ATSS's

Institute of Industrial & Computer Management & Research, Nigdi

Academic year – 2022-2023

MCA Ist Year – Ist Semester

IT11: Java Programming

Practical Journal on

IT11L- Java Programming (SEM-I)

Submitted by
Name: Shashikant S Yadav
Rollno:103

Div:CodeWarriors

Seatno:20342

Date:

Course Outcomes:

Student will be able to

CO1: Demonstrate Collection framework (Apply)

CO2: Develop GUI using AWT and swing (Apply)

CO3: Develop Web application using JSP and Servlet, JDBC (Apply)

CO4: Apply Data Structure to solve problems using JavaScript (Apply)

CO5: Demonstrate the concepts of Core Java (Apply)

ATSS's

Institute of Industrial and Computer Management and Research, Nigdi Pune

MCA Department

INDEX

Students Name: _Shashikant Subechand Yadav_____ Roll No. 103_____

Sr. No	Program Title	Course Outcome	Page No.	Teacher's Sign with Date	Remarks
1.	Print the pattern: 10 5 25 30 15 20 50 35 40 45 55 60 65	CO5			
2.	Design an interface AdvancedArithmetic which contains a method signature int divisor_sum(int n). You need to write a class called MyCalculator which implements the interface. divisor_sum(int n) function takes an integer as input and return the sum of all its divisors. Divisors of 6 are 1, 2, 3 and 6, so divisor_sum should return 12. (0<n<100)	CO5			
3.	Create an Interface 'Animals' with abstract method **'void sound()'** and default method **'void walk()'**. Implement abstract method in class 'Cat' & 'Dog'. Now create an object for each of the subclasses and call their respective methods and default method too.	CO5			
4.	Declare the integer array with 10 numbers. Generate 2 new arrays **Prime** and **NonPrime** with prime and non-prime numbers from main array.	CO5			
5.	Write an application to identify and move all 0's to the end of an array. Maintain the sequence of the other (non-zero) array elements.	CO5			
6.	Write an application which will throw OverwtProductException if Product weight is above 60kg. (Use User defined exception)	CO5			
7.	Given two arrays, 1,2,3,4,5 and 2,3,1,1,0,5,0,2,1 find which number is not present in the second array.	CO5			
8.	Write code to check whether a no is a power of two or not?	CO5			

9.	Write a code to display string in reverse order of words.	**CO5**			
10.	Write a code to accept a string and check if there are two same consecutive letters, delete one of them.	**CO5**			
11.	Write a threaded application to print in one text area 1,2,3,4.... and in other textarea 2,4,9,16	**CO2**			
12.	Write a code to create calculator application using AWT, which will calculate simple Arithmetic operations.	**CO2**			
13.	Write a Menu Driven Program for Blood Donor application for following task a. Insert blood donor details into database. b. Display blood group-wise details of donors c. Update the address of a specific donor. d. Delete the record of donors whose age is below 18.	**CO3**			
14.	Write a servlet to check username & password passed from html page. If it is “Scott” & “tiger”, display welcome message else show the same html page again. [With res.sendRedirect (“http://localhost:8080/login.html”)]	**CO3**			
15.	Write a program to draw a circle on panel and move the circle as mouse is moving.	**CO2**			
16.	Write a servlet to add a Cookie to clients machine that stores username, current date & time. Display the same.	**CO3**			
17.	Write java program to generate 10 terms of Fibonacci series using threads.	**CO2**			
18.	Create a menu driven program for Bank account(acc_no, Name, amt) (Hint: use vector) 1. Add 2. Search 3. Delete 4. Display	**CO1**			
19.	Write a program to store employee in TreeSet and make sure employees are stored in sorted order of their age.	**CO1**			
20.	Create the list of patients and display the names of patients starting with ‘A’	**CO1**			

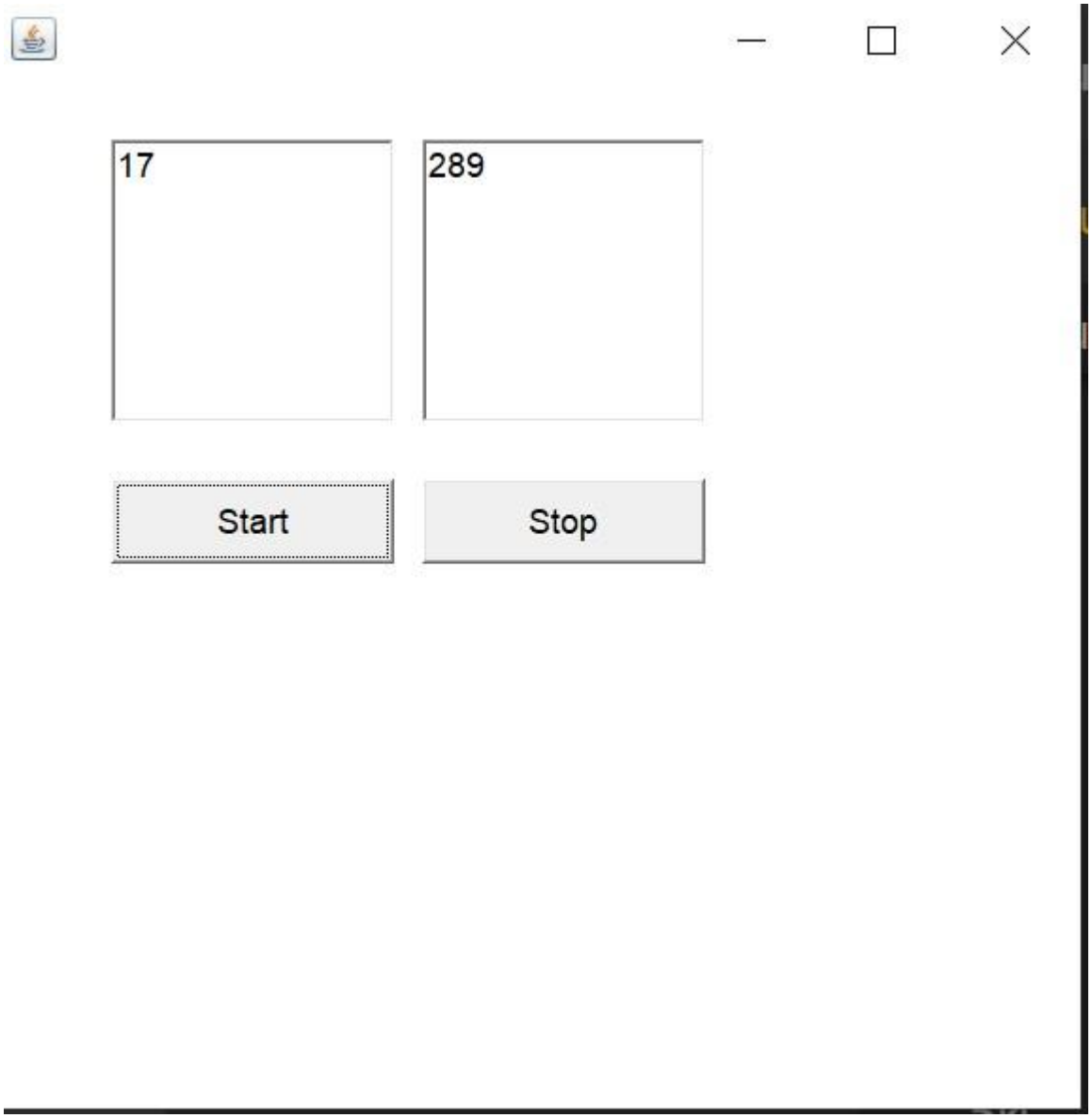
11. Write a threaded application to print in one text area 1,2,3,4.... and in other textarea 2,4,9,16 ....

Solution:-

```
import java.awt.*;
import java.awt.event.*;
class MultiThreding extends Frame implements
ActionListener,Runnable
{
Button b1, b2;
TextField txt1, txt2;
int cnt;
Thread t1 = new Thread(this, "txt1");
Thread t2 = new Thread(this, "txt12");
public MultiThreding()
{
setLayout(null);
txt1 = new TextField();
txt2 = new TextField();
b1 = new Button("Start");
b2 = new Button("Stop");
txt1.setBounds(50,50,100,100);
txt2.setBounds(160,50,100,100);
b1.setBounds(50,170,100,30);
b2.setBounds(160,170,100,30);
add(txt1);
add(txt2);
b1.addActionListener(this);
b2.addActionListener(this);
add(b1);
add(b2);
setSize(400,400);
setVisible(true);
cnt=0;
addWindowListener(new WindowAdapter()
{
public void windowClosing(WindowEvent e)
{
System.exit(0);
}
});
}
public void actionPerformed(ActionEvent ae)
{
String str;
str=ae.getActionCommand();
if (str.equals("Start"))
```

```
{
t1.start();
t2.start();
}
else if (str.equals("Stop"))
{
t1.stop();
t2.stop();
}
}
public void run()
{
try
{
for (int i=1; i<=100;i++)
{
txt1.setText(""+i);
t1.sleep(150);
txt2.setText(""+ i*i);
t2.sleep(150);
}
}
catch (Exception ex)
{
ex.printStackTrace();
}
}
public static void main(String[] args)
{
new MultiThreding().show();
}
}
```

Output:



12. Write a code to create calculator application using AWT, which will calculate simple Arithmetic operations.

Solution:-

```
import java.awt.event.*; import javax.swing.*;
import java.awt.*;

class calculator extends JFrame implements ActionListener {

   static JFrame f;

   static JTextField l;
```

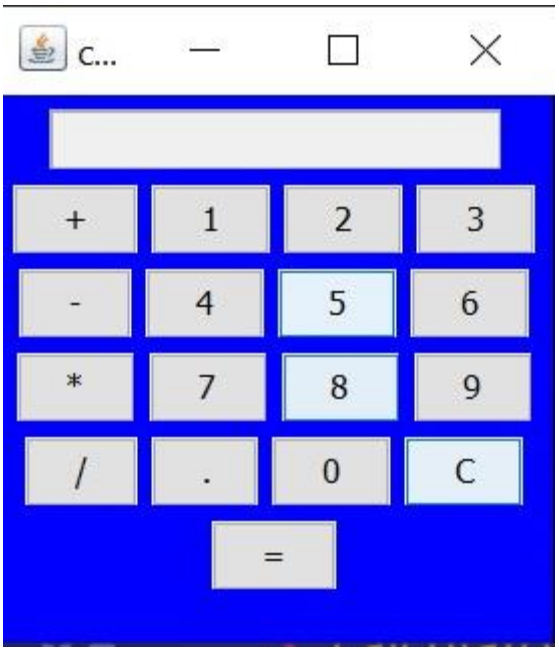
```
   String s0, s1, s2;

   calculator() {
      s0 = s1 = s2 = "";
   }

   public static void main(String args[]) {

      f = new JFrame("calculator");

      try {
         // set look and feel

UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassNa me());
         } catch (Exception e) {
         System.err.println(e.getMessage());
      }

      calculator c = new calculator();

      l = new JTextField(16);

      l.setEditable(false);

      JButton b0, b1, b2, b3, b4, b5, b6, b7, b8, b9, ba, bs, bd, bm, be, beq, beq1;

      b0 = new JButton("0");       b1 = new
JButton("1");       b2 = new JButton("2");
b3 = new JButton("3");       b4 = new
JButton("4");       b5 = new JButton("5");
b6 = new JButton("6");       b7 = new
JButton("7");       b8 = new JButton("8");
b9 = new JButton("9");


      beq1 = new JButton("=");


      ba = new JButton("+"); bs = new JButton("-
         ");
```

```
            bd = new JButton("/");
        bm = new JButton("*");

        beq = new JButton("C");


        // create . button

        be = new JButton(".");


        // create a panel

        JPanel p = new JPanel();


        // add action listeners       bm.addActionListener(c);
    bd.addActionListener(c);       bs.addActionListener(c);
    ba.addActionListener(c);       b9.addActionListener(c);
    b8.addActionListener(c);       b7.addActionListener(c);
    b6.addActionListener(c);       b5.addActionListener(c);
    b4.addActionListener(c);       b3.addActionListener(c);
    b2.addActionListener(c);       b1.addActionListener(c);
    b0.addActionListener(c);       be.addActionListener(c);
    beq.addActionListener(c);

        beq1.addActionListener(c);


           p.add(l);

           p.add(ba);

        p.add(b1);

        p.add(b2);

        p.add(b3);

        p.add(bs);

        p.add(b4);

        p.add(b5);

        p.add(b6);

        p.add(bm);

        p.add(b7);

        p.add(b8);

        p.add(b9);

        p.add(bd);

        p.add(be);

        p.add(b0);

        p.add(beq);
```

```
        p.add(beq1);

        p.setBackground(Color.blue);

        f.add(p);

        f.setSize(200, 220);
        f.show();
    }

    public void actionPerformed(ActionEvent e) {
        String s = e.getActionCommand();

                                            if ((s.charAt(0) >= '0' && s.charAt(0) <= '9') || s.charAt(0) == '.') {

            if (!s1.equals(""))                s2 =
s2 + s;            else

                s0 = s0 + s;


            l.setText(s0 + s1 + s2);
        } else if (s.charAt(0) == 'C') {

            s0 = s1 = s2 = "";

            l.setText(s0 + s1 + s2);
        } else if (s.charAt(0) == '=') {

            double te;

            if (s1.equals("+"))
                te = (Double.parseDouble(s0) + Double.parseDouble(s2));            else if (s1.equals("-"))
                te = (Double.parseDouble(s0) - Double.parseDouble(s2));            else if (s1.equals("/"))
                te = (Double.parseDouble(s0) / Double.parseDouble(s2));            else
                te = (Double.parseDouble(s0) * Double.parseDouble(s2));

            l.setText(s0 + s1 + s2 + "=" + te);

            s0 = Double.toString(te);
```

```
            s1 = s2 = "";
        } else {

            if (s1.equals("") || s2.equals(""))              s1 = s;

            else {              double te;

                if (s1.equals("+"))
                    te = (Double.parseDouble(s0) + Double.parseDouble(s2));              else if (s1.equals("-"))
                    te = (Double.parseDouble(s0) - Double.parseDouble(s2));              else if (s1.equals("/"))
                    te = (Double.parseDouble(s0) / Double.parseDouble(s2));              else
                    te = (Double.parseDouble(s0) * Double.parseDouble(s2));

                s0 = Double.toString(te);

                s1 = s;

                s2 = "";
            }

            l.setText(s0 + s1 + s2);
        }
    }
}
```

