Course Title	Programming with JAVA lab				Course Type		HC	
Course Code	B20EF0306	Credits	1		Class		III Semester	
Course Structure	TLP	Credits	Contact Hours	Work Load	Total Number of Classes Per Semester		Assessment in Weightage	
	Theory	-	-	-				
	Practice	1	2	2	Theory	Practical	CIE	SEE
	-	-	-	-				
	Total	**1**	**2**	**2**	**-**	**26**	**25**	**25**

COURSE OVERVIEW:

This Laboratory course supplements the material taught in the theory course programming with J2SE (Java2 standard Edition). The objective of this course is to get hands-on experience in JAVA programming and implementing the concepts learnt in the theory course. Laboratory exercises will normally be conducted using windows operating System with Eclipse Integrated Development Environment (IDE). The Students will experience the classes, objects, Exception handling, wrapper classes, strings and collection framework and also thread concepts.

COURSE OBJECTIVE (S):

The objectives of this course are to:

1. Learn fundamentals of object-oriented programming in Java using hands-on.
2. Familiarize Java environment to create, debug and run simple Java programs using JDK, JRE, and JVM and also learn how to use Integrated Development Environment (IDE) either Eclipse /Net beans to create Java Application.
3. Demonstrate Primitives types as objects using wrapper classes and Collection Framework Data Structure and importance of strings, Exception Handling, Abstract classes and interfaces.
4. Develop an application to address real time issues.

COURSE OUTCOMES (COs)

After the completion of the course, the student will be able to:

CO#	Course Outcomes	POs	PSOs
CO1	Apply basic syntactic concepts of creating classes, Objects and constructors with all primitive types and Control statements.	1 to 3,5,9,10,12	1
CO2	Solve problems using Access Modifies, polymorphism, inheritance etc.	1 to 5, 9,10,12	1,3
CO3	Develop solutions using the concepts of abstract classes, interfaces, packages.	1 to 5, 9,10,12	1,2, 3
CO4	Interpret the concepts of Exception Handling, Strings.	1 to 5, 9,10,12	1,2, 3
CO5	Design Java applications using objects instead of primitives using wrapper classes and Collection Framework Data structure, Inter thread communication	1 to 6, 9,10,12	1,2, 3

CO6	Design and develop solution to address real time applications.	1 to 6, 9,10,12	1,2, 3

BLOOM'S LEVEL OF THE COURSE OUTCOMES

CO#	Bloom's Level					
	Remember (L1)	Understand (L2)	Apply (L3)	Analyze (L4)	Evaluate (L5)	Create (L6)
CO1			√			
CO2					√	
CO3						√
CO4					√	
CO5						√
CO6						√

COURSE ARTICULATION MATRIX

CO#/ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	2	2		1				1	2		2	2		
CO2	1	2	2	2	1				2	2		2	2		3
CO3	1	2	1	1	1				2	2		2	2	3	3
CO4	1	2	3	1	2				2	2		2	3	3	3
CO5	1	2	2	1	3	2			3	2		2	3	3	3
CO6	1	2	2	1	3	2			3	2		3	3	3	3

Note: 1-Low, 2-Medium, 3-High

PRACTICE:

No	Title of the Experiment	Tools and Techniques	Expected Skill /Ability
	Part-A		
1.	Write a Java program using class and objects concept, to read two distances (in feet and inches) and print their sum in feet and inches, note that if total inches are more than 12 then it would be consider as 1 feet.	Windows/Linux OS, IDE	Understanding class and objects.
2.	Write a java program to find ncr and npr using recursion.	Windows/Linux OS, IDE	Applying concept of Recursion using java
3.	Write a java program to read 2 decimal numbers, convert decimal number to binary number and also find sum of 2 binary numbers.	Windows/Linux OS, IDE	Understanding the concept of Constructors, Control
4.	Write a java program to demonstrate the working of the banking system, where it consisting of the functionalities a. Display all account details b. Search by account number c. Deposit the amount d. Withdraw the amount	Windows/Linux OS, IDE	Understanding the concept of instance methods and instance variables
5.	Write a java program to check whether the given matrix is sparse matrix or not. And also find sum of all the elements of the sparse matrix	Windows/Linux OS, IDE	Usage of Arrays
6.	Write a java program to demonstrate the User defined exception.	Windows/Linux OS, IDE	Creation of User defined Exception using Inheritance
7	Write a java program to demonstrate Serializable Marker interface.	Windows/Linux OS, IDE	Understanding Serialization and Deserialization using marker interface concepts
8	Consider Banking Application to pay the monthly EMI for the given principal amount, duration and rate of interest. Demonstrate this application using abstract class and interfaces. Note: At least consider minimum three Banks classes	Windows/Linux OS, IDE	Understanding data abstraction using abstract classes and interface
9	Write a java program to swap (exchange) first and last character of each word in the given string.	Windows/Linux OS, IDE	Usage of immutable concepts using Strings.

10	Write a java program using Hash Map to check two strings are Anagram or not.	Windows/Linux OS, IDE	Application of Map Interface in strings
11	Write a Java program to Demonstrate Comparator interface and Array list to sort student's information by considering student names and students roll number.	Windows/Linux OS, IDE	Creation of List interface to demonstrate the application of comparator interface.
12	Write a java program for producer and consumer problem.	Windows/Linux OS, IDE	Creation of threads in inter process communication

Sl. No.	Part B Mini Project
1	Develop a project for Airline reservation system List with the following modules: 1. PASSENGER a) Add member b) Delete member c) Search for member d) Edit member 2. FLIGHT a. Add Flight b. Delete Flight c. Search Flight d. Display Flights 3. RESERVATION a. Book b. Cancel Title: Airline Reservation system Problem Definition: *Airline Reservation System*" main aim is to provide the online ticket & seat reservation of National and International Flights and give the information about flight departures. Solution: Develop a project to implement an Airline reservation system with the following modules: 1. PASSENGER a. Add member b. Delete member c. Search for member

	d. Edit member 2. FLIGHT a. Add Flight b. Delete Flight c. Search Flight d. Display Flights 3. RESERVATION a. Book b. Cancel

TEXT BOOKS:

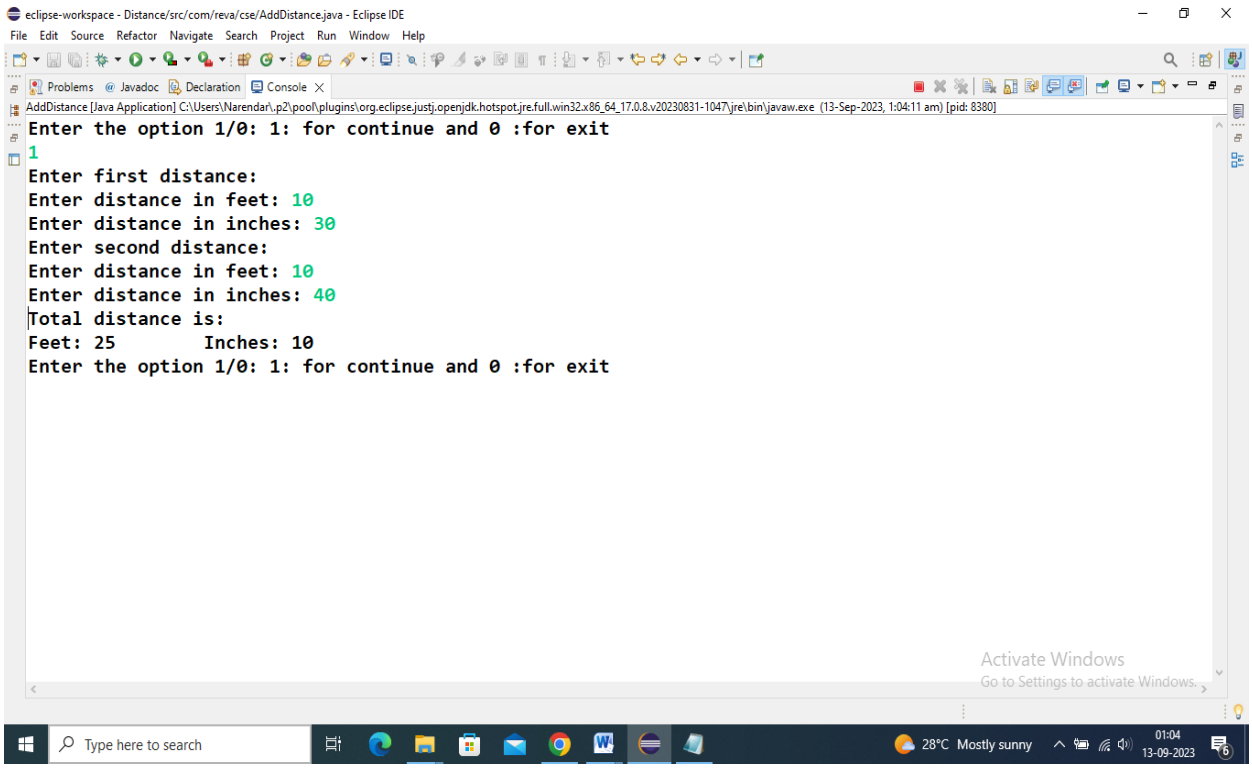
1. Cay S. Horstmann, “Core Java® SE 9 for the Impatient”, Addison Wesley, Second Edition, 2018.
2. HerbertSchild, “Java™: TheCompleteReference”, McGraw-Hill, Tenth Edition,2018.
3. David Gallardo, Ed Burnette, Robert Mcgovern, “Eclipse in Action a guide for java developers”, Manning Publications, 2003.
4. Ed Burnette, “Eclipse IDE Pocket Guide: Using the Full-Featured IDE”, O'Reilly Media, Inc, USA, 2005.

1) **Write a Java program using class and objects concept, to read two distances (in feet and inches) and print their sum in feet and inches, note that if total inches are more than 12 then it would be consider as 1 feet.**

```
package com.reva.cse;
import java.util.Scanner;
public class AddDistance
{
        int feet;
        int inches;

        public void Read()
        {
         Scanner sc=new Scanner(System.in);
          System.out.print("Enter distance in feet: ");
          feet = sc.nextInt();
          System.out.print("Enter distance in inches: ");
          inches = sc.nextInt();
        }

        public void Display()
        {
          System.out.println("Feet: " + feet + "\tInches: " + inches);
        }

        public void Addition(AddDistance D1, AddDistance D2)
        {
                inches = D1.inches + D2.inches;
                feet = D1.feet + D2.feet + (inches / 12);
          inches = inches % 12;
        }

       public static void main(String[] args)
       {
                while(true)
                {
                System.out.println("Enter the option 1/0: 1: for continue and 0 :for exit");
                Scanner sc=new Scanner(System.in);
                        int val=sc.nextInt();
                   switch(val)
                   {
                     case 1:
                          AddDistance D1 = new AddDistance();
                          AddDistance D2 = new AddDistance();
                          AddDistance D3 = new AddDistance();

                            //read first distance
                            System.out.println("Enter first distance: ");
                            D1.Read();

                            //read second distance
                            System.out.println("Enter second distance: ");
                            D2.Read();

                            //add distances
                            D3.Addition(D1, D2);

                            //print distance
                            System.out.println("Total distance is:");
                            D3.Display();
                            break;
```

```
                    case 0:System.exit(0);
                    default:System.out.println("Enter the valid Input 1/0");
                  }
                }
            }
    }
```

Output:

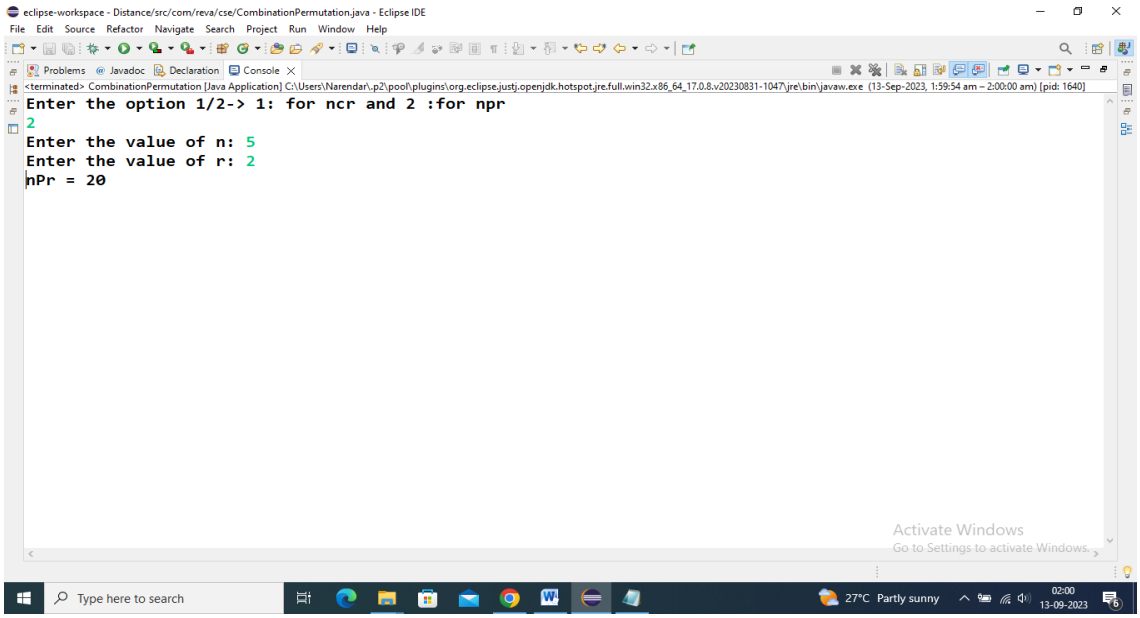


2 Write a java program to find nCr and nPr using recursion.

```
package com.reva.cse;
import java.util.Scanner;
public class CombinationPermutation
{
  // Function to find factorial
   static int factorial(int n)
   {
      if (n == 0 || n == 1)
         return 1;
      else
         return n * factorial(n - 1);
   }
   // Function to calculate nCr
   static int nCr(int n, int r)
   {
      if (r == 0 || r == n)
         return 1;
      else
         return factorial(n) / (factorial(r) * factorial(n - r));
   }
```

```
    // Function to calculate nPr
    static int nPr(int n, int r)
    {
        if (r == 0)
            return 1;
        else
            return factorial(n) / factorial(n - r);
    }
    public static void main(String[] args)
    {
                System.out.println("Enter the option 1/2-> 1: for ncr and 2 :for npr");
                Scanner sc=new Scanner(System.in);
                int val1=sc.nextInt();
                System.out.print("Enter the value of n: ");
                int n = sc.nextInt();
                 System.out.print("Enter the value of r: ");
                 int r = sc.nextInt();
                switch(val)
                {
                   case 1:
                               if (n >= r)
                               {
                                   System.out.println("nCr = " + nCr(n, r));
                               } else
                               {
                                  System.out.println("Invalid input! n should be greater than or equal to r.");
                               }
                               break ;

                  case 2:
                               if (n >= r)
                               {
                                   System.out.println("nPr = " + nPr(n, r));
                               } else
                               {
                                  System.out.println("Invalid input! n should be greater than or equal to r.");
                               }
                               break ;

                 default:  System.out.println("Invalid Option");

              }
        }
}
```

Output:



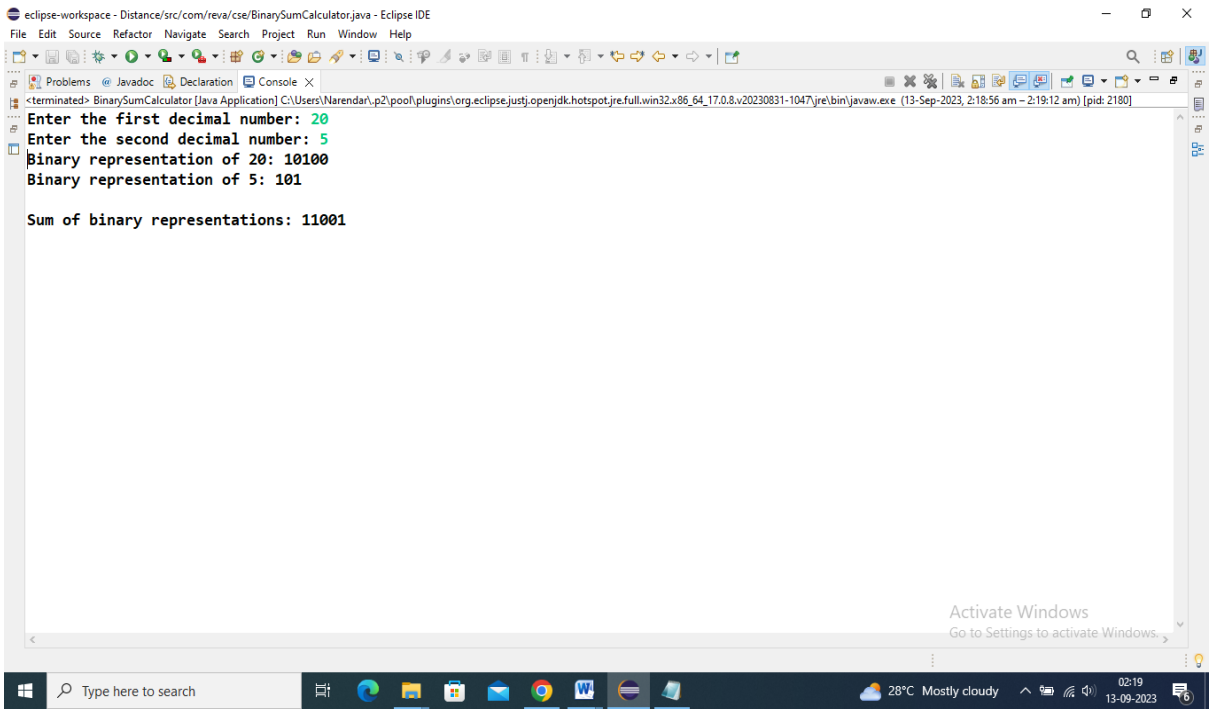
3) Write a java program to read 2 decimal numbers, convert decimal number to binary number and also find sum of 2 binary numbers.

Note: Don't use any built-in methods to convert decimal number to binary number

```
package com.reva.cse;
import java.util.Scanner;
public class BinarySumCalculator
{
   public static void main(String[] args)
   {
      Scanner scanner = new Scanner(System.in);
      // Read the decimal numbers
      System.out.print("Enter the first decimal number: ");
      int decimal1 = scanner.nextInt();
      System.out.print("Enter the second decimal number: ");
      int decimal2 = scanner.nextInt();
      // Convert decimal numbers to binary strings
      String binary1 = decimalToBinary(decimal1);
      String binary2 = decimalToBinary(decimal2);
      System.out.println("Binary representation of " + decimal1 + ": " + binary1);
      System.out.println("Binary representation of " + decimal2 + ": " + binary2);
      // Calculate the sum of binary numbers
      String sumBinary = addBinary(binary1, binary2);
      System.out.println("\nSum of binary representations: " + sumBinary);
      scanner.close();
   }

   // Convert decimal to binary
   public static String decimalToBinary(int decimal)
   {
      if (decimal == 0)
      {
         return "0";
      }
      StringBuilder binary = new StringBuilder();
      while (decimal > 0)
      {
         int remainder = decimal % 2;
```

```
            binary.insert(0, remainder);
            decimal /= 2; //decimal=decimal/2
        }
        return binary.toString();
    }
    // Add two binary strings
    public static String addBinary(String binary1, String binary2)
    {
        StringBuilder result = new StringBuilder();
        int carry = 0;
        int i = binary1.length() - 1;
        int j = binary2.length() - 1;
        while (i >= 0 || j >= 0 || carry > 0)
        {
            int digit1 = i >= 0 ? binary1.charAt(i--) - '0' : 0;
            int digit2 = j >= 0 ? binary2.charAt(j--) - '0' : 0;
            int sum = digit1 + digit2 + carry;
            result.insert(0, sum % 2);
            carry = sum / 2;
        }
        return result.toString();
    }
}
```

Output:



4) **Write a Java Program to demonstrate the working of the banking system, where it consisting of the functionalities.**
    a. **Display all account details.**
    b. **Search by account number**
    c. **Deposit the amount**
    d. **Withdraw the amount**

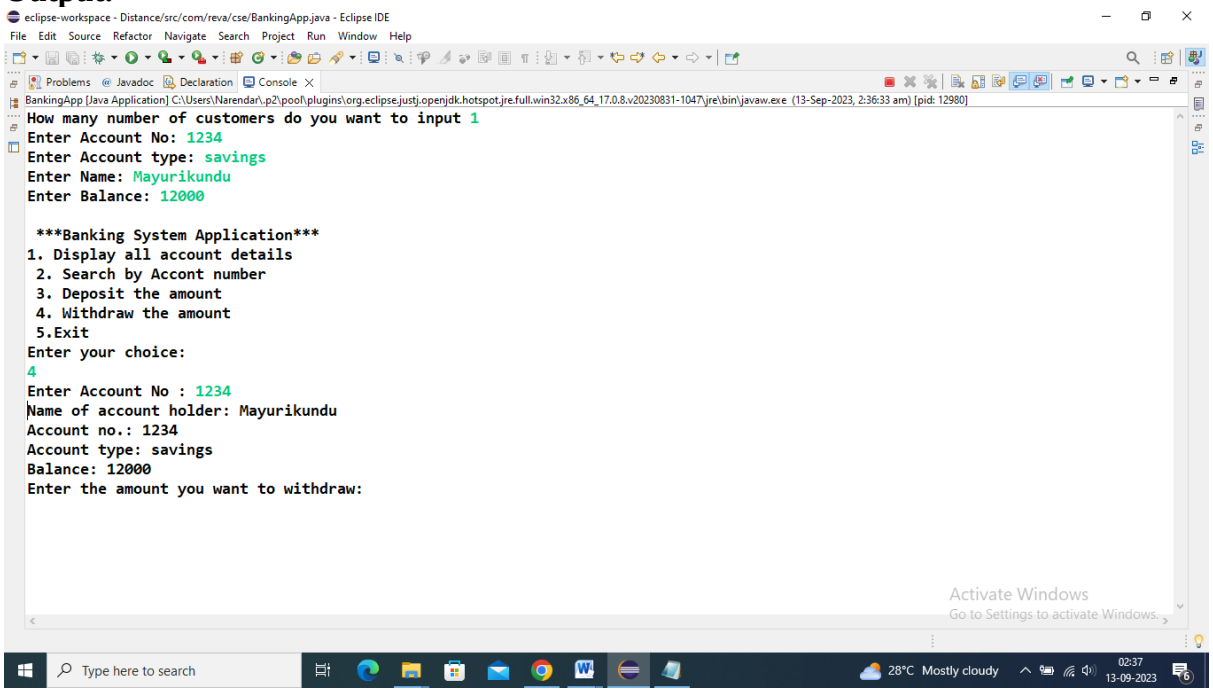
```
package com.reva.cse;
```

```
import java.util.Scanner;
class BankDetails
{
   private String accno;
   private String name;
   private String acc_type;
   private long balance;
   Scanner sc = new Scanner(System.in);
   //method to open new account
   public void openAccount()
   {
                System.out.print("Enter Account No: ");
                accno = sc.next();
                System.out.print("Enter Account type: ");
                acc_type = sc.next();
                System.out.print("Enter Name: ");
                name = sc.next();
                System.out.print("Enter Balance: ");
                balance = sc.nextLong();
   }
   //method to display account details
   public void showAccount()
   {
                System.out.println("Name of account holder: " + name);
                System.out.println("Account no.: " + accno);
                System.out.println("Account type: " + acc_type);
                System.out.println("Balance: " + balance);
   }
   //method to deposit money
   public void deposit()
   {
                long amt;
                System.out.println("Enter the amount you want to deposit: ");
                amt = sc.nextLong();
                balance = balance + amt;
   }
   //method to withdraw money
   public void withdrawal()
   {
                long amt;
                System.out.println("Enter the amount you want to withdraw: ");
                amt = sc.nextLong();
                if (balance >= amt)
                {
                   balance = balance - amt;
                   System.out.println("Balance after withdrawal: " + balance);
                } else
                {
                   System.out.println("Your balance is less than " + amt +
                    "\tTransaction failed...!!" );
                }
   }
   //method to search an account number
   public boolean search(String ac_no)
   {
      if (accno.equals(ac_no))
      {
         showAccount();
         return (true);
      }
      return (false);
   }
```

```
}
        public class BankingApp
        {
           public static void main(String arg[])
           {
             Scanner sc = new Scanner(System.in);
              //create initial accounts
             System.out.print("How many number of customers do you want to input ");
             int n = sc.nextInt();
             BankDetails C[] = new BankDetails[n];
             for (int i = 0; i < C.length; i++)
             {
                C[i] = new BankDetails();
                C[i].openAccount();
             }
              // loop runs until number 5 is not pressed to exit
              int ch;
              do {
                  System.out.println("\n ***Banking System Application***");
                  System.out.println("1. Display all account details \n 2. Search by Accont number\n "
                            + "3. Deposit the amount \n 4. Withdraw the amount \n 5.Exit ");
                 System.out.println("Enter your choice: ");
                 ch = sc.nextInt();
                    switch (ch) {
                       case 1:
                          for (int i = 0; i < C.length; i++)
                          {
                             C[i].showAccount();
                          }
                          break;
                       case 2:
                          System.out.print("Enter account no. you want to search: ");
                          String ac_no = sc.next();
                          boolean found = false;
                          for (int i = 0; i < C.length; i++)
                          {
                             found = C[i].search(ac_no);
                             if (found)
                             {
                                break;
                             }
                          }
                          if (!found) {
                             System.out.println("Search failed! Account doesn't exist..!!");
                          }
                          break;
                       case 3:
                          System.out.print("Enter Account no. : ");
                          ac_no = sc.next();
                          found = false;
                          for (int i = 0; i < C.length; i++) {
                             found = C[i].search(ac_no);
                             if (found) {
                                C[i].deposit();
                                break;
                             }
                          }
                          if (!found) {
                             System.out.println("Search failed! Account doesn't exist..!!");
                          }
                          break;
```

```
                case 4:
                    System.out.print("Enter Account No : ");
                    ac_no = sc.next();
                    found = false;
                    for (int i = 0; i < C.length; i++) {
                        found = C[i].search(ac_no);
                        if (found) {
                            C[i].withdrawal();
                            break;
                        }
                    }
                    if (!found) {
                        System.out.println("Search failed! Account doesn't exist..!!");
                    }
                    break;
                case 5:
                    System.out.println("See you soon...");
                    break;
            }
        }
        while (ch != 5);
    }
}
```

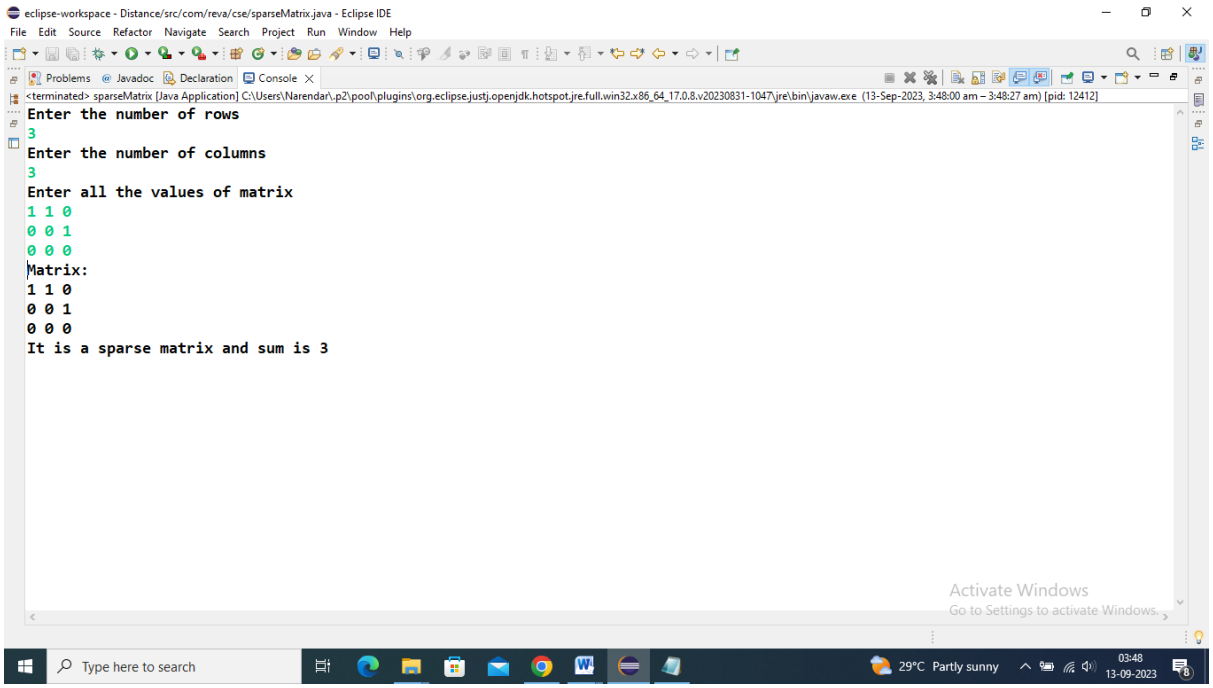
Output:



5) Write a java program to check whether the given matrix is sparse matrix or not. And also find sum of all the elements of the sparse matrix.

```
package com.reva.cse;
import java.util.Scanner;
public class sparseMatrix
{
        public static void main(String[] args)
```

```
  {
    // declare variables
    int m, n;
    // To take input from the user
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the number of rows ");
    // Initialize the number of rows
    m = sc.nextInt();
    System.out.println("Enter the number of columns ");
    // Initialize the number of columns
    n = sc.nextInt();
    // declare a mxn order array
    int a[][] = new int[m][n];
    System.out.println("Enter all the values of matrix ");
    // Initialize the matrix elements
    for (int i = 0; i < m; i++)
    {
      for (int j = 0; j < n; j++)
      {
        a[i][j] = sc.nextInt();
      }
    }
    System.out.println("Matrix:");
    for (int i = 0; i < m; i++)
    {
      for (int j = 0; j < n; j++)
      {
          System.out.print(a[i][j] + " ");
      }
      System.out.println("");
    }
    int size= m*n;   //Stores the size of the matrix
    int count=0;    //Variable to check for the number of 0 elements
    int sum = 0;    //calculate sum if the element is non-zero
    //Loop to count all zero element present in matrix
    for(int i = 0; i < m; i++)
    {
      for(int j = 0; j < n; j++)
      {
        if(a[i][j] == 0)    //Check if element is 0 or not
          count++;    //Increment the count if 0 element is found
          else sum = sum + a[i][j];    //calculate sum
      }
    }
    if(count>(size/2))
    System.out.println("It is a sparse matrix and sum is "+ sum);
    else
    System.out.println("It is not a sparse matrix");
  }
}
```

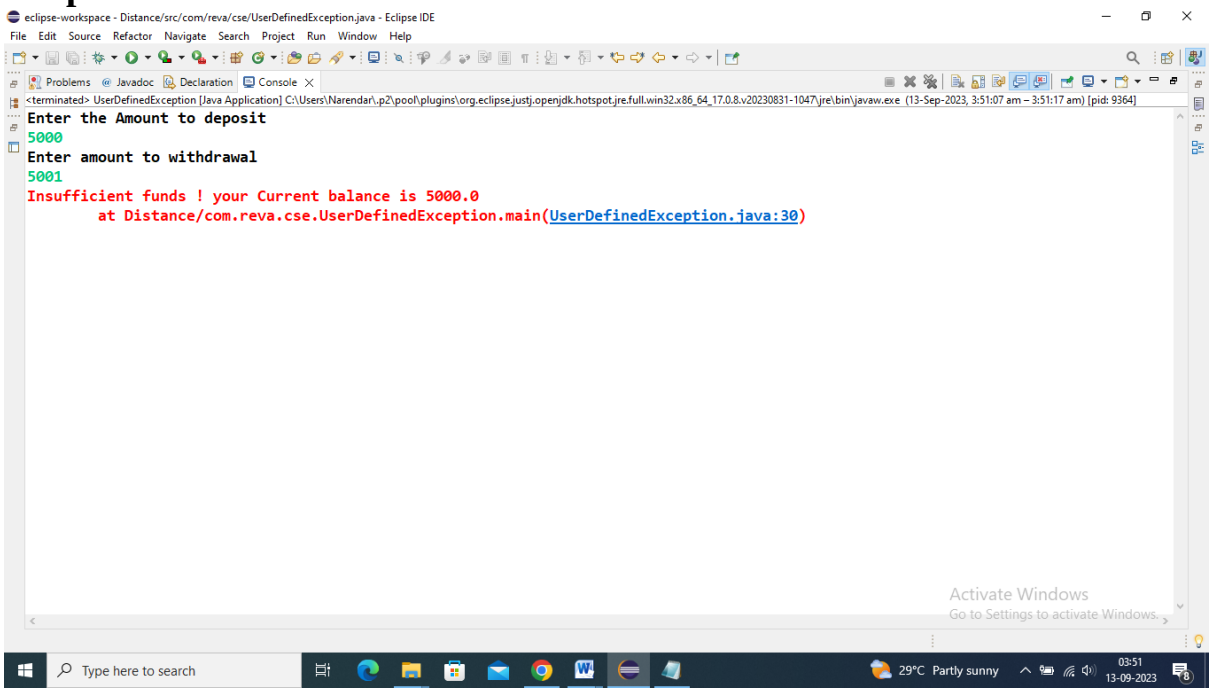


6) Write a java program to demonstrate the User defined exception.

```
package com.reva.cse;
import java.util.Scanner;
class Balance extends Exception
{
        String message;
        public Balance(String message)
        {
           this.message = message;
        }
        @Override
        public String toString()
        {
           return message;
        }
}

public class UserDefinedException
{
        static double current_balance ;
        public static void main(String[] args) throws Balance
        {
           Scanner s = new Scanner(System.in);
           System.out.println("Enter the Amount to deposit");
           current_balance=s.nextInt();
           System.out.println("Enter amount to withdrawal");
           int n = s.nextInt();
           try {
                   if (current_balance < n)
                   {
                      throw new Balance("Insufficient funds ! your Current balance is " +
                      current_balance);
                   } else
                   {
```

```
                    System.out.println("Please Take The Money : " + n);
                    current_balance=current_balance-n;
                    System.out.println("Balance Available After withdral : " + current_balance);
                }
            }
            catch (Balance mab)
            {
             mab.printStackTrace();
            }
        }
}
```

Output:

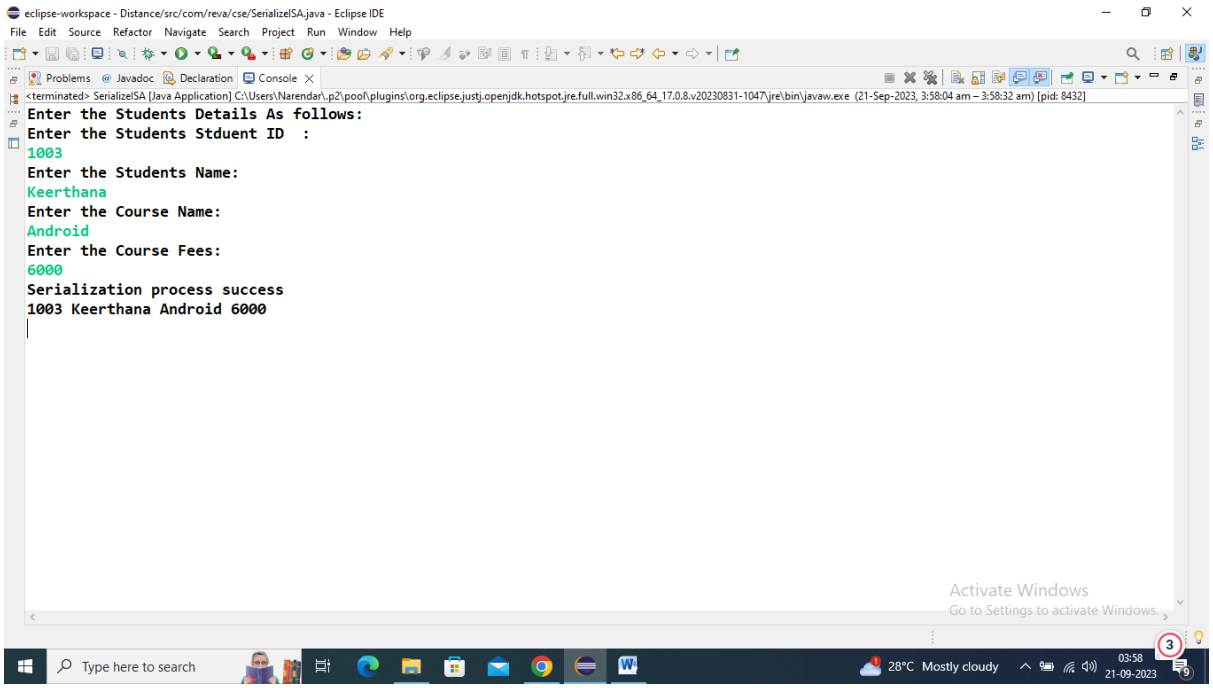


7) Write a java program to demonstrate Serializable Marker interface.

```
package com.reva.cse;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;
import java.util.Scanner;

class Person implements Serializable
{
        int id;
        String name;
        Person(int id, String name)
        {
         this.id = id;
         this.name = name;
        }
}

class Student extends Person
```

```
{
        String course;
        int fee;
        public Student(int id, String name, String course, int fee)
        {
         super(id,name);
         this.course=course;
         this.fee=fee;
        }
}
public class SerializeExample
{
        public static void main(String args[])
        {
                try
                {
                        //Creating the object
                  System.out.println("Enter the Students Details As follows:");
                  Scanner sc=new Scanner(System.in);
                  System.out.println("Enter the Students Stduent ID  :");
                  int id=sc.nextInt();
                  System.out.println("Enter the Students Name:");
                  String name=sc.next();
                  System.out.println("Enter the Course Name:");
                  String course=sc.next();
                  System.out.println("Enter the Salary:");
                  int salary=sc.nextInt();
                        Student s1 =new Student(id,name,course,salary);
                        //Creating stream and writing the object
                        FileOutputStream fout=new FileOutputStream("f.txt");
                        ObjectOutputStream out=new ObjectOutputStream(fout);
                        out.writeObject(s1);
                        out.flush();
                        //closing the stream
                        out.close();
                        System.out.println("Serialization process success");
                 }
                 catch(Exception e)
                 {
                        System.out.println(e);
                 }
                try
                {
                        //Creating stream to read the object
                       ObjectInputStream in=new ObjectInputStream(new FileInputStream("f.txt"));
                        Student s=(Student)in.readObject();
                        //printing the data of the serialized object
                        System.out.println(s.id+" "+s.name+" "+s.course+" "+s.fee);
                        //closing the stream
                        in.close();
                }
                catch(Exception e)
                {
                        System.out.println(e);
                 }
        }
}
```

Output:



8) Consider Banking Application to pay the monthly EMI for the given principal amount, duration and rate of interest. Demonstrate this application using abstract class and interfaces.

Note: At least consider minimum three Banks classes

```
package com.cse.reva;
import java.util.Scanner;
//Abstract class for a Bank
abstract class Bank
{
 public abstract double calculateEMI(double principal, int duration, double rateOfInterest);
}

//Bank1 class implementing the LoanCalculator interface
class HDFC extends Bank
{
 @Override
 public double calculateEMI(double principal, int duration, double rateOfInterest)
 {
    double monthlyInterestRate = rateOfInterest / 1200;
    double emi = (principal * monthlyInterestRate) / (1 - Math.pow(1 + monthlyInterestRate, -duration));
    return emi;
 }
}

class SBI extends Bank
{
 @Override
 public double calculateEMI(double principal, int duration, double rateOfInterest)
 {
    double monthlyInterestRate = rateOfInterest / 1200;
    double emi = (principal * monthlyInterestRate) / (1 - Math.pow(1 + monthlyInterestRate, -duration));
    return emi;
 }
}
class IDFC extends Bank
```

```
{

 @Override
 public double calculateEMI(double principal, int duration, double rateOfInterest)
 {
    double monthlyInterestRate = rateOfInterest / 1200;
    double emi = (principal * monthlyInterestRate) / (1 - Math.pow(1 + monthlyInterestRate, -duration));
    return emi;
 }
}

public class Main {
 public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter Principal Amount: INR");
    double principal = scanner.nextDouble();
    System.out.print("Enter Duration (in months): ");
    int duration = scanner.nextInt();
    System.out.print("Enter Rate Interest for HDFC (%): ");
    double rateOfInteresthdfc = scanner.nextDouble();
    System.out.print("Enter Rate Interest for SBI (%): ");
    double rateOfInterestsbi = scanner.nextDouble();
    System.out.print("Enter Rate Interest for IDFC (%): ");
    double rateOfInterestidfc = scanner.nextDouble();
    HDFC hdfc = new HDFC();
    SBI sbi=new SBI();
    IDFC idfc=new IDFC();
    double emihdfc = hdfc.calculateEMI(principal, duration, rateOfInteresthdfc);
    double emisbi = bank1.calculateEMI(principal, duration, rateOfInterestsbi);
    double emiidfc = bank1.calculateEMI(principal, duration, rateOfInterestidfc);
    System.out.println("Monthly EMI (HDFC): INR =>" + emihdfc);
    System.out.println("Monthly EMI (SBI): INR =>" + emisbi);
    System.out.println("Monthly EMI (IDFC): INR =>" + emiidfc);
 }
}
```

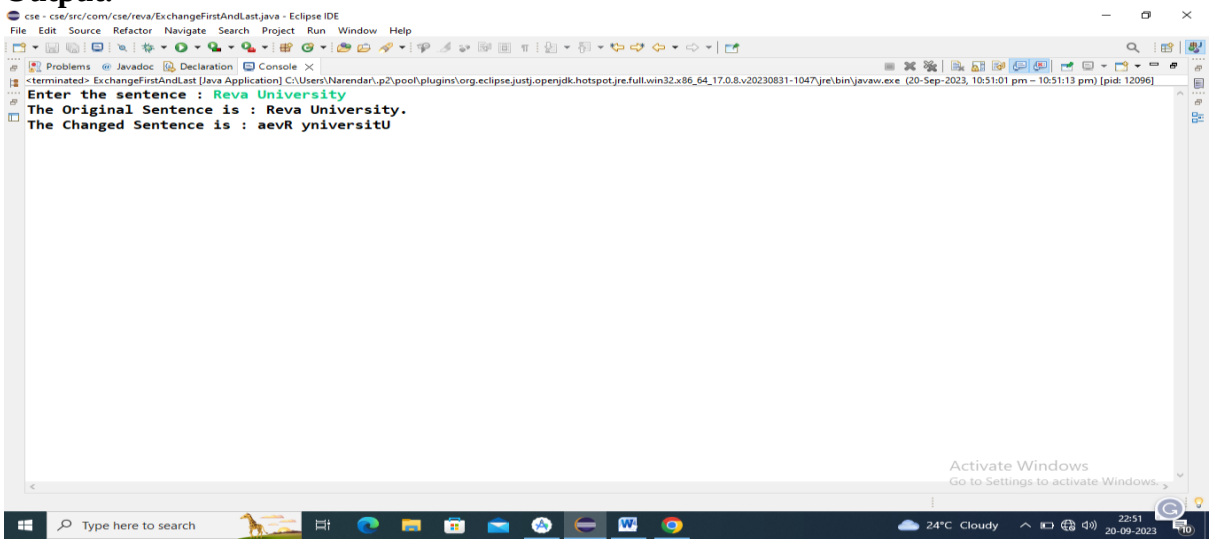
9) Write a java program to swap (exchange) first and last character of each word in the given string.

```
package com.cse.reva;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

public class ExchangeFirstAndLast
{
        // create object.
        static BufferedReader br=new BufferedReader (new InputStreamReader (System.in));
        String sentence, reverse;
        int size;
        // default constructor
        ExchangeFirstAndLast()
        {
                sentence="";
                reverse="";
                size=0;
        }
        // create function to read sentence.
        void ReadSentence()throws IOException
        {
                // enter the sentence here.
```

```
		System.out.print("Enter the sentence : ");

		sentence=br.readLine();
		size=sentence.length();

		// check the ending of sentence with full stop.
		if(sentence.charAt(size-1)!='.')
		{
			// if it is not finished with '.' then add it in last.
			sentence=sentence+".";
			size=size+1;
		}
}
void exfirstlast()
{
		// create string variable.
		String s1="";
		char ch;
		for(int i=0;i<size;i++)
		{
			ch=sentence.charAt(i);
			if(ch!=' ' && ch!='.')
			{
				s1=s1+ch;
			}
			else
			{
				// find length of the word.
				int l=s1.length();

				for(int j=0;j<l;j++)
				{
					// exchange the first alphabet with the last
					if(j==0)
						ch=s1.charAt(l-1);
					// exchange the last alphabet with the first
					else if(j==(l-1))
						ch=s1.charAt(0);
					else
						ch=s1.charAt(j);
					reverse=reverse+ch;
				}
				reverse=reverse+" ";
				s1="";
			}
		}
}

// create display function.
void display()
{
		System.out.println("The Original Sentence is : "+sentence);
		System.out.println("The Changed Sentence is : "+reverse);
}

public static void main(String args[])throws IOException
{
		ExchangeFirstAndLast ob=new ExchangeFirstAndLast();
		ob.ReadSentence();
		ob.exfirstlast();
		ob.display();
}
```

}

Output:

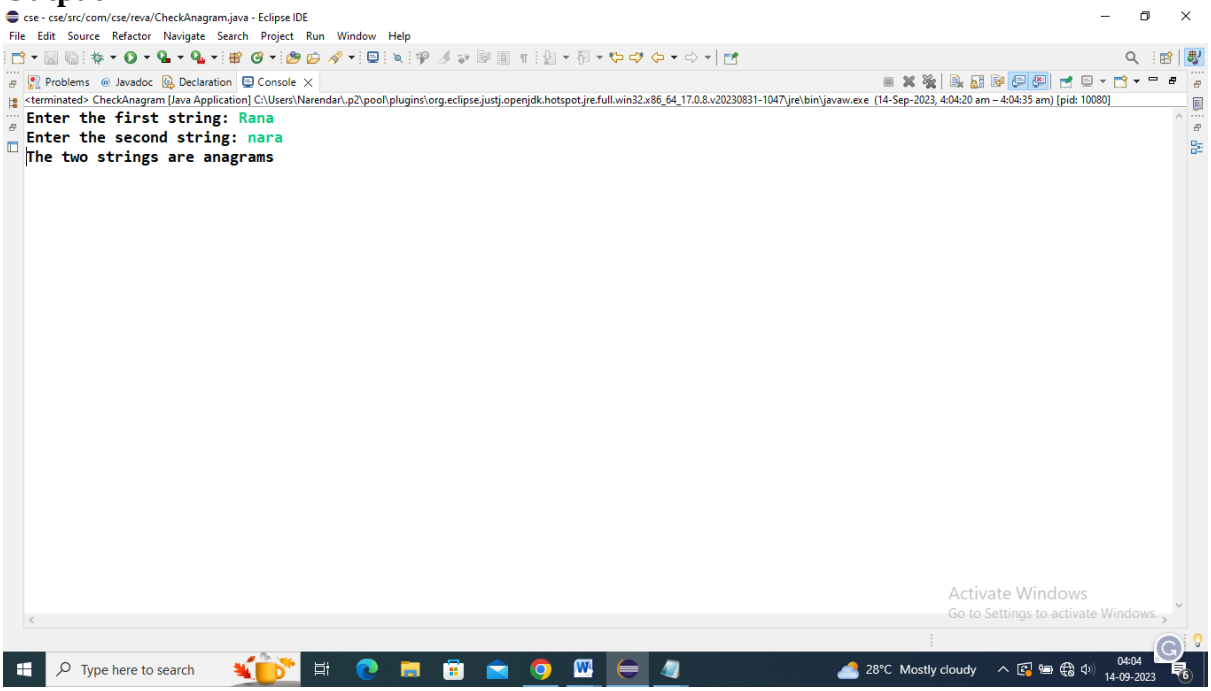


10) Write a java program using Hash Map to check two strings are Anagram or not.

```
import java.util.HashMap;
import java.util.Scanner;
public class CheckAnagram
{
        public static void main(String args[])
        {
            Scanner scanner = new Scanner(System.in);
            System.out.print("Enter the first string: ");
            String str1 = scanner.nextLine();
            System.out.print("Enter the second string: ");
            String str2 = scanner.nextLine();
            // convert strings to lowercase
            str1 = str1.toLowerCase();
                str2 = str2.toLowerCase();
                // initialize hashmaps
          HashMap<Character,Integer>hmap1=newHashMap<Character,Integer>();
HashMap<Character, Integer> hmap2 = new HashMap<Character,Integer>();
                //convert string to character array
                char arr1[] = str1.toCharArray();
                char arr2[] = str2.toCharArray();
                //for loop for first string
                for (int i = 0; i < arr1.length; i++)
                {
                        //if character not present add to hashmap
                        if (hmap1.get(arr1[i]) == null)
                        {
                                hmap1.put(arr1[i], 1);
                        } else
                        {
                                Integer c = (int) hmap1.get(arr1[i]);
                                hmap1.put(arr1[i], ++c);
              }
            }
                //for loop for second string
                   for (int j = 0; j < arr2.length; j++)
```

```
                {
                 if (hmap2.get(arr2[j]) == null)
                   hmap2.put(arr2[j], 1);
                 else
                 {
                   Integer d = (int) hmap2.get(arr2[j]);
                   hmap2.put(arr2[j], ++d);
                 }
                }
            //check if hmaps are equal
            if (hmap1.equals(hmap2))
              System.out.println("The two strings are anagrams");
            else
              System.out.println("The two strings are NOT anagrams");
          }
        }
```

Output



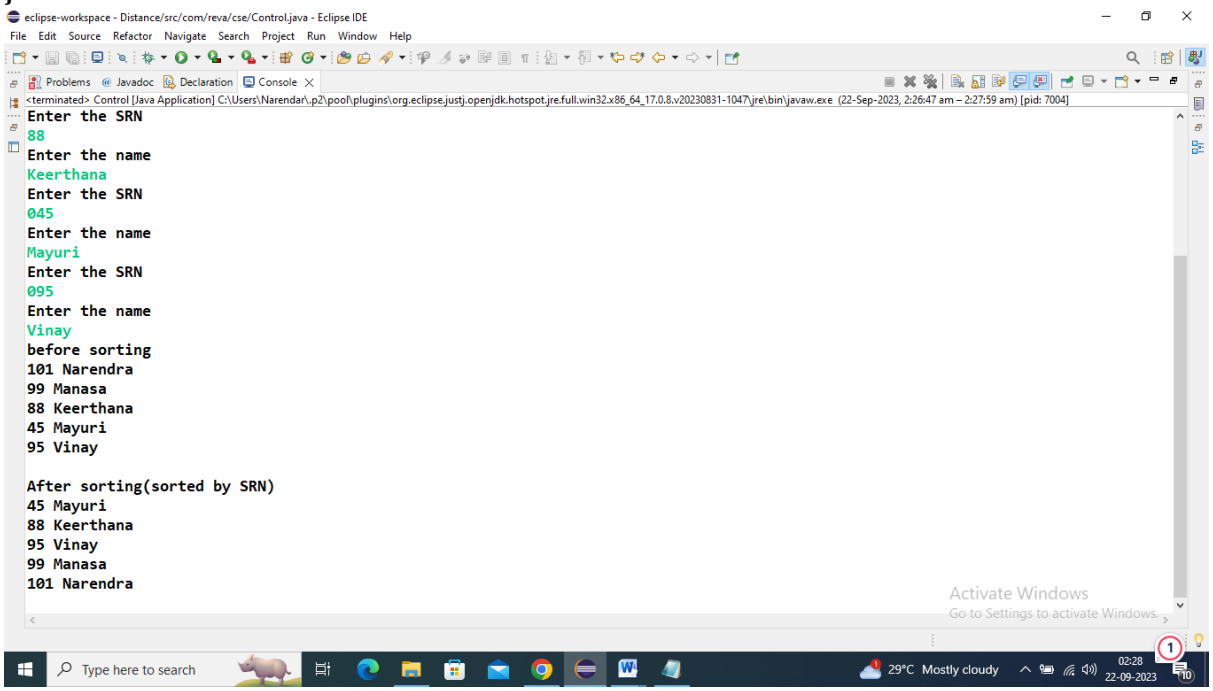
11) Write a Java program to Demonstrate Comparator interface and Array list to sort student's information by considering student names and students roll number.

Note: Sorting should be done according srn and student name

```
package com.reva.cse;
import java.util.*;
// create the Student class
class Student1
{
   int srn;
   String name;
   // constructor
   Student1(int srn, String name)
   {
      this.srn = srn;
      this.name = name;
```

```
    }
}

// creates the comparator for comparing srn value
class StudentComparator implements Comparator<Student1>
{
    // override the compare() method
    public int compare(Student1 s1, Student1 s2)
    {
        if (s1.srn == s2.srn)
            return 0;
        else if (s1.srn > s2.srn)
            return 1;
        else
            return -1;
    }
}

public class Control
{
    public static void main(String[] args)
    {
        // create the ArrayList object
            int n;
            System.out.println("Enter the Number student details you want to sort");
            Scanner sc=new Scanner(System.in);
            n=sc.nextInt();
            ArrayList<Student1> s = new ArrayList<Student1>();
            for(int i=0;i<n;i++)
            {
                        System.out.println("Enter the SRN");
                        int srn=sc.nextInt();
                        System.out.println("Enter the name");
                        String name=sc.next();
          s.add(new Student1(srn, name));
            }
        System.out.println("before sorting");
        for (Student1 st : s)
        {
            System.out.println(st.srn + " " + st.name
                        + " " );
        }
        System.out.println();

        System.out.println(
            "After sorting(sorted by SRN)");

        // call the sort function
        Collections.sort(s, new StudentComparator());
        for (Student1 st : s)
        {
            System.out.println(st.srn + " " + st.name
                        + " " );
        }
    }
```

}



12) Write a java program for producer and consumer problem.

```
package com.reva.cse;
class Thread1
{
   int num;
   boolean vs=false;
   synchronized int pop()
   {
      if (!vs)
         try
         {
            wait();
         }
         catch (Exception e)
         {
            System.out.println("Excepton occurs at : "+e);
         }
      System.out.println("pop" +num);
      try
      {
         Thread.sleep(1000);
      }
      catch (Exception e)
      {
         System.out.println("Excepton occurs at : "+e);
      }
      vs=false;
      notify();
      return num;
   }
   synchronized int push(int num)
   {
      if (vs)
         try
         {
```

```
            wait();
          }
          catch (Exception e)
          {
            System.out.println("Excepton occur at : "+e);
          }
        this.num=num;
        vs=true;
        System.out.println("push"+num);
        try
        {
          Thread.sleep(1000);
        }
        catch (Exception e)
        {
          System.out.println("Excepton occur at : "+e);
        }
        notify();
        return num;
    }
}


class Producer implements Runnable
{
    Thread1 t;
    Producer(Thread1 t)
    {
        this.t=t;
        new Thread(this,"Producer").start();
    }
    public void run()
    {
        int x=0;
        int i = 0;
        while (x<10)
        {
            i=i+5;
          t.push(i);
          x++;
        }
    }
}


class Consumer implements Runnable
{
    Thread1 t;
    Consumer(Thread1 t)
    {
        this.t=t;
        new Thread(this,"Consumer").start();
    }
    public void run()
    {
        int x=0;
        while (x<10)
        {
          t.pop();
          x++;
        }
    }
```

```
}


public class  ProducerConsumer
{
   public static void main(String[] args)
   {
      Thread1 t=new Thread1();
      new Producer(t);
      new Consumer(t);
   }
}
```

Output: