NITTE
EDUCATION TRUST

N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC

Department of Computer science and Engineering

Laboratory Manual

Subject Name: PROGRAMMING IN JAVA

Subject Code: 18CS505

Subject Name: Programming in Java

Subject code: 18CS505

Semester: V

Software tools used
1. JDK 1.7
2. ECLIPSE IDE

Marks distribution

CIE**: Total Marks: 50**

Continuous Evaluation	**30**
MSE – Execution and writeup (Part A + Part B)	**20**

SEE: **Total Marks: 50**

Execution (Part A + Part B)	10+20
Write up	5+5
Viva	10
Change of one question	-5

Prepared by:

Faculty Name:

Designation:

List of Experiments

PART A		
SL. NO	PROGRAMS	PAGE NO
1.	Write java program to design a class called Bank Account. Include following data members like name of the depositor, account number and balance. Initialize the data members using constructor **(read the input from keyboard).** Use following methods. a) Deposit an amount b) to withdraw an amount c) to display name and balance.	
2.	Create an abstract class Figure with abstract method area and two integer dimensions. Extend this class to inherit three more classes Rectangle, Triangle and Square which implements the area method. Write a java program to show how the area can be computed dynamically during run time for Rectangle, Square and Triangle. **(Use constructor to initialize data members).**	
3.	Design a class called Student having data members like **student name and roll number** and method called **show** to display the name and roll number. Derive a class called **Test** from student class having **three subjects' marks as a data members** and method named **show_marks** to display the marks. Create an interface called **Sports** contains constant **sports marks** and method **show_sportswt**. Derive a class **Result from Test class** and implement the **interface Sports** .Calculate total marks by considering sports marks. Then display the total in Result class.	
4.	Create a class as Student. Write a program in Java to display the names and roll numbers of students. Create an array of 10 students and initialize the array with user input. Handle ArrayIndexOutOfBoundsExeption, so that any such problem doesn't cause illegal termination of program. Read a character from user and display the student names starting with given character.	
5.	Write a java program that creates multiple threads. Main thread generates a random integer. First child thread computes the square of next three the numbers and there should be idle time of one second between displaying of each computed number. Second child thread computes the cube of next three the numbers and there should be idle time of two seconds between displaying of each computed number.	

6.	Develop a set of methods, which work with an integer array. The methods to be implemented are:- **(i)** min (which finds the minimum element in the array) **(ii)**scale.(Multiply the array elements by suitable scale factor) Place this in a package called p1. Write a main method in a separate file (Driver Class) to use the methods of package p1.	
7.	Write a Swing program to create frame window and three text fields to input three floating point numbers from the user and displays sum, average and largest of these three numbers in a dialog box.	
8.	Write a program to copy the contents of two files in to one file.	
9.	Write a multithread java program to display the message [Welcome] [To][Java] by three separate child threads using the concept of synchronization.	
10	Design a form using html which contains two textboxes namely First name and Last name and a submit button. After entering values for textboxes and clicking Submit button it should display first name and last name as a server response. Write a JSP code for this scenario using GET or POST method.	

PART B		
SL.NO	PROGRAMS	PAGE NO
11.	Create a Swing that is having a List View to select the city, Text box to enter the name, phone number. Radio button to choose the gender and check box to choose hobbies. Create an alert to display the selections made by the user.	
12.	Write a Java program to perform the following operations in a Random Access File (assume three fields). i) Insert a record ii) Display a record	
13.	Write a JDBC program using **sql queries** to perform the following operations on a table Accounts. (Accounts Table consists of three fields namely Account Number, Name and Balance). i) Insert a record ii) Delete a record iii) Display all records	
14.	Write a JDBC program using **Java methods** to perform the following operations on a table Called Student. (Student Table consists of two fields namely Roll number and Name of the Student). i) Insert a record ii) Update a Record iii) Search the name of a given USN.	
15.	Create a socket program using TCP sockets to send message between client and server.	
16.	Create a socket program using UDP sockets to send message between client and server.	

Part -A

1. Write java program to design a class called Bank Account. Include following data members like name of the depositor, account number and balance. Initialize the data members using constructor **(read the input from keyboard).** Use following methods.
   a) Deposit an amount b) to withdraw an amount c) to display name and balance.

```
import java.util.Scanner;
class BankAcc
{
   Scanner s = new Scanner(System.in);
   String name;
   int acc_no,bal;
   BankAcc(String ip,int acc,int bal)
   {
      name = ip;
      acc_no = acc;
      this.bal = bal;
   }
   void show()
   {
      System.out.println("Name: "+name + "\nBalance: "+bal+"\n");
   }
   void deposit()
   {
      System.out.println("Enter the amount to be deposited: ");
      int amt = s.nextInt();
      bal+=amt;
      System.out.println("Current balance: "+ bal);
   }
   void draw()
   {
      System.out.println("Enter the amount to be withdrawn: ");
      int amt = s.nextInt();
      if(amt>bal)
         System.out.println("Insufficient funds\n");
      else
      {
         bal-=amt;
         System.out.println("Current balance: "+bal);
      }
   }
}
public class demo
{
   public static void main(String[] args) {
      Scanner s = new Scanner(System.in);
      System.out.println("Enter the name of the account holder: ");
      String ip = s.nextLine();
      System.out.println("Enter the account number: ");
```

```
        int acc = s.nextInt();
        System.out.println("Enter the balance: ");
        int bal = s.nextInt();
        BankAcc account = new BankAcc(ip,acc,bal);
        boolean flag = true;
        while(flag)
        {
            System.out.println("Enter your choice:\n1.Deposit\n2.Withdraw\n3.Display details\n>>");
            int ch = s.nextInt();
            switch (ch) {
                case 1: account.deposit();
                        break;
                case 2: account.draw();
                           break;
                  case 3: account.show();
                          break;
                default: flag = false;
            }
        }
    }
}
```

2. Create an abstract class Figure with abstract method area and two integer dimensions. Extend this class to inherit three more classes Rectangle, Triangle and Square which implements the area method. Write a java program to show how the area can be computed dynamically during run time for Rectangle, Square and Triangle. **(Use constructor to initialize data members).**

```
abstract class figure
{
    abstract int area();
    int dim1,dim2;
}
class rectangle extends figure
{
    rectangle(int l,int b)
    {
        dim1 = l;
        dim2 = b;
    }
    int area() {
        return dim1*dim2;
    }
}
class triangle extends figure
{
    triangle(int b,int h)
    {
        dim1 = b;
        dim2 = h;
    }
    int area()
```

```
    {
        return (int) (0.5*dim1*dim2);
    }
}
class square extends figure
{
    square(int s)
    {
        dim1 = s;
    }
    int area()
    {
        return dim1*dim1;
    }
}
public class abs {
    public static void main(String[] args) {
        figure f;
        rectangle r = new rectangle(1,2);
        triangle t = new triangle(3,4);
        square s = new square(2);
        f = r;
        System.out.println("Area of rectangle: " + f.area());
        f = t;
        System.out.println("Area of triangle: "+f.area());
        f = s;
        System.out.println("Area of square: "+f.area());
    }
}
```

3. Design a class called Student having data members like **student name and roll number** and method called **show** to display the name and roll number. Derive a class called **Test** from student class having **three subjects' marks as a data members** and method named **show_marks** to display the marks. Create an interface called **Sports** contains constant **sports marks** and method **show_sportswt**. Derive a class **Result from Test class** and implement the **interface Sports** .Calculate total marks by considering sports marks. Then display the total in Result class.

```
import java.util.Scanner;

class student{
    String name;
    int rollno;
    student(String name,int rno)
    {
        this.name = name;
        rollno = rno;
    }
    void show()
    {
        System.out.println("Name: "+name+"\nRoll Number: "+rollno);
    }
```

```
}
class test extends student
{
   int sub1,sub2,sub3;
   test(int sub1,int sub2,int sub3,String name,int rno)
   {
      super(name,rno);
      this.sub1 = sub1;
      this.sub2 = sub2;
      this.sub3 = sub3;
   }
   void show_marks()
   {
      System.out.println("Subject 1: "+sub1+"\nSubject 2: "+sub2+"\nSubject 3: "+sub3);
   }
}
interface sports
{
   final int sport_mark = 40;
   void show_sportswt();
}
class result extends test implements sports
{
   int total = sub1+sub2+sub3+sport_mark;
   public void show_sportswt() {
      System.out.println("Sports marks: "+sport_mark);
   }
   result(String name,int rollno,int sub1,int sub2,int sub3)
   {
      super(sub1,sub2,sub3,name,rollno);
   }
   void display()
   {
      System.out.println("Total Marks: "+total);
   }
}
public class synch {
   public static void main(String[] args) {
     Scanner s = new Scanner(System.in);
    String name = s.nextLine();
     int rno = s.nextInt();
   int sub1 = s.nextInt();
   int sub2 = s.nextInt();
     int sub3 = s.nextInt();
   result res = new result(name,rno,sub1,sub2,sub3);
   res.display();
    }
}
```

4. Create a class as Student. Write a program in Java to display the names and roll numbers of

students. Create an array of 10 students and initialize the array with user input. Handle ArrayIndexOutOfBoundsExeption, so that any such problem doesn’t cause illegal termination of program. Read a character from user and display the student names starting with given character.

```
import java.util.Scanner;

class Student {
   String name;
   int rno;

   Student(String name, int rno) {
      this.name = name;
      this.rno = rno;
   }

   void display() {
      System.out.println("Name: " + name + "\nRoll Number: " + rno);
   }
}

public class studexception {
   public static void main(String[] args) {
      String name;
      int rno;
      Scanner s = new Scanner(System.in);
      Scanner s2 = new Scanner(System.in);
      Student st[] = new Student[5];
      try {
         for (int i = 0; i < 5; i++) {
            System.out.println("Enter the name: ");
            name = s.nextLine();
            System.out.println("Enter the roll number: ");
            rno = s2.nextInt();
            st[i] = new Student(name, rno);
         }
      } catch (ArrayIndexOutOfBoundsException e) {
         e.printStackTrace();
      }
      System.out.println("Enter the first character: ");
      char check = s.next().charAt(0);
      try {
         for (int i = 0; i < 5; i++) {
            if (check == st[i].name.charAt(0)) {
               st[i].display();
            }
         }
      } catch (ArrayIndexOutOfBoundsException e) {
         e.printStackTrace();
      }
   }
```

```
    }
```

5. Write a java program that creates multiple threads. Main thread generates a random integer. First child thread computes the square of next three the numbers and there should be idle time of one second between displaying of each computed number. Second child thread computes the cube of next three the numbers and there should be idle time of two seconds between displaying of each computed number.

```
import java.util.Random;

class T1 extends Thread{
   int val;
   T1(int n){
      val = n;
   }

   public void run() {
      try {
         for (int i = 1; i <= 3; i++) {
            System.out.println((val + i) + ": " + Math.pow((val + i), 2));
            Thread.sleep(1000);
         }
      }catch (InterruptedException e){
         e.printStackTrace();
      }
   }
}
class T2 extends Thread{
   int val;
   T2(int n){
      val = n;
   }

   public void run() {
      try {
         for(int i=1;i<=3;i++){
            System.out.println((val+i)+": "+Math.pow((val+i),3));
            Thread.sleep(2000);
         }
      }catch (InterruptedException e){
         e.printStackTrace();
      }
   }
}
public class multithread {
   public static void main(String[] args) {
      Random rand = new Random();
      int x = rand.nextInt(11);
      System.out.println(x+"\n");
      T1 thread1 = new T1(x);
      thread1.start();
```

```
        T2 thread2 = new T2(x);
        thread2.start();
    }
}
```

6. Develop a set of methods, which work with an integer array. The methods to be implemented are:-

(i) min (which finds the minimum element in the array)

(ii)scale.(Multiply the array elements by suitable scale factor)

Place this in a package called p1. Write a main method in a separate file (Driver Class) to use the methods of package p1

```
package p1;

public class intarr {
    public int min(int[] arr){
        int temp = arr[0];
        for(int i=1;i<arr.length;i++)
        {
            if(arr[i]<temp)
                temp = arr[i];
        }
        return temp;
    }
    public void scale(int[] arr,int n){
        for(int i=0;i<arr.length;i++)
            arr[i]*=n;
    }
}
```

Driver

```
import p1.intarr;

import java.util.Scanner;

public class driver {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        int n = s.nextInt();
        int[] arr = new int[n];
        for(int i=0;i<n;i++)
        {
            int val = s.nextInt();
            arr[i] = val;
        }
        intarr obj = new intarr();
        System.out.println(obj.min(arr));
        int val = s.nextInt();
        obj.scale(val,arr);
```

```
        for(int i:arr){
            System.out.println(i+"\t");
        }
    }
}
```

7. Write a Swing program to create frame window and three text fields to input three floating point numbers from the user and displays sum, average and largest of these three numbers in a dialog box.

```
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
class fswing implements ActionListener {
    JTextField tf1,tf2,tf3;
    JButton jb1,jb2,jb3;
    JLabel jl;
    fswing(){
        JFrame jf = new JFrame("Compute");
        jf.setSize(300,300);
        jf.setLayout(new FlowLayout());

        tf1 = new JTextField(20);
        tf2 = new JTextField(20);
        tf3 = new JTextField(20);
        jf.add(tf1);
        jf.add(tf2);
        jf.add(tf3);
        jb1 = new JButton("Sum");
        jb1.addActionListener(this);
        jb2 = new JButton("Avg");
        jb2.addActionListener(this);
        jb3 = new JButton("Largest");
        jb3.addActionListener(this);
        jf.add(jb1);
        jf.add(jb2);
        jf.add(jb3);
        jl = new JLabel("Select your choice!");
        jf.add(jl);
        jf.setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        double a = Double.parseDouble(tf1.getText());
        double b = Double.parseDouble(tf2.getText());
        double c = Double.parseDouble(tf3.getText());
        double largest;
        if(e.getActionCommand()=="Sum"){
            JOptionPane.showMessageDialog(null,a+b+c);
        }
```

```
        else if(e.getActionCommand()=="Avg"){
            JOptionPane.showMessageDialog(null,(a+b+c)/3);
        }
        else{
            if(a>b && a>c)
                largest = a;
            else if(b>a && b>c)
                largest = b;
            else
                largest = c;
            JOptionPane.showMessageDialog(null,largest);
        }
    }
}
public class sumavglarge {
    public static void main(String[] args) {
        new fswing();
    }
}
```

8. Write a program to copy the contents of two files in to one file.

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.PrintWriter;
import java.util.Scanner;

public class filecpy {
    public static void main(String[] args) throws FileNotFoundException {
        PrintWriter pw = new PrintWriter("f3.txt");
        File f1 = new File("f1.txt");
        File f2 = new File("f2.txt");
        Scanner s1 = new Scanner(f1);
        Scanner s2 = new Scanner(f2);
        while(s1.hasNext()){
            String temp = s1.nextLine();
            pw.println(temp);
        }
        while(s2.hasNext()){
            String temp = s2.nextLine();
            pw.println(temp);
        }
        pw.flush();
        pw.close();
    }
}
```

9. Write a multithread java program to display the message [Welcome] [To][Java] by three separate child threads using the concept of synchronization.

```
class First extends Thread{
```

```
    public void display(String msg){
        System.out.print("["+msg);
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.out.println("]");
    }
}
class Second extends Thread{
    First fobj;
    String msg;
    Second(First fp,String msg){
        fobj = fp;
        this.msg = msg;
        start();
    }

    public void run() {
        synchronized (fobj) {
            fobj.display(msg);
        }
    }
}
public class multithr {
    public static void main(String[] args) {
        First fnew = new First();
        Second s1 = new Second(fnew,"welcome");
        Second s2 = new Second(fnew,"to");
        Second s3 = new Second(fnew,"java");
    }
}
```

10. Design a form using html which contains two textboxes namely First name and Last name and a submit button. After entering values for textboxes and clicking Submit button it should display first name and last name as a server response. Write a JSP code for this scenario using GET or POST method.

Index.html

```
<!DOCTYPE html>
<html>
<title>Sample Example </title>
<body>
<h1> <center> Example of JSP </center> </h1>
<b> Mathematics</b>
<hr>
<form method="post" action="add.jsp">
<form>
    <label for="firstname">First Name: </label> <br/>
```

```
        <input type="text" id="firstname" name="firstname"/> <br/>
  <label for="lastname">Last Name: </label>
        <input type="text" id="lastname" name="lastname"/> <br/>
        <input type="submit" value="submit">
 </form>


</form>
</body>
</html>
```

add.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
   pageEncoding="UTF-8"%>
<html>
  <head>
    <title>Using GET Method to Read Form Data</title>
  </head>

  <body>
    <h1>Using GET Method to Read Form Data</h1>
    <ul>
      <li><p><b>First Name:</b>
        <%= request.getParameter("firstname")%>
      </p></li>
      <li><p><b>Last  Name:</b>
        <%= request.getParameter("lastname")%>
      </p></li>
    </ul>

  </body>
</html>
```

PART-B

11. Create a Swing that is having a List View to select the city, Text box to enter the name, phone number. Radio button to choose the gender and check box to choose hobbies. Create an alert to display the selections made by the user.

```
import javax.swing.*;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
```

```
import java.awt.event.ItemListener;
import java.util.ArrayList;

class SwingEle implements ListSelectionListener, ActionListener, ItemListener {
    JScrollPane jsp;
    JCheckBox c1,c2,c3,c4;
    String[] cities = {"Mlore","Blore","Mysuru","Madikeri","Udupi","Bhatkal","Puttur","Sullia"};
    JList<String> jList = new JList<String>(cities);
    JRadioButton r1,r2,r3;
    JTextField tf1,tf2;
    ArrayList<String> hobby = new ArrayList<String>();
    SwingEle(){
        JFrame jf = new JFrame("Big Swing");
        jf.setSize(300,300);
        jf.setLayout(new FlowLayout());
        jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        JLabel jl1 = new JLabel("Name");
        tf1 = new JTextField(20);
        JLabel jl2 = new JLabel("Number");
        tf2 = new JTextField(20);
        jf.add(jl1);
        jf.add(tf1);
        jf.add(jl2);
        jf.add(tf2);

        jList.addListSelectionListener(this);
        jsp = new JScrollPane(jList);
        jsp.setPreferredSize(new Dimension(100,100));
        jf.add(jsp);

        r1 = new JRadioButton("Male");
        r1.addActionListener(this);
        r2 = new JRadioButton("Female");
        r2.addActionListener(this);
        r3 = new JRadioButton("Other");
        r3.addActionListener(this);
        ButtonGroup bg = new ButtonGroup();
        bg.add(r1);
        bg.add(r2);
        bg.add(r3);
        jf.add(r1);
        jf.add(r2);
        jf.add(r3);

        c1 = new JCheckBox("Swimming");
        c1.addItemListener(this);
        c2 = new JCheckBox("Music");
        c2.addItemListener(this);
        c3 = new JCheckBox("Movies");
        c3.addItemListener(this);
        c4 = new JCheckBox("Cricket");
```

```
    c4.addItemListener(this);
    jf.add(c1);
    jf.add(c2);
    jf.add(c3);
    jf.add(c4);

    JButton sub = new JButton("Submit");
    sub.addActionListener(this);
    jf.add(sub);
    jf.setVisible(true);

}

@Override
public void valueChanged(ListSelectionEvent e) {

}

@Override
public void actionPerformed(ActionEvent e) {

    String val = e.getActionCommand();
    String city,name,num,gender;
    if(val.equals("Submit")){
        city = jList.getSelectedValue();
        name = tf1.getText();
        num = tf2.getText();
        if(r1.isSelected())
            gender = r1.getActionCommand();
        else if(r2.isSelected())
            gender = r2.getActionCommand();
        else
            gender = r3.getActionCommand();
        if(c1.isSelected())
            hobby.add(c1.getActionCommand());
        if(c2.isSelected())
            hobby.add(c2.getActionCommand());
        if(c3.isSelected())
            hobby.add(c3.getActionCommand());
        if(c4.isSelected())
            hobby.add(c4.getActionCommand());
        String res = "Name: "+name+"\nNumber: "+num+"\nGender: "+gender+"\nHobbies: \n";
        for (String s : hobby) res += s + "\n";
        JOptionPane.showMessageDialog(null,res);
        hobby.clear();
    }
}

@Override
public void itemStateChanged(ItemEvent e) {
```

```
    }
}

public class bigswing {
    public static void main(String[] args) {
        new SwingEle();
    }
}
```

12. Write a Java program to perform the following operations in a Random Access File (assume three fields).

i) Insert a record ii) Display a record

```
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;

public class randfile {
    private static final int REC_SIZE = 42;
    private static final int NAME_SIZE = 15;
    private static RandomAccessFile ranFile;

    public static void main(String[] args) throws IOException {
        ranFile = new RandomAccessFile("acc.dat","rw");
        writeRecord(1000,"Demo",1000);
        showRecords();
    }
    public static void writeRecord(long accno,String name,float bal) throws IOException {
        long pos = ranFile.length();
        ranFile.seek(pos);
        ranFile.writeLong(accno);
        writeString(name,NAME_SIZE);
        ranFile.writeFloat(bal);
    }
    public static void writeString(String name,int size) throws IOException {
        int len = name.length();
        if(len<size){
            ranFile.writeChars(name);
            for(int i=len;i<size;i++)
                ranFile.writeChar(' ');
        }else{
            ranFile.writeChars(name.substring(0,size));
        }
    }
    public static void showRecords() throws IOException {
        long n = ranFile.length()/REC_SIZE;
        ranFile.seek(0);
        for(int i=0;i<n;i++){
            long accno = ranFile.readLong();
            String name = readString(NAME_SIZE);
            float bal = ranFile.readFloat();
```

```
            System.out.println(accno+" "+name+" "+bal);
        }
    }
    public static String readString(int size) throws IOException {
        String res = "";
        for(int i=0;i<size;i++)
            res+=ranFile.readChar();
        return res;
    }
}
```

13. Write a JDBC program using **sql queries** to perform the following operations on a table Accounts. (Accounts Table consists of three fields namely Account Number, Name and Balance).
i) Insert a record ii) Delete a record iii) Display all records

```
import java.util.*;
import java.sql.*;
class Dbdemo
{
static Connection link;
static Statement stm;
static ResultSet rs;
Dbdemo() throws SQLException
{
try
{
Class.forName("com.mysql.jdbc.Driver");
link = DriverManager.getConnection("jdbc:mysql://172.16.2.3/student","student", "student");
}
catch(ClassNotFoundException cnfEx)
{
System.out.println("* Unable to load driver! *");
System.exit(1);
}
catch(SQLException sqlEx)
{
System.out.println("* Cannot connect to database! *");
System.exit(1);
}
stm = link.createStatement ( );
}
void displayinfo() throws SQLException
{
String s = "SELECT * FROM Account85";
rs = stm.executeQuery (s);
while(rs.next())
{
System.out.println(rs.getInt(1) +"\t"+ rs.getString(2)+"\t" + rs.getString(3)+"\t" + rs.getString(4));
}
```

```
}
void insertinfo() throws SQLException
{
int r1=0;
Scanner s=new Scanner(System.in);
System.out.println("Enter accNum");
int accno=s.nextInt();
System.out.println("Enter surname");
String sur=s.next();
System.out.println("Enter firstname");
String fname=s.next();
System.out.println("Enter balance");
double bal=s.nextFloat();
try{
String s1 = "insert into Account85 values("+accno+",'"+sur+"','"+fname+"',"+bal+")";
r1 = stm.executeUpdate (s1);
}
catch(Exception e)
{
System.out.println(e);
}
System.out.println(r1 +"rows affected");
}
void deleteinfo() throws SQLException
{
Scanner s1=new Scanner(System.in);
System.out.println("Enter the account no.");
int acc = s1.nextInt();
String s2 = "DELETE FROM Account85 WHERE accNum="+acc ;
stm.executeUpdate (s2);
System.out.println("Database deleted successfully!!!!");
}
void updateinfo() throws SQLException
{
Scanner s1=new Scanner(System.in);
System.out.println("Enter the account no.");
int acc = s1.nextInt();
System.out.println("enter the new name");
String name=s1.next();
String s3 = "update Account85 set firstNames='" +name+"' where accNum="+acc;
stm.executeUpdate (s3);
System.out.println("Database updated successfully!!!!");
}
void closecon() throws SQLException
{
rs.close();
stm.close();
link.close();
}
}
public class JDBCdemo {
```

```
public static void main(String[] args) throws SQLException {
// TODO Auto-generated method stub
boolean f=true;
Dbdemo d1 = new Dbdemo();
d1.displayinfo();
while(f)
{
System.out.println("1:insert 2:delete 3:display 4:update 5:exit");
System.out.println("enter your option");
Scanner s=new Scanner(System.in);
int op=s.nextInt();
switch(op)
{
case 1:d1.insertinfo();
break;
case 2:d1.deleteinfo();
break;
case 3:d1.displayinfo();
break;
case 4:d1.updateinfo();
break;
case 5:f=false;
}
}
d1.closecon();
}
}
```

14. Write a JDBC program using **Java methods** to perform the following operations on a table Called Student. (Student Table consists of two fields namely Roll number and Name of the Student).

i) Insert a record ii) Update a Record iii) Search the name of a given USN.

```
import java.util.*;
import java.sql.*;
class Sample
{
static Connection con;
 static Statement stm;
 static ResultSet rs;
 Sample() throws SQLException
{
 try
 {

 Class.forName("com.mysql.jdbc.Driver"); // step 1
con = DriverManager.getConnection("jdbc:mysql://172.16.2.3/student","student",
"student"); // step 2

 }
```

```
catch(ClassNotFoundException cnfEx)
{
System.out.println("* Unable to load driver! *");
System.exit(1);
}
catch(SQLException sqlEx)
{
System.out.println("* Cannot connect to database! *");
System.exit(1);
}
stm=con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE); // step 3
rs=stm.executeQuery("select *from st");
}
 public void insert() throws SQLException
 {
 Scanner inp=new Scanner(System.in);

 System.out.println("Enter the usn");
 int usn;
 usn=inp.nextInt();
 System.out.println("Enter the fname");
 String fname=inp.next();

 rs.moveToInsertRow();

 rs.updateInt("rollno",usn);
 rs.updateString("name",fname);
 rs.insertRow();
 }

 void delete(int r)throws SQLException
 {

 rs.absolute(r);
 rs.deleteRow();
 }

 void update(int r) throws SQLException
 {
 rs.absolute(r);
 Scanner s1= new Scanner(System.in);
 String fname= s1.nextLine();
 rs.updateString("name", fname);
 rs.updateRow();
 }


 void search (int r)throws SQLException
{
 String s = "SELECT * FROM st";
```

```
 rs = stm.executeQuery (s);
 int pos=0;
 while(rs.next())
 {
 if(r==rs.getInt(1))
 {
 pos=1;
 break;
 }
 }

 if(pos==1)
 {
 System.out.println("search is successful");
 }

 else
 {
 System.out.println("Record not found");
 }

 }

}
public class J2M {
public static void main(String[] args) throws SQLException {
// TODO Auto-generated method stub
Sample obj=new Sample();
boolean flag=true;
while(true)
{
System.out.println("1.Insert 2.Delete 3.Update 4.Search");
Scanner s=new Scanner(System.in);
int choice=s.nextInt();
switch(choice)
{
case 1:obj.insert();
break;
case 2:
 System.out.println("enter the row to be deleted");
 int r=s.nextInt();
obj.delete(r);
 break;

case 3: System.out.println("enter the row to be updated");
 int r1=s.nextInt();
 obj.update(r1);
 break;

case 4:
System.out.println("enter the usn to be searched");
```

```
int r2=s.nextInt();
 obj.search(r2);
 break;
default:
System.exit(0);
break;
}
}
}
}
```

15. Create a socket program using TCP sockets to send message between client and server.

Server

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.*;
import java.util.Scanner;

public class tcpServer {
    private static ServerSocket servSock;
    private static int PORT = 1234;

    public static void main(String[] args) {
        System.out.println("Connecting to port..\n");
        try {
            servSock = new ServerSocket(PORT);
        } catch (IOException e) {
            e.printStackTrace();
            System.exit(1);
        }
        do {
            handleClient();
        }while (true);
    }
    public static void handleClient(){
        Socket link = null;
        try {
            link = servSock.accept();
            PrintWriter pw = new PrintWriter(link.getOutputStream());
            Scanner s = new Scanner(link.getInputStream());
            int num = 0;
            String message;
            message = s.nextLine();
            while (!message.equals("close")){
                num++;
                System.out.println("Received");
                pw.println("Message "+num+": "+message);
                message = s.nextLine();
            }
            pw.println(num+" messages received");
```

```
        } catch (IOException e) {
            e.printStackTrace();
        }
        finally {
            try {
                link.close();
                System.exit(1);
            } catch (IOException e) {
                e.printStackTrace();
                System.exit(1);
            }
        }
    }
}
```

Client

```
import java.io.IOException;
import java.io.PrintWriter;
import java.net.InetAddress;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.Scanner;

public class tcdClient {
    private static InetAddress host;
    private static int PORT = 1234;

    public static void main(String[] args) {
        try {
            System.out.println("Connecting...\n");
            host = InetAddress.getLocalHost();
        } catch (UnknownHostException e) {
            e.printStackTrace();
            System.exit(1);
        }
        accessServer();
    }
    public static void accessServer(){
        Socket link = null;
        try {
            link = new Socket(host.getHostName(),PORT);
            Scanner s = new Scanner(link.getInputStream());
            Scanner ip = new Scanner(System.in);
            PrintWriter pw = new PrintWriter(link.getOutputStream());
            String msg,response;
            do{
                msg = ip.nextLine();
                pw.println(msg);
                response = s.nextLine();
                System.out.println("\nSERVER>"+response);
```

```
            }while (!msg.equals("Close"));
        } catch (IOException e) {
            e.printStackTrace();
        }
        finally {
            try {
                System.out.println("Closing connection...\n");
                link.close();
                System.exit(1);
            } catch (IOException e) {
                e.printStackTrace();
                System.exit(1);
            }
        }
    }
}
```

16. Create a socket program using UDP sockets to send message between client and server.

Server

```
import java.io.IOException;
import java.net.*;

public class udpServer {
    private static DatagramSocket datagramSocket;
    private static DatagramPacket inPacket, outPacket;
    private static byte[] buffer;
    private static int port = 1234;

    public static void main(String[] args) {
        try {
            datagramSocket = new DatagramSocket(port);
        } catch (SocketException e) {
            e.printStackTrace();
        }
        handleClient();
    }

    public static void handleClient() {
        String messageIn = "", messageOut="";
        int n=0;
        buffer = new byte[256];
        try {
            do {

                inPacket = new DatagramPacket(buffer, buffer.length);
                datagramSocket.receive(inPacket);
                InetAddress host = inPacket.getAddress();
                int clientPort = inPacket.getPort();
                messageIn = new String(inPacket.getData(), 0, inPacket.getLength());
```

```
            n++;
            System.out.println("Received");
            messageOut = "Message " + n + ": " + messageIn;
            outPacket = new DatagramPacket(messageOut.getBytes(), messageOut.length(), host,
clientPort);
            datagramSocket.send(outPacket);
         } while (!messageIn.equals("close"));
      }catch (Exception e)
      {
         e.printStackTrace();
      }finally {
         datagramSocket.close();
         System.exit(1);
      }
   }
}
```

Client

```
import java.net.*;
import java.util.Scanner;

public class udpClient {
   private static InetAddress host;
   private static int port = 1234;
   private static DatagramSocket datagramSocket;
   private static DatagramPacket inPacket,outPacket;
   private static byte[] buffer;
   public static void main(String[] args) {
      System.out.println("Connecting..\n");
      try {
         host = InetAddress.getLocalHost();
         datagramSocket = new DatagramSocket();
      } catch (Exception e) {
         e.printStackTrace();
      }
      accessServer();
   }
   public static void accessServer(){
      String messageIn = "",messageOut;
      Scanner s = new Scanner(System.in);
      try{
      do {
         buffer = new byte[256];
         messageOut = s.nextLine();
         outPacket = new DatagramPacket(messageOut.getBytes(),messageOut.length(),host,port);
         datagramSocket.send(outPacket);
         inPacket = new DatagramPacket(buffer, buffer.length);
         datagramSocket.receive(inPacket);
         messageIn = new String(inPacket.getData(),0,inPacket.getLength());
         System.out.println("Server>"+messageIn);
```

```
        }while (!messageOut.equals("close"));
        }catch (Exception e)
        {
           e.printStackTrace();
        }finally {
           datagramSocket.close();
           System.exit(1);
        }
     }
}
```