**A picture containing text, clipart

Description automatically generated**

**TERM WORK**

**ON**

**JAVA PROGRAMING**

**LANGUAGE**

**(PCS 308)**

**2021-22**

**SUBMITTED TO: SUBMITTED BY:**

**DR. PRATEEK SRIVASTAVA DEVANSH RAUTELA**

**ASSOCIATE PROFESSOR UNIVERSITY ROLL NO. : 2018314**

**GEHU, D.DUN SECTION : C**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GRAPHIC ERA HILL UNIVERSITY, DEHRADUN**

**A picture containing text, clipart

Description automatically generated**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Program No.** | **Program Name** | **Page No.** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |
| **4** |  |  |
| **5** |  |  |
| **6** |  |  |
| **7** |  |  |
| **8** |  |  |
| **9** |  |  |
| **10** |  |  |
| **11** |  |  |
| **12** |  |  |

**A picture containing text, clipart

Description automatically generated**

**DEPARTMENT OF CSE**

|  |
| --- |
| **Photograph**  **Passport Size** |

**STUDENT LAB REPORT SHEET**

**Name of Student .................................................... Mob. No ......................................**

**Address Permanent .....................................................................................................**

**Father’s Name ........................... Occupation ...................... Mob. No .........................**

**Mother’s Name ........................... Occupation ...................... Mob. No .......................**

**Section ............ Branch ............ Semester ............ Class Roll No ............ Grade A B C**

**Local Address ................................... Email ............................................ Marks 5 3 1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S.N**  **o.** | **Practical** | **D.O.P.** | **Date of Submiss**  **ion** | **Grade (Viva)** | **Grade**  **(Report**  **File)** | **Total**  **Marks**  **(out of**  **10)** | **Student’s Signature** | **Teacher’s Signatur**  **e** |
| **1** |  |  |  |  |  |  |  |  |
| **2** |  |  |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |  |  |
| **5** |  |  |  |  |  |  |  |  |
| **6** |  |  |  |  |  |  |  |  |
| **7** |  |  |  |  |  |  |  |  |
| **8** |  |  |  |  |  |  |  |  |
| **9** |  |  |  |  |  |  |  |  |
| **10** |  |  |  |  |  |  |  |  |
| **11** |  |  |  |  |  |  |  |  |
| **12** |  |  |  |  |  |  |  |  |

**PRACTICAL 1**

**Create a class “Student” having following instance variables and methods.**

**Instance variables : ID, Name, Branch and University.**

**Method: setDetails() and showDetails().**

**The setDetails() method sets the values of ID, Name, Branch and University.**

**And showDetails() method shows the values of each field.**

**CODE :**

**class Student{**

**int ID;**

**String name;**

**String Branch;**

**String University;**

**void setDetails(int r, String n, String br, String un)**

**{**

**name = n;**

**ID = r;**

**Branch = br;**

**University = un;**

**}**

**void showDetails() {**

**System.out.println("Name : "+name);**

**System.out.println("ID : "+ID);**

**System.out.println("Branch : "+Branch);**

**System.out.println("University : "+University);**

**}**

**public static void main(String[] args)**

**{**

**Student Details = new Student();**

**int r=Integer.parseInt(args[0]);**

**String n=args[1];**

**String br=args[2];**

**String un=args[3];**

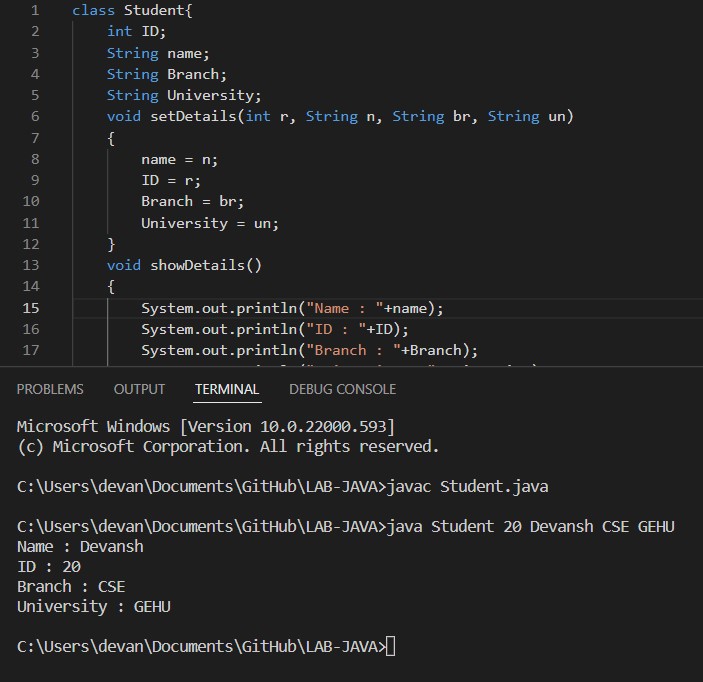
**Details.setDetails(r,n,br,un);**

**Details.showDetails();**

**}**

**}**

**OUTPUT :**

****

**PRACTICAL 2**

**Write a JAVA Program to demonstrate the working of a banking – system.**

**Instance variables : name, account\_no, amount.**

**Instance methods : deposit(), withdraw(), checkBalance(), insert(), and display().**

**Here we can deposit and withdraw amount from our account using deposit() and withdraw() methods respectively.**

**The insert() methods to initialize state and display() method is to display state values.**

**CODE :**

**public class Bank {**

**int account\_no;**

**String name;**

**int amount;**

**void insert(int account\_no, String name, int amount)**

**{**

**this.account\_no = account\_no;**

**this.name = name;**

**this.amount = amount;**

**}**

**void deposit(int amount)**

**{**

**this.amount += amount;**

**}**

**void withdraw(int amount)**

**{**

**if(this.amount >= amount)**

**{**

**System.out.println("Money withdrawn : "+amount);**

**this.amount -= amount;**

**System.out.println("Amount remaining :"+this.amount);**

**}**

**else{**

**System.out.println("Low Balance");**

**}**

**}**

**void display()**

**{**

**System.out.println("Name : "+this.name);**

**System.out.println("Account Number : "+this.account\_no);**

**System.out.println("Account Balance : "+this.amount);**

**}**

**public static void main(String args[])**

**{**

**Bank obj = new Bank();**

**obj.insert(20000,"Devansh Rautela",100000);**

**obj.display();**

**obj.withdraw(200000);**

**obj.display();**

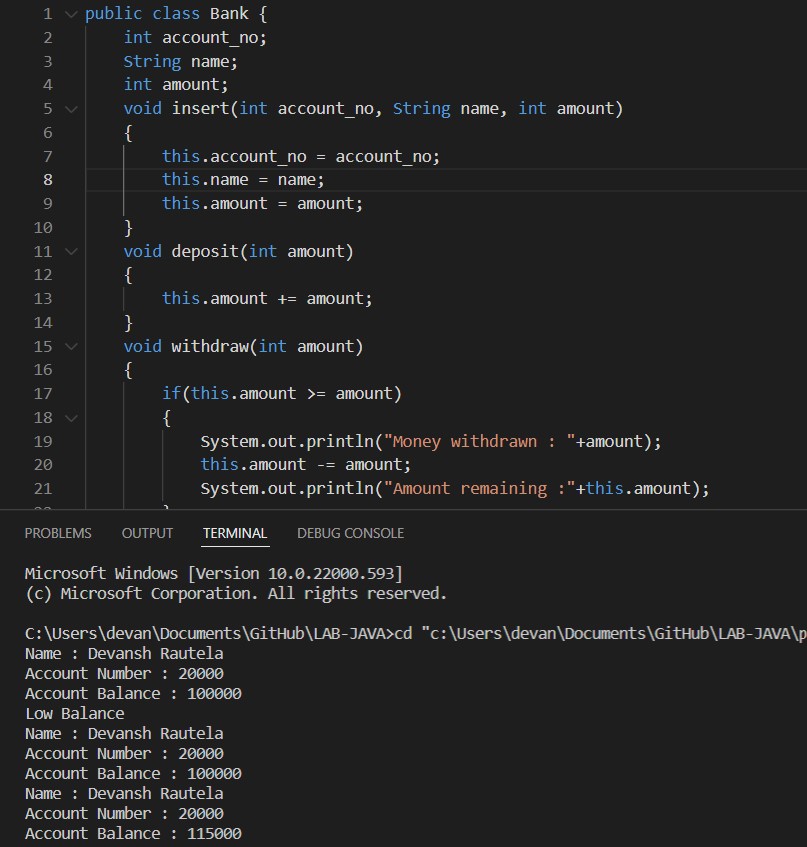
**obj.deposit(15000);**

**obj.display();**

**}**

**}**

**OUTPUT :**

****

**PRACTICAL 3**

**Write a Program to sum two numbers. Here inputs are provided through command line arguments.**

**CODE :**

**public class sum**

**{**

**public static void main(String[] args)**

**{**

**int n1 = Integer.parseInt(args[0]);**

**int n2 = Integer.parseInt(args[1]);**

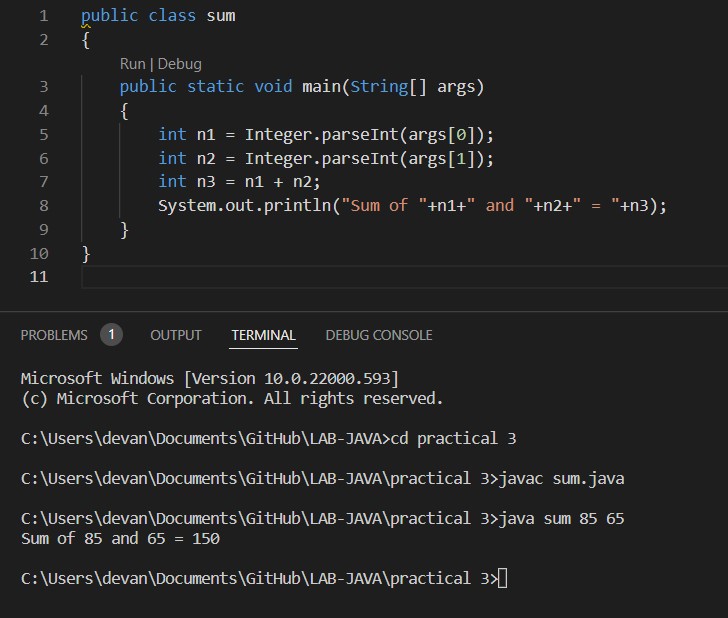
**int n3 = n1 + n2;**

**System.out.println("Sum of "+n1+" and "+n2+" = "+n3);**

**}**

**}**

**OUTPUT :**

****

**PRACTICAL 4**

**Create class Employee with following attributes and methods ID, name, department and salary.**

**This setDetails() method sets the values of ID, name, department, and salary.**

**And showDetails() method shows the value of each field.**

**Note :**

1. **Values must be entered through Scanner class.**
2. **Use proper constructor.**
3. **Use “this” reference variables to avoid ambiguity.**

**CODE :**

**public class Employee {**

**int ID;**

**String name;**

**String department;**

**int salary;**

**Employee(int ID, String name, String department, int salary)**

**{**

**this.ID = ID;**

**this.name = name;**

**this.department = department;**

**this.salary = salary;**

**}**

**void setDetails(int r, String na, String br, int un)**

**{**

**name = na;**

**ID = i;**

**department = dp;**

**salary = s;**

**}**

**void display()**

**{**

**System.out.println("Name : "+name);**

**System.out.println("Roll no. : "+ID);**

**System.out.println("Department : "+department);**

**System.out.println("Salary : "+salary);**

**}**

**public static void main(String args[])**

**{**

**Employee xyz = new Employee(0, "", "", 0);**

**int i = Integer.parseInt(args[0]);**

**String na = args[1];**

**String dp = args[2];**

**int s = Integer.parseInt(args[3]);**

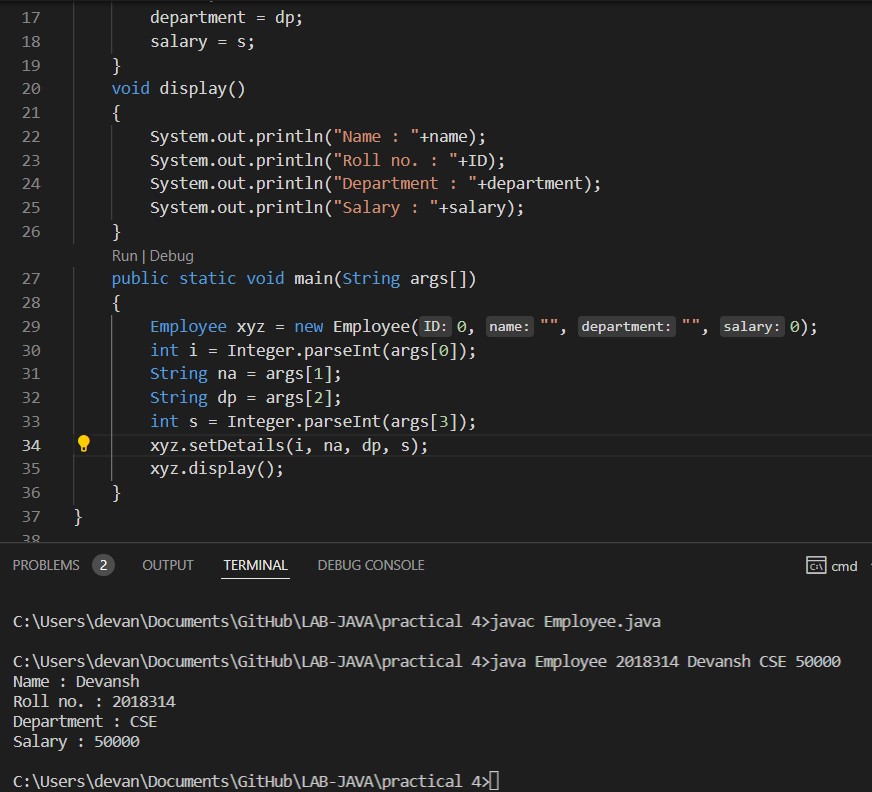
**xyz.setDetails(i, na, dp, s);**

**xyz.display();**

**}**

**}**

**OUTPUT :**

****

**PRACTICAL 5**

**Re-write program 1 with better memory management approach.**

**Note: use of static keyword**

**CODE :**

**public class memory {**

**int roll;**

**String name;**

**String branch;**

**static String univ="gehu";**

**void set(int r, String n, String b)**

**{**

**name = n;**

**roll = r;**

**branch = b;**

**}**

**void display()**

**{**

**System.out.println("Name : "+name);**

**System.out.println("Roll no. : "+roll);**

**System.out.println("Branch : "+branch);**

**System.out.println("University : "+univ);**

**}**

**public static void main(String args[])**

**{**

**memory detail = new memory();**

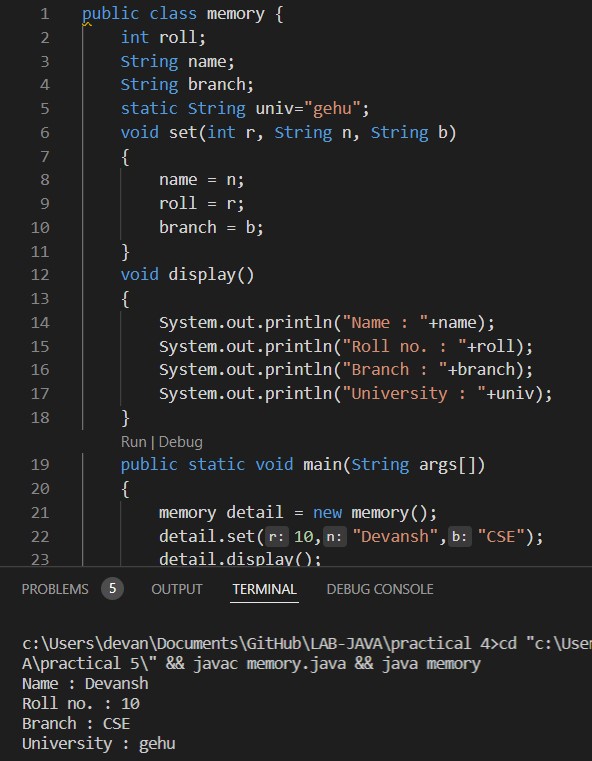
**detail.set(10,"Devansh","CSE");**

**detail.display();**

**}**

**}**

**OUTPUT:**

****

**PRACTICAL 6**

Apply following functions on the String "Java".

(i) Try to concat "Welcome" and write down your observation.

(ii) Find character at index 1

(iii) Find index of first 'a'.

(iv) Find index of second 'a'

(v) Compare "Java" to "JAVA"

(vi) Compare "Java" to "JAVA" ignoring the case

(vii) Find the index of first 'a' from last

**CODE :**

public class handlers {

public static void main(String [] args)

{

String s = "java";

System.out.println(s.concat(" WELCOME")); //doesn't changes original string

System.out.println(s.charAt(1));

System.out.println(s.indexOf('a'));

System.out.println(s.indexOf('a',s.indexOf('a')+1));

System.out.println(s.equals("JAVA"));

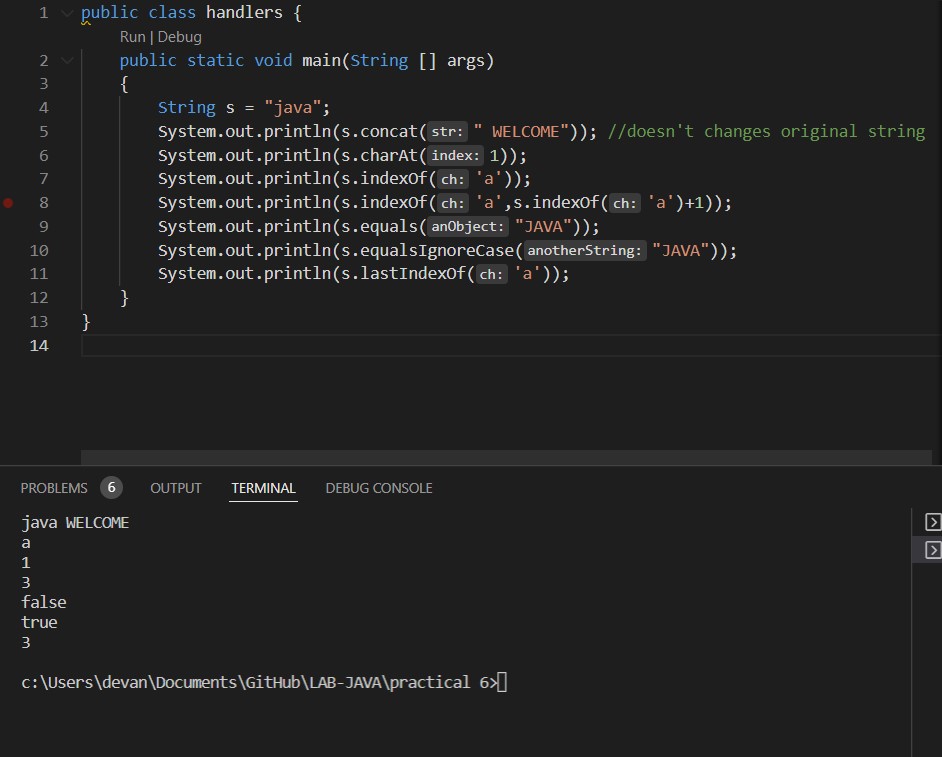
System.out.println(s.equalsIgnoreCase("JAVA"));

System.out.println(s.lastIndexOf('a'));

}

}

**OUTPUT :**

****

**PRACTICAL 7**

Apply following functions on StringBuffer object "HELLO"

(i) Append "Java"

(ii) Insert "Java" at index 1

(iii) Replace with "Java" with characters between index 1 to 2

(iv) Delete characters between index 1 and 2

(v) Reverse the string "HELLO"

**CODE :**

public class handlers {

public static void main(String[] args)

{

StringBuffer s = new StringBuffer("HELLO");

System.out.println(s.append("java"));

System.out.println(s.insert(1,"java"));

System.out.println(s.replace(1, 2, "java"));

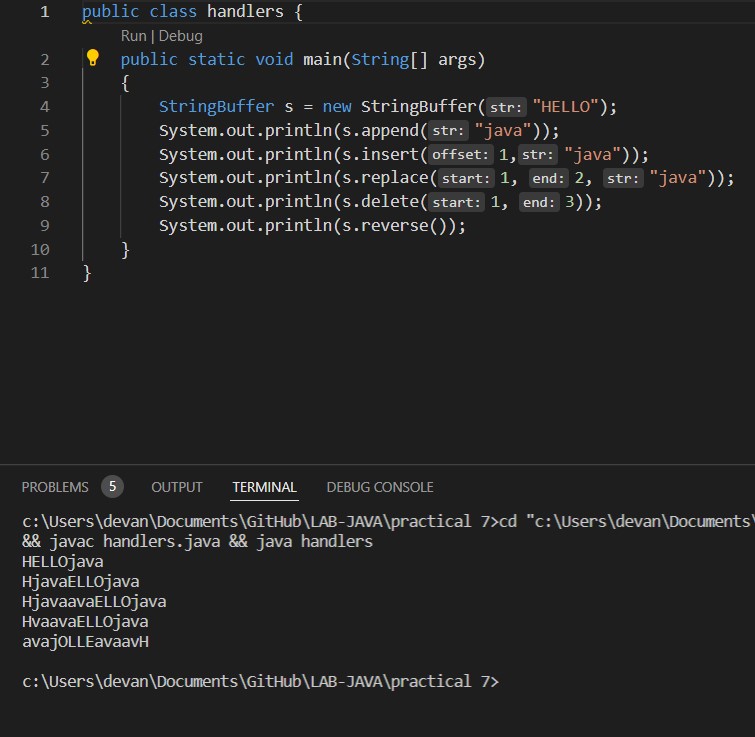
System.out.println(s.delete(1, 3));

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

}

}

**OUTPUT :**

****

**PRACTICAL 8**

Create a class “Student” having following instance variables and methods.

Instance variables: ID, Name, Branch, city and university

While creating constructors with one, two, three, four and five arguments reuse the constructors by **construction chaining**

**CODE :**

public class Student {

int roll;

String name;

String branch;

String city;

String univ;

Student(int roll)

{

this.roll = roll;

}

Student(int roll, String name)

{

this(roll);

this.roll = roll;

this.name = name;

}

Student(int roll, String name, String branch)

{

this(roll,name);

this.roll = roll;

this.name = name;

this.branch = branch;

}

Student(int roll, String name, String branch, String city)

{

this(roll, name, branch);

this.roll = roll;

this.name = name;

this.branch = branch;

this.city = city;

}

Student(int roll, String name, String branch, String city,String univ)

{

this(roll, name, branch, city);

this.roll = roll;

this.name = name;

this.branch = branch;

this.city = city;

this.univ = univ;

}

void display()

{

System.out.println("Name : "+name);

System.out.println("Rollno. : "+roll);

System.out.println("Branch : "+branch);

System.out.println("City : "+city);

System.out.println("University : "+univ);

}

public static void main(String[] args)

{

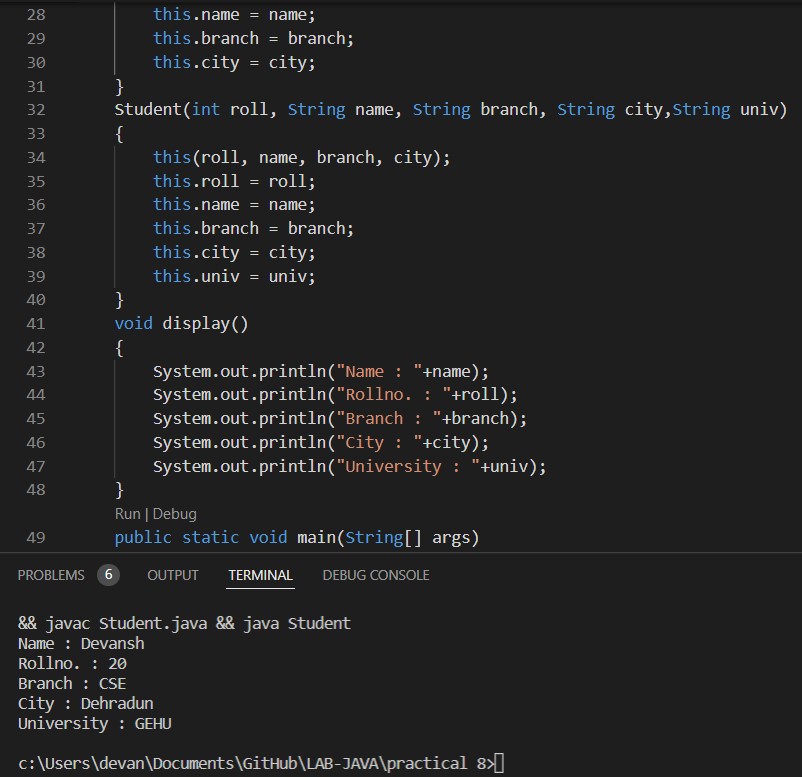
Student detail = new Student(20, "Devansh", "CSE","Dehradun","GEHU");

detail.display();

}

}

**OUTPUT :**

****

**PRACTICAL 9**

Create two dimensional integer array and insert, search and traverse this array.

**Note:** Use Scanner class to insert data.

**CODE :**

import java.io.InputStream;

import java.util.\*;

public class array {

public static void main(String[] args)

{

Scanner sc = new Scanner(System.in);

int arr[][] = new int[3][3]; //input

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

{

for(int j=0; j<arr[i].length; j++)

{

arr[i][j]=sc.nextInt();

}

}

//display

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

{

for(int j=0; j<arr[i].length; j++)

{

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

}

System.out.println("");

}

//search

int key = sc.nextInt();

int f=0;

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

{

for(int j=0; j<arr[i].length; j++)

{

if(arr[i][j]==key)

{

System.out.println(key+" is present");

f=1;

break;

}

}

}

if(f==0)

{

System.out.println(key+" not found");

}

}

}

**PARCTICAL 10**

**Create a jagged array having three rows. Where 1st row contains 3 columns, 2nd row contains 4 columns and 3rd row contains 2 columns. Insert and traverse it.**

**CODE :**

**import java.io.InputStream;**

**import java.util.\*;**

**public class jagged {**

**public static void main(String[] args)**

**{**

**int arr1[][] = new int[3][];**

**arr1[0]=new int[3];**

**arr1[1]=new int[4];**

**arr1[2]=new int[2];**

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

**for(int i=0; i<arr1.length; i++) //input**

**{**

**for(int j=0; j<arr1[i].length; j++)**

**{**

**arr1[i][j]=sc.nextInt();**

**}**

**}**

**for(int i=0; i<arr1.length; i++) //display**

**{**

**for(int j=0; j<arr1[i].length; j++)**

**{**

**System.out.print(arr1[i][j]+" ");**

**}**

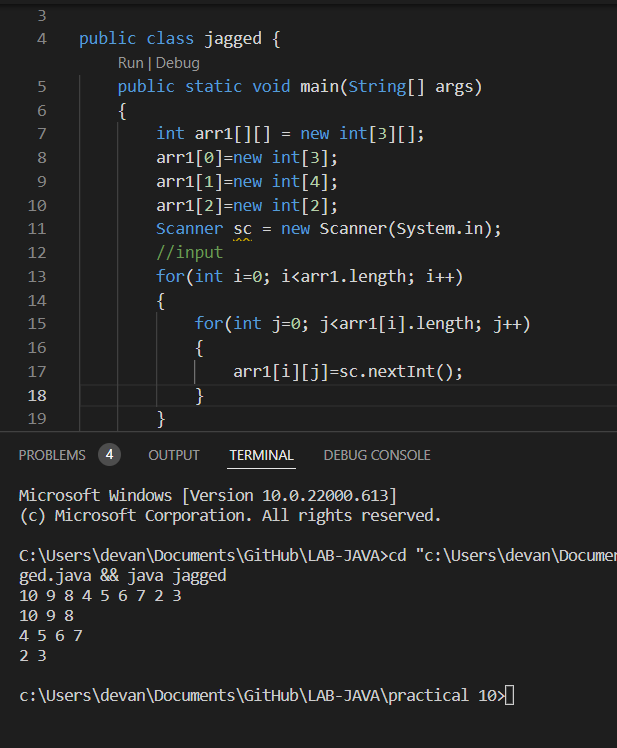
**System.out.println("");**

**}**

**}**

**}**

**OUTPUT :**

****

**PRACTICAL 11**

**Create a class “Shape” having area() method to calculate area. Overload the area() method for shapes like triangle, rectangle and circle.**

**CODE :**

**public class Shape {**

**int area(int a)**

**{**

**return a\*a;**

**}**

**int area(int i, int b)**

**{**

**return i\*b;**

**}**

**int area(float I, float b)**

**{**

**return (int)((I\*b)/2);**

**}**

**int area(int r, double p)**

**{**

**return(int)(p\*r\*r);**

**}**

**public static void main(String[] args)**

**{**

**Shape d = new Shape();**

**System.out.println("Area of square : "+d.area(10));**

**System.out.println("Area of rectangle : "+d.area(10,5));**

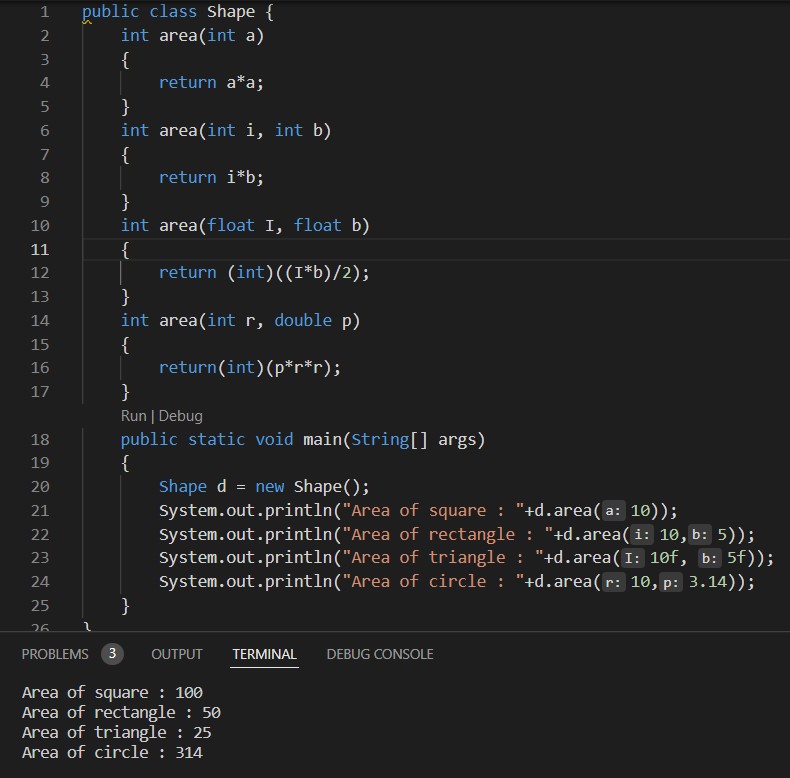
**System.out.println("Area of triangle : "+d.area(10f, 5f));**

**System.out.println("Area of circle : "+d.area(10,3.14));**

**}**

**}**

**OUTPUT :**

****

**PRACTICAL 12**

**Create a class “Bank” having method getRateOfInterest(). Create child classes as HDFC, SBI and PNB and override getRateOfInterest() and return interest rates as 4.0, 4.5 and 5% correspondingly.**

**Use concept of Upcasting to implement this scenario.**

**CODE :**

**class Bank{**

**void getRoi()**

**{**

**System.out.println("5.5%");**

**}**

**}**

**class hdfc extends bank**

**{**

**void getRoi()**

**{**

**System.out.println("hdfc 4%");**

**}**

**}**

**class pnb extends bank**

**{**

**void getRoi()**

**{**

**System.out.println("pnb 4.5%");**

**}**

**}**

**class sbi extends bank**

**{**

**void getRoi()**

**{**

**System.out.println("sbi 5%");**

**}**

**}**

**class call**

**{**

**public static void main(String[] args) {**

**bank r;**

**b=new bank();**

**b.getRoi();**

**b=new pnb();**

**b.getRoi();**

**b=new hdfc();**

**b.getRoi();**

**b=new sbi();**

**b.getRoi();**

**}**

**}**

**OUTPUT :**

**Text

Description automatically generated**