Output:-



13. Write a Menu Driven Program for Blood Donor application for following task
a. Insert blood donor details into database.
b. Display blood group-wise details of donors
c. Update the address of a specific donor.
d. Delete the record of donors whose age is below 18.

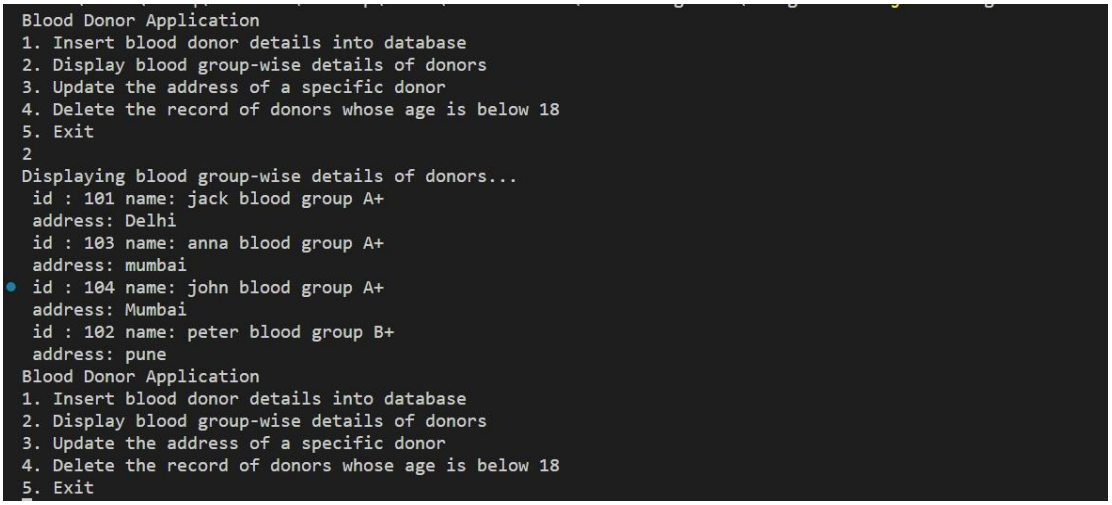
Solution:

```
import java.sql.*; import java.util.Scanner;
 public class Assignment1 {     public static void main(String[] args) {
       // create a connection to the MySQL database
          String url = "jdbc:mysql://localhost:3306/mydatabase";
          String user = "root";
          String password = "worldW@r3";
         try {
          // Driver
          Class.forName("com.mysql.cj.jdbc.Driver");
          Connection con = DriverManager.getConnection(url, user, password);

          Scanner scanner = new Scanner(System.in);
                  int choice;           do{
             // display menu and get user input
          System.out.println("Blood Donor Application");
          System.out.println("1. Insert blood donor details into database");
          System.out.println("2. Display blood group-wise details of donors");
          System.out.println("3. Update the address of a specific donor");
System.out.println("4. Delete the record of donors whose
age is below 18");
          System.out.println("5. Exit");           choice = scanner.nextInt();

          if(choice<1 || choice>5){
             System.out.println("Invalid choice");
          }
```

```
        if(choice==1){
           System.out.println("Inserting blood donor details into database...");

                // get user input for blood donor details                System.out.print("Enter donor Id: ");
int id = scanner.nextInt();
                System.out.print("Enter donor name: ");
                String name = scanner.next();
                System.out.print("Enter donor blood group:
");
                String bloodGroup = scanner.next();
                System.out.print("Enter donor address: ");
                String address = scanner.next();

                PreparedStatement ps = con.prepareStatement("insert into bloodDonor values(?, ?, ?, ?)");
                 ps.setInt(1, id);                ps.setString(2, name);                ps.setString(3,
bloodGroup);                ps.setString(4, address);

                ps.executeUpdate();
                System.out.println("Donor details have been inserted successfully");
        }
        if(choice==2){
            System.out.println("Displaying blood group-wise details of donors...");
        PreparedStatement ps =
con.prepareStatement("SELECT * FROM bloodDonor ORDER BY bloodgroup");

                ResultSet rs = ps.executeQuery();

                // display donor details
                String prevBloodGroup = "";
                 while (rs.next()) {                int id = rs.getInt(1);
                   String name = rs.getString(2);
                   String bloodGroup = rs.getString(3);
                   String address = rs.getString(4);

                   System.out.print(" id : " + id);
                   System.out.print(" name: " + name);                System.out.println(" blood group "+
bloodGroup);
                   System.out.print(" address: " + address);
                   System.out.println();

                }
        }
        if(choice==3){
           System.out.println("Updating the address of a specific donor...");

                // get user input for donor name and new address
                System.out.print("Enter donor id ");                int id = scanner.nextInt();
                System.out.print("Enter new address: ");
                String address = scanner.next();
```

```
                PreparedStatement ps = con.prepareStatement("update bloodDonor set address=? where
id=?");
                 ps.setString(1, address);                    ps.setInt(2, id);
        int rowsUpdated = ps.executeUpdate();

                if (rowsUpdated > 0) {
                   System.out.println("Address of donor has been updated successfully ");
                } else {
                   System.out.println("No such donor found in the database");
                }
        }
        if(choice==4){
                System.out.println("Deleting the record of donors whose age is below 18...");

                 PreparedStatement ps = con.prepareStatement("DELETE FROM bloodDonor WHERE age <
18");
                 int rowsDeleted = ps.executeUpdate();
                System.out.println(rowsDeleted + " donor records have been deleted successfully");
        }
         if(choice==5){
           System.out.println("Exiting the Blood Donor Application...");            return;
        }

      }while(choice!=5);

        // close the connection          con.close();
      } catch (Exception e) {
        System.err.println("Error: " + e.getMessage());
      }
   }
}
```

OUTPUT



14. Write a servlet to check username & password passed from html page. If it is “Scott” & “tiger”, display welcome message else show the same html page
again. [With res.sendRedirect (“http://localhost:8080/login.html”)]

Solution:

index.html

```
<!DOCTYPE html>
<html>
 <head>
  <meta charset="ISO-8859-1" />
  <title>Insert title here</title>
 </head>
 <body>
  <h1>Login Form</h1>

  <form method="post" action="Register">
   <label id="name">Name</label>
   <input type="text" name="name" /> <br />
   <br />
   <label id="password">Password</label>
   <input type="text" name="password" /> <br />
   <br />
   <button type="submit">Submit</button>
  </form>
 </body>
</html>
```

Register.java

```
 import java.io.*; import javax.servlet.ServletException; import javax.servlet.annotation.WebServlet; import javax.servlet.http.HttpServlet; import javax.servlet.http.HttpServletRequest; import javax.servlet.http.HttpServletResponse;

/**
*          Servlet implementation class Register
 */
@WebServlet("/Register") public class Register extends HttpServlet {  private static final long serialVersionUID = 1L;

   /**
*          @see HttpServlet#HttpServlet()
    */    public Register() {        super();
      // TODO Auto-generated constructor stub
   }

 /**
*          @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse response)
  */  protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
```

```
 // TODO Auto-generated method stub   resp.setContentType("text/html");   PrintWriter pw =
resp.getWriter();

 String username = req.getParameter("name");
 String password = req.getParameter("password");
    if(username.equals("Scott") && password.equals("tiger")) {    pw.write("Welcome "+username);
return;
 }
 resp.sendRedirect("http://localhost:8080/JavaAssignment/Exercise1.
html");
 }

}
```

Output:

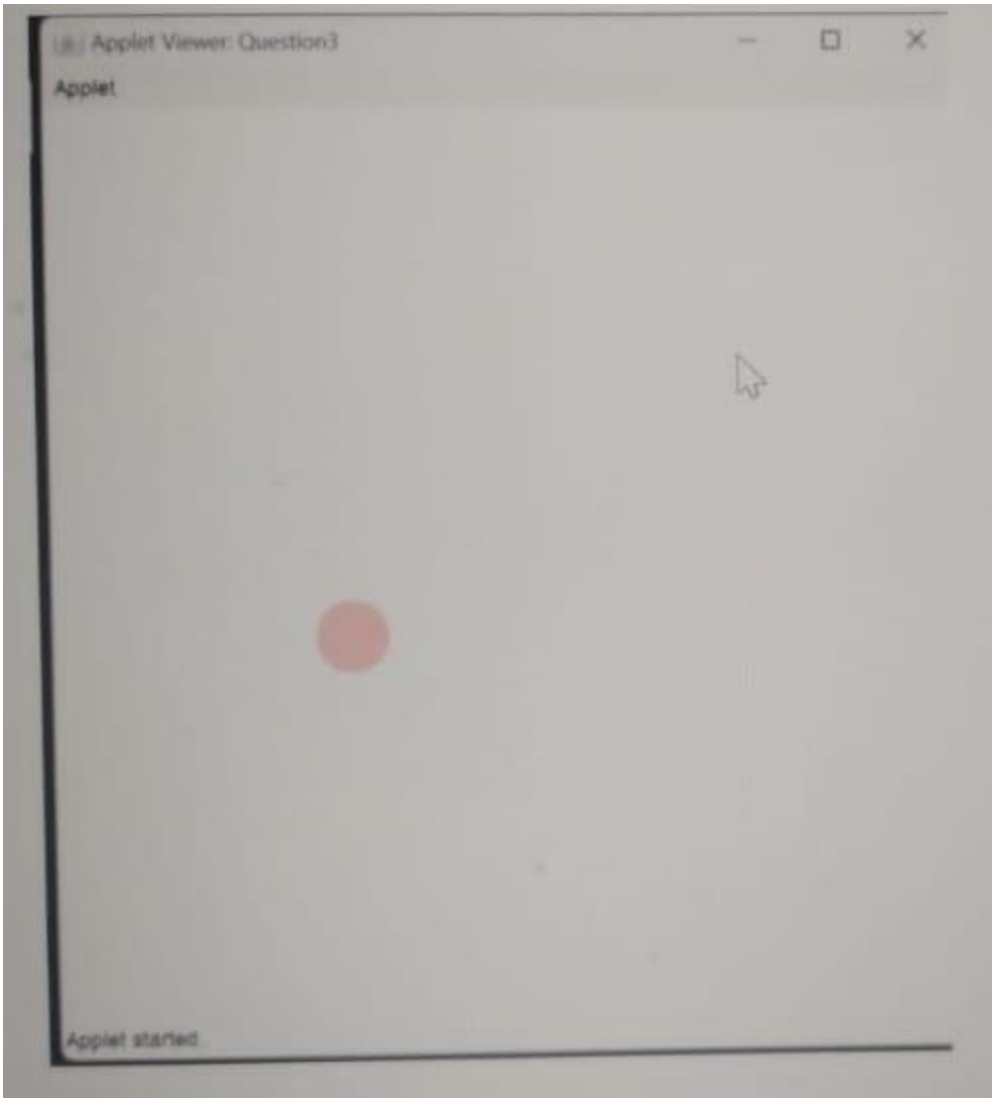
15. Write a program to draw a circle on panel and move the circle as mouse is moving.

Solution:-

```
import java.awt.*; import
java.awt.event.*; import java.applet.*;

// <applet code="Question3" height=500 width=500></applet>


public class Question3 extends Applet {   int x,y;


 public void init(){

  addMouseListener(new MyAdaptor(this));

 }


 public void drawCircle(int x, int y){

  this.x = x;     this.y = y;

  repaint();

 }


 public void paint(Graphics g){     g.setColor(Color.pink);

  g.fillOval(x, y, 40, 40);
```

```
  }

}

class MyAdaptor extends MouseAdapter{
  Question3 a1;
  MyAdaptor(Question3 a2){
    a1 = a2;
  }

  public void mouseClicked(MouseEvent e){
    a1.drawCircle(e.getX(), e.getY());
  }
}
```

Output:



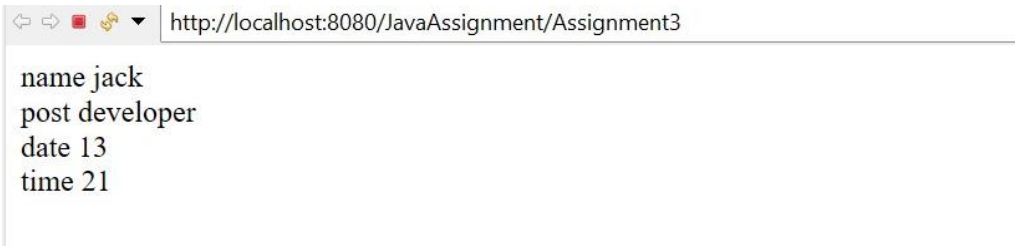
16. Write a servlet to add a Cookie to clients machine that stores username, current date & time. Display the same.

Solution:

```
import java.io.*;
 import javax.servlet.ServletException;
 import javax.servlet.annotation.WebServlet;
 import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
 import javax.servlet.http.HttpServletRequest;
 import javax.servlet.http.HttpServletResponse;



@WebServlet("/Assignment3") public class Assignment3 extends HttpServlet {  private static final long serialVersionUID = 1L;


   /**
*          @see HttpServlet#HttpServlet()
    */    public Assignment3() {        super();
      // TODO Auto-generated constructor stub
   }


 /**
*          @see HttpServlet#doGet(HttpServletRequest request,
HttpServletResponse response)
  */  protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
  // TODO Auto-generated method stub   resp.setContentType("text/html");   PrintWriter pw = resp.getWriter();
    // Date
  java.util.Date date = new java.util.Date();
  String currentDate = String.valueOf(date.getDate());
```

```
String time = String.valueOf(date.getHours());



Cookie c1 = new Cookie("name", "jack");
Cookie c2 = new Cookie("post", "developer");
Cookie c3 = new Cookie("date", currentDate);
Cookie c4 = new Cookie("time", time);
  resp.addCookie(c1);  resp.addCookie(c2);  resp.addCookie(c3);  resp.addCookie(c4);

String name, value;
Cookie c[] = req.getCookies();

for(Cookie i: c) {   name = i.getName();   value = i.getValue();
 pw.println(name+" "+value+"<br>");
}
}

}
```

OUTPUT



17. Write java program to generate 10 terms of Fibonacci series using threads.

Solution:-

```
public class Code4 { public static void main(String[] args) {
      Thread thread1 = new Thread(new Runnable(){ int n = 10;

       public void run() {
```

```
        int a = 0, b = 1, c;
        System.out.print(a + " " + b + " ");
        for (int i = 2; i < n; i++) { c = a + b;
         System.out.print(c + " ");
         a  = b;
         b  = c;
        }
       }
      });
      thread1.start();


    }
}
```

Output:-

```
0 1 1 2 3 5 8 13 21 34
```

18. Create a menu driven program for Bank account(acc_no, Name, amt) (Hint: use vector)
1. Add 2. Search 3. Delete 4. Display

Solution:-

```
import java.util.Scanner;
 import java.util.Vector;

class BankAccount { private int
    acc_no; private String name;
    private double amt;

    public BankAccount(int acc_no, String name, double amt) {
        this.acc_no = acc_no; this.name = name;
        this.amt = amt;
    }

    public int getAccNo() {
        return acc_no;
```

```
    }

    public String getName() {
        return name;
    }

    public double getAmt() {
        return amt;
    }

    public String toString() {
                                            return " account number "+acc_no+" name "+ name+" amount
"+amt;
    }
}

public class Code3 { public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Vector<BankAccount> accounts = new Vector<>(); int choice; do {
            System.out.println("Bank Account Management System");
            System.out.println("1. Add account");
            System.out.println("2. Search account");
            System.out.println("3. Delete account");
            System.out.println("4. Display accounts");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: "); choice =
            input.nextInt(); int acc_no;

            if(choice<1 || choice>5){
                System.out.println("Invalid choice");
            }

            // add if(choice==1){
                System.out.print("Enter account number: "); acc_no =
                input.nextInt(); input.nextLine();
                System.out.print("Enter name: ");
```

```
    String name = input.nextLine();
    System.out.print("Enter amount: "); double amt =
    input.nextDouble();

    accounts.add(new BankAccount(acc_no, name, amt)); System.out.println("Account added
    successfully!");

}

// search if(choice==2){

    System.out.print("Enter account number: "); acc_no =
    input.nextInt(); input.nextLine(); boolean found = false;

                    for (BankAccount account : accounts) {
                      if (account.getAccNo() == acc_no) {
                 System.out.println(account); found = true;

                break;

            }

        }

        if (!found) {

            System.out.println("Account not found");

        }

}

// delete if(choice==3){

    System.out.print("Enter account number: "); acc_no =
    input.nextInt(); input.nextLine();

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

                    if (accounts.get(i).getAccNo() == acc_no) {
            accounts.remove(i);

            System.out.println("Account deleted successfully!"); break;

        }

    }

}


// display if(choice==4){

    if (accounts.isEmpty()) {

        System.out.println("No accounts to display");

    } else { for (BankAccount account : accounts) {
        System.out.println(account);
        }

    }
```

```
            }


             if(choice==5){
                System.out.println("Exit"); return;
            }
                        } while (choice != 5);
    }
}
```

Output:-

19. Write a program to store employee in TreeSet and make sure employees are stored in sorted order of their age.

Solution:-

```
import java.util.*; public class Code1 {

public static void main(String[] args) {

  TreeSet<Employee> employees = new TreeSet<>(); employees.add(new
  Employee("Pravin", 30)); employees.add(new Employee("Ram", 25));
  employees.add(new Employee("Sham", 20));

  employees.add(new Employee("Mayur", 5));


  for (Employee employee : employees) { System.out.println(employee);

  }
```

```
}
}
class Employee implements Comparable<Employee> { String name;
int age;
public Employee(String name, int age) {
  this.name = name;
  this.age = age;
}
public int compareTo(Employee e) {
  return this.age - e.age;
}
public String toString() { return name =" "+ age;
}
}
```

Output:

20. Create the list of patients and display the names of patients starting with ‘A’

Solution:-

```
import java.util.*;
public class Code2 {
public static void main(String[] args) {
  ArrayList<String> patients = new ArrayList<String>();
  // Add patients to the list patients.add("Alice");
  patients.add("Bob"); patients.add("Alex");
  patients.add("Amy");
```

```
   // Display names of patients starting with 'A'
   for (String patient : patients) { if
    (patient.startsWith("A")) {
      System.out.println(patient);
       }
      }
}
}
```

Output: