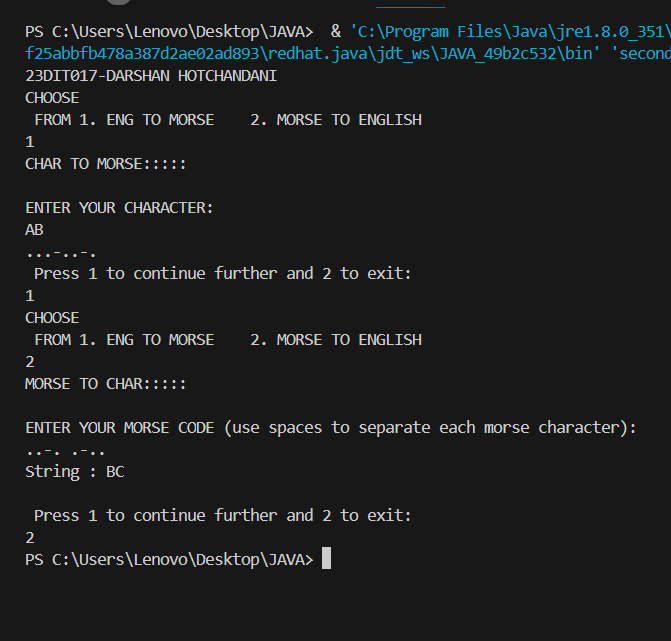
            n=obj5.nextInt();

            } while (n!=2);

    }

}

**OUTPUT :**

****

**CONCLUSION :**

So, from this practical I learnt how to take input as an string in java, how to acceess the particular element from the string using charat . The I came to know about morse code more, also how to compare something and print accordingly. Then main thing I learnt is to take input from user using the scanner function . By importing java.util.\* all the files that are there are included. Also learnt how to append a string by using + .

1. **AIM** : A typical mobile number in India is “+91-AA-BBB-CCCCC”. Where the first two digits (AA)indicate a mobile system operator, the next three (BBB) denote the mobile switching code(MSC) while the remaining five digits (CCCCC) are unique to the subscriber. Write an application that takes a mobile number as an input from a user in above mentioned format and display code for mobile system operator, mobile switching code and last 5 digits which are unique to subscriber.

Ex. For an input +91-94-999-65789, output should be :Mobilesystem operator code is 94 MSC is 999 Unique code is 65789

**CODE:**

import java.util.Scanner;

class Third {

    public static void main(String[] args) {

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        Scanner obj = new Scanner(System.in);

        String mob = obj.nextLine();

        // Check if the input length is exactly 10

        if (mob.length() == 10) {

            String mobileOperatorCode = mob.substring(0, 2);

            String mobileSwitchingCode = mob.substring(2, 5);

            String uniqueSubscriberCode = mob.substring(5);

            System.out.println("Mobile system operator code: " + mobileOperatorCode);

            System.out.println("Mobile switching code (MSC): " + mobileSwitchingCode);

            System.out.println("Unique subscriber code: " + uniqueSubscriberCode);

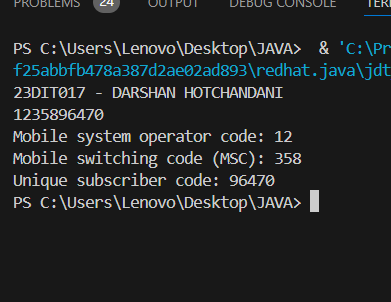
        } else {

            System.out.println("Invalid input length. Please enter a 10-digit mobile number.");

        }

    }

}

****

**OUTPUT :**

**CONCLUSION:**

So, from this practical main thing I learnt was about usage of the function named substr , which is used for accessing the substring from a string . .nextline() is used for taking the input from the user in form of string. So substr is very useful function for accessing from a string from any character to other character.

1. **AIM :** An electric appliance shop assigns code 1 to motor,2 to fan,3 to tube and 4 for wires. All other items have code 5 or more. While selling the goods, a sales tax of 8% to motor,12% to fan,5% to tube light,7.5% to wires and 3% for all other items is charged. A list containing the product code and price in two different arrays. Write a java program using switch statement to prepare the bill.

**CODE :**

import java.util.Scanner;

class Fourth {

    public static void main(String[] args) {

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        String[] product = {"Motor", "Fan", "Tube", "Wire", "Other"};

        double[] rate = {1000, 500, 200, 100, 300};

        double[] tax = {8, 12, 5, 7.5, 3};

        System.out.println("GREETINGS!!! WELCOME TO SNAP APPLIANCES: ");

        int a6;

        double fp = 0;

        StringBuilder billDetails = new StringBuilder();

        Scanner obj = new Scanner(System.in);

        do {

            System.out.println("\nCHOOSE 1: Motor \t 2: Fan \t 3: Tube \t 4: Wire \t 5: Other");

            int x = obj.nextInt();

            int units;

            double price = 0;

            if (x >= 1 && x <= 5) {

                System.out.println("TELL THE NO. OF UNITS: ");

                units = obj.nextInt();

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

                    double y = (tax[x - 1] \* rate[x - 1]) / 100 + rate[x - 1];

                    price += y;

                }

                fp += price;

                billDetails.append("Product: ").append(product[x - 1])

                            .append(", Units: ").append(units)

                            .append(", Price: ").append(price).append("\n");

            } else {

                System.out.println("Invalid choice. Please select a valid product.");

            }

            System.out.println("DO YOU WANT TO CONTINUE? 1 FOR YES \t 2 FOR NO");

            a6 = obj.nextInt();

        } while (a6 == 1);

      System.out.println("\nYOUR FINAL BILL IS:");

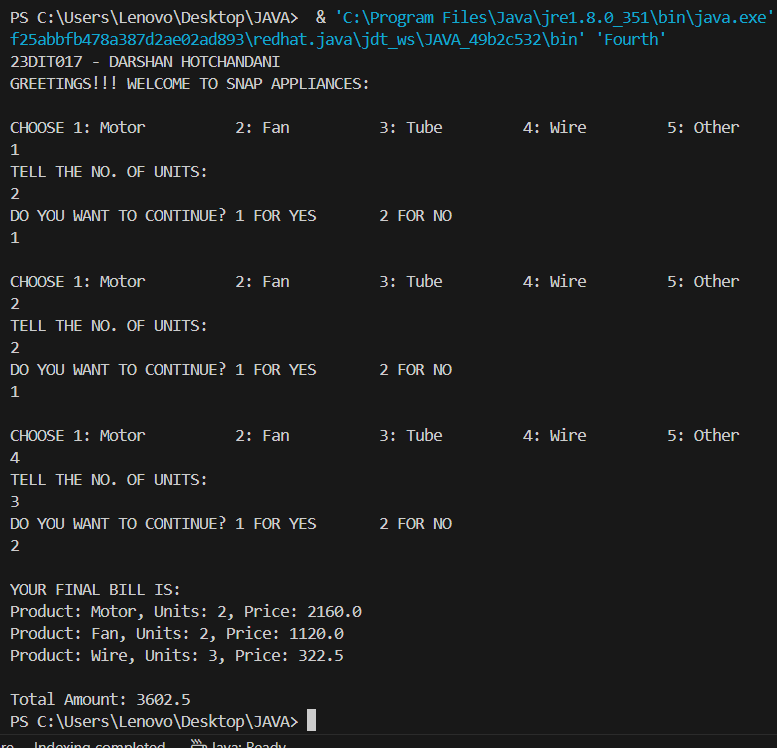
        System.out.println(billDetails.toString());

        System.out.println("Total Amount: " + fp);

    }

}

**OUTPUT :**

****

**CONCLUSION :**

So, from this practical I learnt how to use do while loop in java. Understood the use of conditional logic to handle various cases and apply specific business rules effectively . This task enhanced my skills in writing conditional logic, managing arrays, and performing arithmetic operations in Java. We can also use switch statements in this to access different appliances.

1. **AIM :** Create a Java program that simulates a guessing game, where the computer picks a random number between 1 and 100 and the user has to guess it. We can use the Scanner class to 1 get user input and a loop to allow multiple guesses.

**CODE:**

import java.util.\*;

class fifth{

    public static void main(String[] args){

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        int n,x;

        Random r= new Random();

        x=r.nextInt(100);

        System.out.println("WELCOME TO THE GAME!!");

        do {

            System.out.println("Guess the number: ");

            Scanner obj = new Scanner(System.in);

            n=obj.nextInt(  );

            if(n==x){

                System.out.println("Your guessed right number!!!!");

                break;

            }

            else if (n>x) {

                System.out.println("Number is smaller than this!!!");

            }

            else{

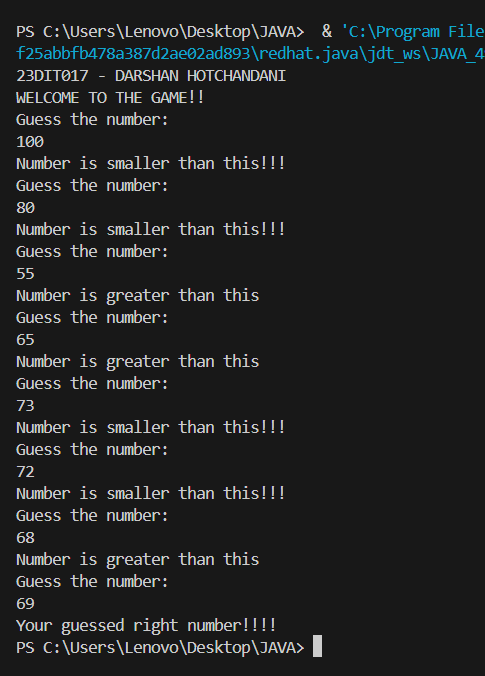
                System.out.println("Number is greater than this");

            }

        } while (true);

    }

}

**OUTPUT :**

**CONCLUSION:**

So, from this practical I learnt about the random() function , which generates any random integer from the bound given. Enjoyed making this short and nice game, of guessing the number , that is generated by computer. Use of if , else if , else becomes very important in this type of any games, or any type of games. Also do while loop is important to iterate till the number is not found. Break!! Is used when we want to stop a loop when we got the answer.

1. **AIM :** Imagine you're tasked with creating a function that takes a string and a number. The goal is to repeat the first few characters of the string a specified number of times. If the string is shorter than the specified length, you should

repeat whatever characters are available. How would you approach this problem?(function)

**CODE:**

import java.util.Scanner;

public class sixth {

    public static void main(String[] args){

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        System.out.println("Enter the string: ");

        Scanner obj = new Scanner(System.in);

        String n=obj.nextLine();

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

        Scanner obj2 = new Scanner(System.in);

        int x=obj2.nextInt(  );

        fun(x, n);

    }

    public static void fun(int x,String y){

        int a;

        String b;

        a=x;

        b=y;

        String c="",d="";

        if(y.length()<a){

            for(int i=0;i<a;i++){

                d=d+y;

            }

        }

        else{

        for(int i=0;i<a;i++){

            c=c+y.charAt(i);

        }

        for(int i=0;i<a;i++){

            d=d+c;

        }

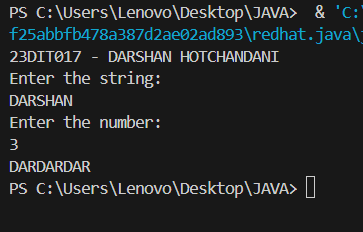
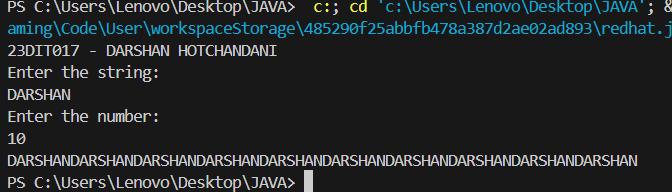
    }

        System.out.println(d);

    }

}

**OUTPUT :**

****

**CONCLUSION :**

So , I learned how to create a function that manipulates strings and handles edge cases effectively. By developing a function that repeats the first few characters of a string a specified number of times, I enhanced my understanding of string operations and loops. I also learned how to handle scenarios where the string length is shorter than the specified length, ensuring the function remained working and versatile. Learnt how to call a function in java.

**7 ) AIM :** Imagine you're working with an array of integers, and your task is to count how many times the number 9 appears in the array. How would you write a Java program that efficiently determines this count, regardless of the array's size or the position of the numbers?

**CODE:**

import java.util.Scanner;

public class seventh {

    public static void main(String[] args){

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

            int count=0;

            System.out.println("Enter array size: ");

            Scanner obj = new Scanner(System.in);

            int x= obj.nextInt();

            Scanner obj2= new Scanner(System.in);

            int arr[]=new int[x];

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

            for(int i=0;i<x;i++){

                arr[i]=obj2.nextInt();

            }

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

                if(arr[i]==9){

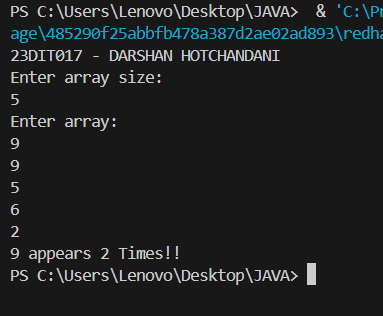
                    count++;

                }

            }

            System.out.println("9 appears "+count+" Times!!");

    }

}****

**OUTPUT :**

**CONCLUSION:**

So, from this practical I learned about the array manipulation. How can we input the array and use it’s elements accordingly. Also learnt about iteration and use of conditions according to the need from the array.

**8 ) AIM :** Suppose you are developing a text transformation tool. Your task is to create a function that takes a string and transforms it such that every character in the original string is doubled. For example, "The" becomes "TThhee". How would you design and

implement this function in Java to handle any input string effectively?

**CODE:**

import java.util.\*;

public class eight {

    public static void main(String[] args){

System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        System.out.println("Enter the string: ");

        Scanner obj = new Scanner(System.in);

        String x= obj.nextLine();

        convert(x);

    }

    public static void convert(String x){

        String a,b="";

        a=x;

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

            b=b+a.charAt(i)+a.charAt(i);

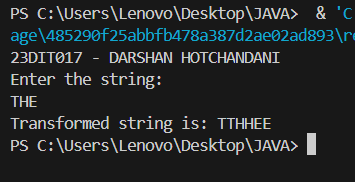
        }

        System.out.println("Transformed string is: "+b);

    }

}

**OUTPUT:**

****

**CONCLUSION :**

So , I learned how to manipulate strings by doubling each character . By designing a function that iterates through the input string and appends each character twice to a new string, I enhanced my understanding of string operations and efficient string building techniques. Through this exercise, I improved my skills in string manipulation, iteration, and use of string buffers or builders in Java.

**9) AIM :** you're a cybersecurity analyst investigating a suspicious string of characters. You need to analyze it thoroughly to uncover any hidden patterns or anomalies. The number of characters in the string to understand its size, Standardize the string for case-insensitive comparisons, Highlight potential keywords or acronyms, and Identify palindromes or potential encryption methods. Sort the string: Analyze character distribution and frequency.

**CODE:**

import java.util.Arrays;

import java.util.Scanner;

public class ninth {

    public static void main(String[] args) {

            System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

            Scanner sc = new Scanner(System.in);

            String s;

            System.out.println("Enter string");

            s = sc.nextLine();

            int size=s.length();

            //lowercase with built in method

            System.out.println("Lower case with built in: ");

            System.out.println(s.toLowerCase());

            //lowercase without built in method

            // Lower case without built-in method

            System.out.println("Lower case without built-in method: ");

            String lowerCase = "";

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

            char ch = s.charAt(i);

            if (ch >= 'A' && ch <= 'Z') {

                ch = (char) (ch + 32);

            }

            lowerCase += ch;

            }

            System.out.println(lowerCase);

            //uppercase with built in method

            System.out.println("upper case with built in method: ");

            System.out.println(s.toUpperCase());

            System.out.println("Upper case without built-in method: ");

            String upperCase = "";

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

            char ch = s.charAt(i);

            if (ch >= 'a' && ch <= 'z') {

                ch = (char) (ch - 32);

            }

            upperCase += ch;

            }

            System.out.println(upperCase);

            //length with built in method

            System.out.println("length with built in method: ");

            System.out.println(s.length());

            //length without built in method

            System.out.println("Length without built-in method");

            int a = 0;

            try {

            while (s.charAt(a) != '\0') {

                a++;

            }

            } catch (StringIndexOutOfBoundsException e) {

            }

            System.out.println(a);

            //reverse with built in method

            String reversedStr = new StringBuilder(s).reverse().toString();

            System.out.println("Reversed string with built-in method: " + reversedStr);

            //reverse without built in method

            String str="";

            for(int i=size-1;i>=0;i--)

            {

            str=str+s.charAt(i);

            }

            System.out.println("Reversed string without built in method :"+str);

          //converting to character array

           char ch[]=new char[size];

           for(int i=0;i<size;i++)

           {

            ch[i]=s.charAt(i);

           }

          // with out using inbuilt sorting function.....

           for(int i=0;i<size;i++)

           {

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

               {

                   if(ch[i]>ch[j])

                   {

                       char temp=ch[i];

                       ch[i]=ch[j];

                       ch[j]=temp;

                   }

               }

           }

           System.out.println("Sorted string without built in method: "+ch);

           //using inbuilt sorting function

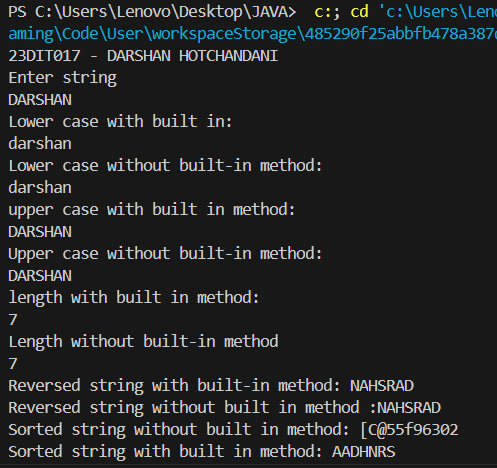
           Arrays.sort(ch);

           String sort=new String(ch);

           System.out.println("Sorted string with built in method: "+sort);

        }

    }

****

**OUTPUT :**

**CONCLUSION :**

So , from this practical I learnt to use various functions like length() , touppercase() , tolowercase() , sort() etc . Also learnt to implement all this things without the use of this all functions. Main thing in this was the uses of for loop for iterating through the string.

**10) AIM :** You're tasked with creating a basic encryption algorithmfor your college project. The first step involves manipulating a given string, "CHARUSATUNIVERSITY".Calculate the number of characters in the string tounderstand its structure, Identify the target character: Thecharacter to be replaced is 'H'. Replace the targetcharacter: Substitute 'H' with the first letter of yourname. For instance, if your name starts with 'A', replace'H' with 'A'. and Transform all characters to lowercasefor consistency, and display the modified string.

**CODE:**

import java.util.Scanner;

public class tenth {

    public static void main(String[] args) {

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        String s = "CHARUSAT UNIVERSITY";

        System.out.println(s);

        System.out.println("Length of this string is: "+s.length());

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

        Scanner sc = new Scanner(System.in);

        String s1 = sc.nextLine();

        //METHOD 1

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

            if(s.charAt(i)== 'H'){

                s=s.replace(s.charAt(i), s1.charAt(0));

            }

        }

        System.out.println("After replacement by method 1: "+s);

        //METHOD 2

        char ch[] = s.toCharArray();

        char ch1[]=s1.toCharArray();

        ch[1] = ch1[0];

        String j = new String(ch);

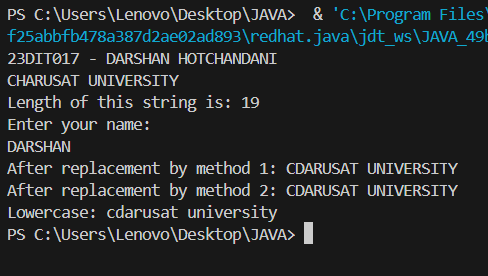
        System.out.println("After replacement by method 2: "+j);

        System.out.println("Lowercase: "+j.toLowerCase());

    }

}

**OUTPUT :**

****

**CONCLUSION :**

I learned how to perform basic string manipulations to create a simple encryption algorithm. By calculating the number of characters in a given string, I gained a better understanding of string properties. I then practiced replacing a specific target character with another character, enhancing my skills in string replacement operations. I improved my ability to write clear and effective Java code for manipulating and transforming strings.

**11) AIM :** You're a budding Java programmer working on a currency conversion application. Your initial task is to convert Pounds to Rupees. To practice different input methods, you decide to implement two approaches: command-line arguments and user input using the Scanner class.

**CODE:**

import java.util.\*;

public class twelfth{

    public static void main(String[] args){

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        long n;

        System.out.println("Enter in Pounds to convert in Rupees: ");

        Scanner a= new Scanner(System.in);

        n=a.nextLong();

        System.out.println("USING USER INPUT : ");

        convert(n);

        System.out.println("USING COMMAND LINE ARGUEMENT : ");

        convert(50);

    }

    static void convert(long n){

        long a;

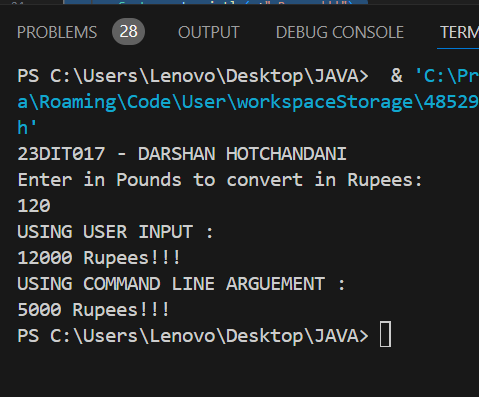
        a=n;

        long r = n \* 100;

        System.out.println(r+" Rupees!!!");

    }

}

**OUTPUT :**

**CONCLUSION :**

So, from this I learnt how to implement currency conversion from Pounds to Rupees. By practicing with both command-line arguments and user input via the Scanner class, I gained experience in handling different input methods. This task enhanced my understanding of basic arithmetic operations, user interaction, etc. Also came to know how the non returning value function works, like in this I’ve made the convert function.

**12) AIM :** Create a class called Employee that includes three piecesof information as instance variables—a first name (typeString), a last name (type String), and a monthly salary(double). Your class should have a constructor thatinitializes the three instance variables. Provide a set and aget method for each instance variable. If the monthly

salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display

each object’s yearly salary. Then give each Employee a 10% raise and display each Employee’s yearly salary again.

**CODE:**

import java.util.Scanner;

public class thirteen {

    public static void main(String[] args) {

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        employee e1 = new employee();

        employee e2 = new employee();

        System.out.println("For first employee: ");

        e1.get();

        System.out.println("For second employee: ");

        e2.get();

        System.out.println("First employee: ");

        e1.set();

        e1.raise();

        System.out.println("Second employee: ");

        e2.set();

        e2.raise();

    }

}

class employee{

    String fname;

    String lname;

    double sal;

        employee(){

            fname="";

            lname="";

            sal=0;

        }

    void get(){

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

        Scanner obj= new Scanner(System.in);

        fname= obj.nextLine();

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

        Scanner obj1= new Scanner(System.in);

        lname= obj1.nextLine();

        System.out.println("Enter yearly salary: ");

        Scanner obj2= new Scanner(System.in);

        sal= obj2.nextDouble();

        if(sal<0){

            sal=0;

        }

    }

    void set(){

        System.out.println("First name: "+fname);

        System.out.println("Last name: "+lname);

        System.out.println("Sal: "+sal);

    }

    void raise(){

        double a;

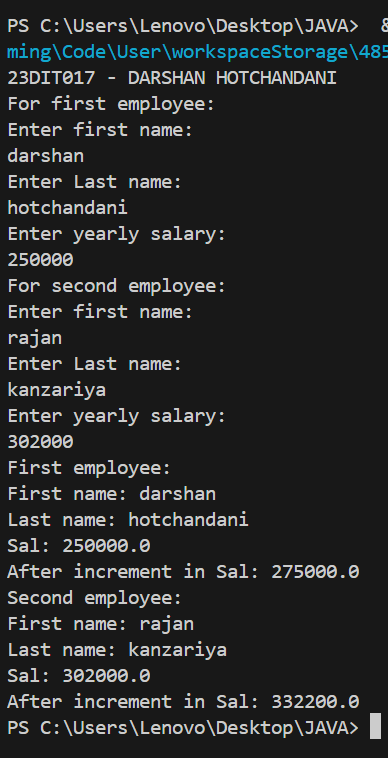
        a=0.1\*sal+sal;

        sal=a;

        System.out.println("After increment in Sal: "+sal);

    }

}

****

**OUTPUT :**

**CONCLUSION :**

So, from this practical I learnt how to create another class in java and call it in the main function for it’s use. Also learnt to implement the member function and how to use the member function . Created the object to use / access the data inside the class. Default constructor is called whenever the object of the particular class is created. We can create as many object as we want and store the data in it.

**13) AIM :** Create a class called Date that includes three pieces ofinformation as instance variables—a month (type int), aday (type int) and a year (type int). Your class should havea constructor that initializes the three instance variablesand assumes that the values provided are correct. Providea set and a get method for each instance variable. Providea method displayDate that displays the month, day andyear separated by forward slashes (/). Write a testapplication named DateTest that demonstrates classDate’s capabilities.

**CODE:**

import java.util.\*;

public class datetest{

    public static void main(String[] args) {

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        date d1 = new date();

        d1.setY();

        d1.setm();

        d1.setd();

        d1.gety();

        d1.getm();

        d1.getd();

        d1.display();

    }

}

class date{

    int month,day,year;

    Scanner obj= new Scanner(System.in);

    date(){

        month=0;

        day=0;

        year=0;

    }

    void setY(){

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

        year= obj.nextInt();

    }

    void setm(){

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

        month= obj.nextInt();

        if(month>12){

            System.out.println("INVALID!!!");

            System.out.println("ENTER VALID Month: ");

            month= obj.nextInt();

        }

    }

    void setd(){

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

        day= obj.nextInt();

        if(day>31){

            System.out.println("INVALID!!!");

            System.out.println("ENTER VALID DAY: ");

            month= obj.nextInt();

        }

    }

    void gety(){

        System.out.println("Entered Year is: " + year);

    }

    void getm(){

        System.out.println("Entered Month is: " + month);

    }

    void getd(){

        System.out.println("Entered Day is: " + day);

    }

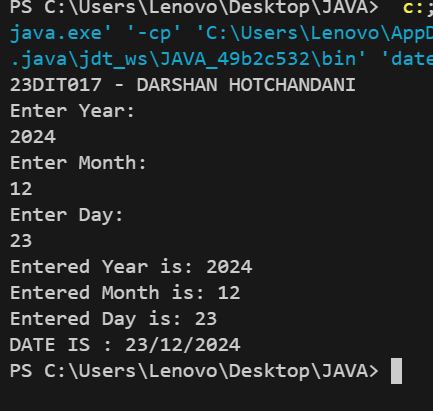
    void display(){

        System.out.println("DATE IS : " + day + "/" + month + "/" + year);

    }

}

**OUTPUT :**

****

**CONCLUSION :**

I developed a class called Date in Java, which encapsulates the month, day, and year as instance variables. I learned to create a constructor to initialize these variables and ensured the values provided were assumed to be correct. Additionally, I created a displayDate method to format and display the date in a user-friendly manner. Through the DateTest application, I demonstrated the capabilities of the Date class. This helped to learnt about class, member function and data member.

**14) AIM :** Write a program to print the area of a rectangle by creatin a class named 'Area' taking the values of its length andbreadth as parameters of its constructor and having amethod named 'returnArea' which returns the area of therectangle. Length and breadth of rectangle are enteredthrough keyboard.

**CODE:**

import java.util.\*;

public class fourteen {

    public static void main(String[] args) {

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        int l ,b;

        Scanner obj = new Scanner(System.in);

        System.out.println("Enter the length: ");

        l = obj.nextInt();

        System.out.println("Enter the Breadth: ");

        b = obj.nextInt();

        Area a1 = new Area(l, b);

        a1.returnarea();

    }

}

class Area{

    int l , b;

    Area(int l1, int b1){

        l=l1;

        b=b1;

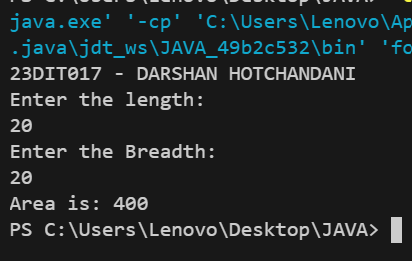
    }

    void returnarea(){

        System.out.println("Area is: " + (l\*b));

    }

}

**OUTPUT :**

**CONCLSUION :**

I created a Java program to calculate the area of a rectangle using a class named Area. By defining a constructor to take length and breadth as parameters and a method returnArea to calculate and return the area, I reinforced my understanding of object-oriented principles such as encapsulation and method implementation.

**15) AIM :** Imagine you're building a scientific calculator application. One crucial feature is handling complex numbers. You decide to create a Complex class to represent complex numbers and perform operations on them.(sum, difference and product)

**CODE:**

import java.util.\*

;

public class fifteen {

    public static void main(String[] args) {

System.out.println("23DIT017- DARSHAN HOTCHANDANI");

        int real ,imag , real1, imag1;

        Scanner obj = new Scanner(System.in);

        System.out.println("Enter the real: ");

        real = obj.nextInt();

        System.out.println("Enter the imag: ");

        imag = obj.nextInt();

        System.out.println("Enter the real2: ");

        real1 = obj.nextInt();

        System.out.println("Enter the imag2: ");

        imag1 = obj.nextInt();

        complex c1= new complex(real,imag);

        complex c2= new complex(real1,imag1);

        complex c3= new complex();

        complex c4= new complex();

        c4=c3.sum(c1, c2);

        System.out.println("SUM IS: ");

        c4.print();

        c4=c3.product(c1, c2);

        System.out.println("PRODUCT IS: ");

        c4.print();

        c4=c3.diff(c1, c2);

        System.out.println("DIFFERENCE IS: ");

        c4.print();

    }

}

class complex{

    int r1 , i1;

    complex(){

        r1=0;

        i1=0;

    }

    complex(int r, int i){

        r1=r;

        i1=i;

    }

    complex sum(complex c , complex c1){

        complex temp = new complex();

        temp.r1= c.r1+c1.r1;

        temp.i1= c.i1+c1.i1;

        return temp;

    }

    void print(){

        System.out.println(r1 +  "+" + i1 + "i");

    }

    complex product(complex c , complex c1){

        complex temp = new complex();

        temp.r1= (c.r1\*c1.r1)-(c.i1\*c1.i1);

        temp.i1= (c.r1\*c1.i1)+(c.i1\*c1.r1);

        return temp;

    }

    complex diff(complex c , complex c1){

        complex temp = new complex();

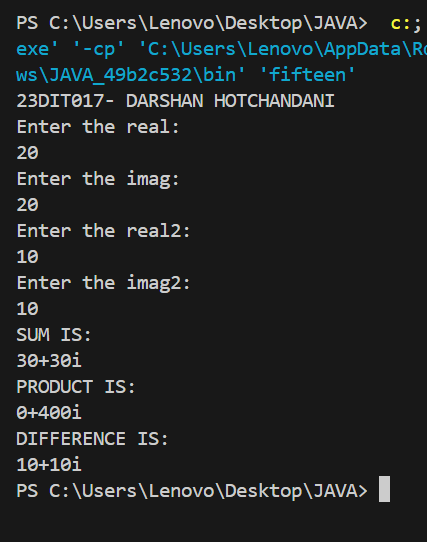
        temp.r1= c.r1-c1.r1;

        temp.i1= c.i1-c1.i1;

        return temp;

    }

}

**OUTPUT:**

**CONCLUSION:**

I developed a Complex class in Java to handle complex numbers and perform arithmetic operations such as sum, difference, and product. By defining methods to add, subtract, and multiply complex numbers, I gained a deeper understanding of how to manipulate objects and implement mathematical operations. Implementing and testing these operations demonstrated the power and flexibility of classes in handling complex data types.

**16) AIM :** Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call 1 - method of parent class by object of parent class 2 - method of child class by object of child class 3 - method of parent class by object of child class.

**CODE:**

public class sixteen {

    public static void main(String[] args) {

        System.out.println("23DIT017 - DARSHAN HOTCHANDANI");

        parent p1= new parent();

        child c1= new child();

        p1.call();

        c1.call1();

        c1.call();

    }

}

class parent{

    void call(){

        System.out.println("This is a parent class.");

    }

}

class child extends parent{

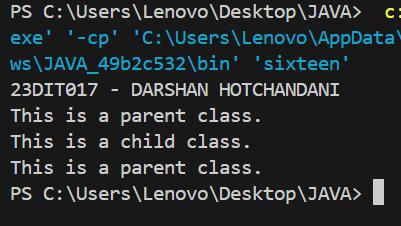
    void call1(){

        System.out.println("This is a child class.");

    }

}

**OUTPUT :**

****

**CONCLUSION :**

I learned how to implement inheritance in Java by creating a parent class and a subclass, each with their own methods. Extends keyword is used to inherit the child class from the parent class. By creating objects of both the parent and child classes, I practiced calling methods specific to each class. Furthermore, I demonstrated that an object of the child class can access the methods of the parent class, highlighting the concept of inheritance and method overriding.