**JAVA ASSIGNMENT – 3**

**Name :- Rohan Dhanani Jitendrabhai**

**PRN :- 1132220432**

**Roll No :- 58**

Q – 1) 

import java.lang.\*;

class A extends Thread {

    public void run() {

        for (int i = 0; i < 100; i++) {

            if (i % 8 == 0) {

                System.out.println("divisible by 8 - " + i);

            }

        }

    }

}

class B extends Thread {

    public void run() {

        for (int i = 51; i < 100; i++) {

            if (i % 2 == 0) {

                System.out.println("even number between 51 to 100 - " + i);

            }

        }

    }

}

class C extends Thread {

    public void run() {

        for (int i = 0; i < 10; i++) {

            System.out.println(i + " - Java is Awesome");

        }

    }

}

class Que1 {

    public static void main(String[] args) throws InterruptedException {

        A t = new A();

        t.start();

        B t1 = new B();

        t1.start();

        C t2 = new C();

        t2.start();

    }

}

Q – 2) Write an application that will create following threads: -- Which will print A to Z 50 times ? -- And 15 – terms of Fibonacci Series

import java.lang.\*;

import java.io.\*;

class A extends Thread {

    public void run() {

        char c;

        for (int i = 1; i <= 50; i++) {

            System.out.print(i + " - ");

            for (c = 'A'; c <= 'Z'; ++c) {

                System.out.print(c + " ");

            }

            System.out.println("");

        }

    }

}

class Fibonacci extends Thread {

    public void run() {

        try {

            int a = 0, b = 1, c = 0;

            BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

            System.out.print("Enter the Limit for fabonacci: ");

            int n = Integer.parseInt(br.readLine());

            System.out.println("Fibonacci series:");

            while (n > 0) {

                System.out.print(c + " ");

                a = b;

                b = c;

                c = a + b;

                n = n - 1;

            }

            System.out.println(" ");

        } catch (Exception ex) {

            ex.printStackTrace();

        }

    }

}

public class Que2 {

    public static void main(String[] args) {

        try {

            Fibonacci fib = new Fibonacci();

            fib.start();

            fib.sleep(4000);

            A t = new A();

            t.start();

        } catch (Exception ex) {

            ex.printStackTrace();

        }

    }

}

Q – 3) Write a program to create two threads which will display message ‘n’ number

of times. While creating thread pass the message and n as parameters.

Message should appear in alternate order.

import java.lang.\*;

class A implements Runnable {

    public void run() {

        for (int i = 0; i < 10; i++) {

            System.out.println("I am a thread A.");

        }

    }

}

class B implements Runnable {

    public void run() {

        for (int i = 0; i < 10; i++) {

            System.out.println("I am a thread B.");

        }

    }

}

public class Que3 {

    public static void main(String[] args) {

        A r = new A();

        B s = new B();

        Thread t = new Thread(r);

        t.start();

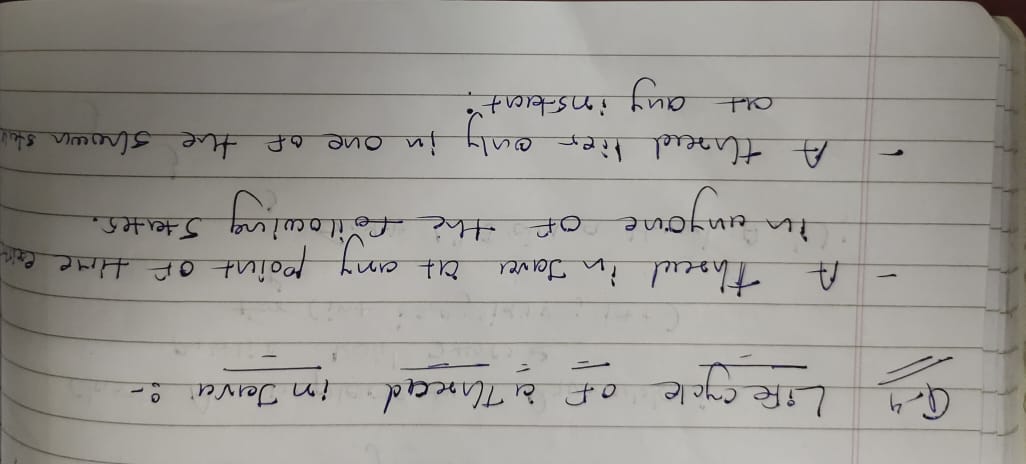
        Thread t1 = new Thread(s);

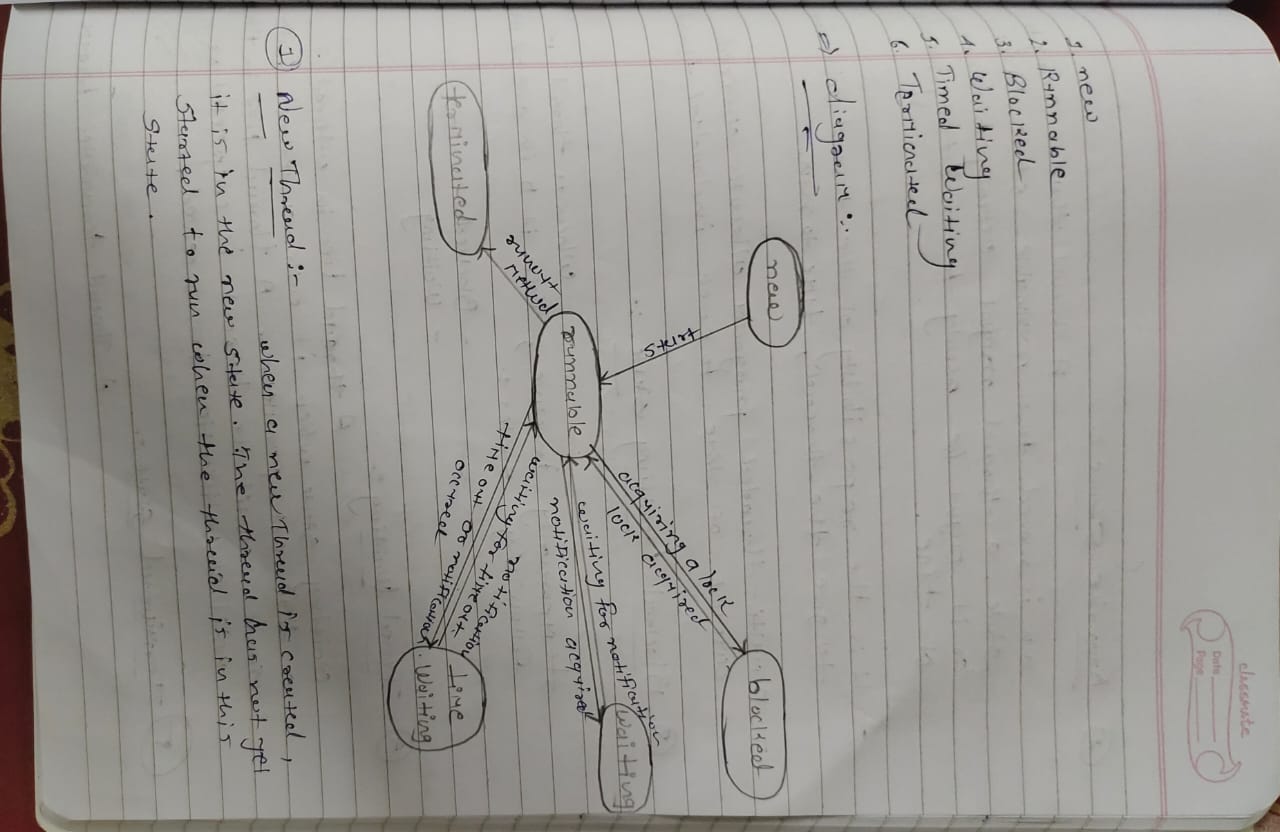
        t1.start();

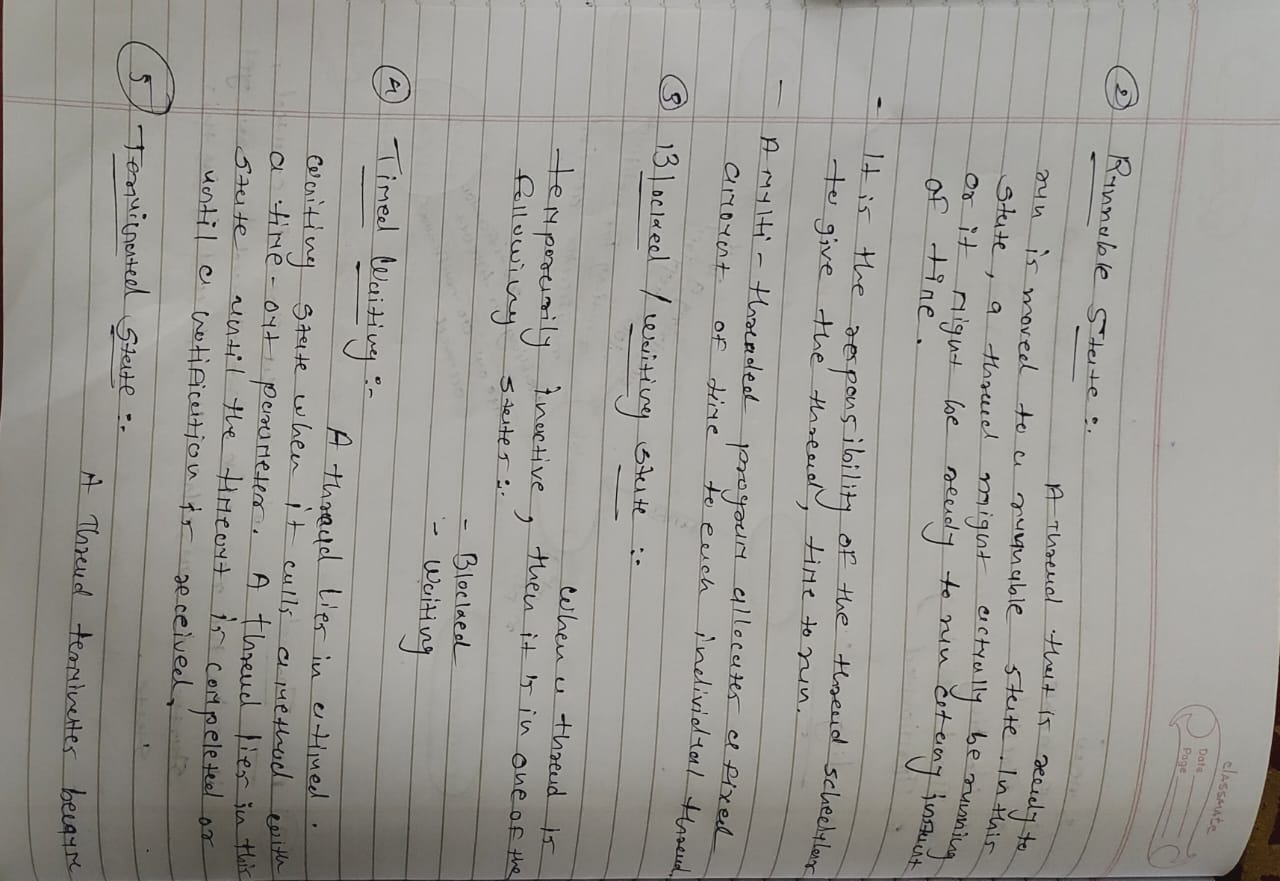
    }

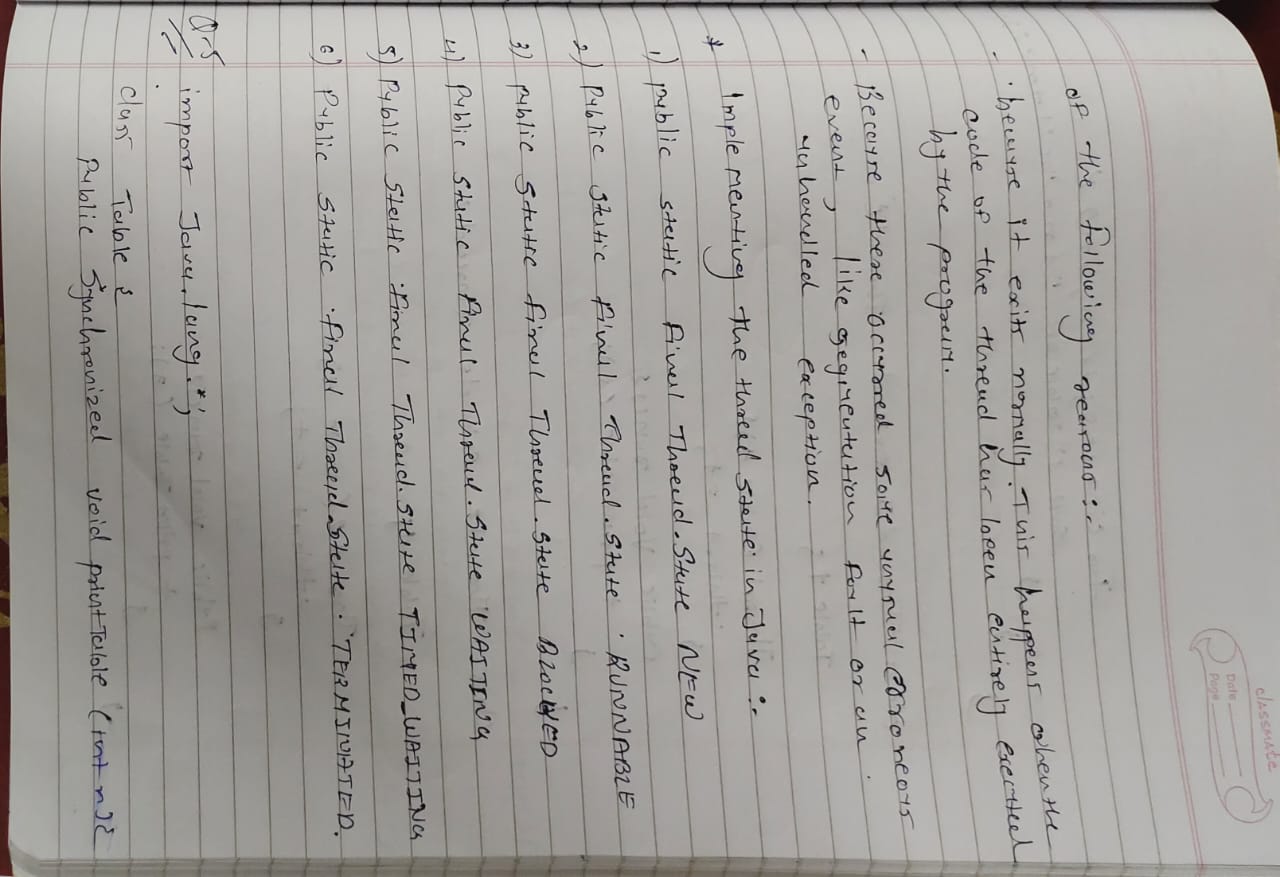
}

Q – 4) Explain about Thread Life Cycle.









Q- 5) Write a program which demonstrate Thread Synchronization by using Synchronized statement and Synchronized Methods.

(Write short note on Thread Synchronization and explain with example)

import java.lang.\*;

class Table {

    public synchronized void printTable(int n) {

        for (int i = 0; i <= 10; i++) {

            System.out.println("Table of " + n + " - " + n \* i);

        }

    }

}

class Thread1 extends Thread {

    Table t;

    Thread1(Table t) {

        this.t = t;

    }

    public void run() {

        t.printTable(5);

    }

}

class Thread2 extends Thread {

    Table t;

    Thread2(Table t) {

        this.t = t;

    }

    public void run() {

        t.printTable(7);

    }

}

public class Que5 {

    public static void main(String[] args) {

        Table obj = new Table();

        Thread1 t1 = new Thread1(obj);

        Thread2 t2 = new Thread2(obj);

        t1.start();

        t2.start();

    }

}

Q – 6) Write a Java Program which demonstrate Interthread communication by using wait(), notify() and notifyAll().

class Customer {

    int amount = 10000;

    synchronized void withdraw(int amount) {

        System.out.println("going to withdraw...");

        if (this.amount < amount) {

            System.out.println("Less balance; waiting for deposit...");

            try {

                wait();

            } catch (Exception e) {

            }

        }

        this.amount -= amount;

        System.out.println("withdraw completed...");

    }

    synchronized void deposit(int amount) {

        System.out.println("going to deposit...");

        this.amount += amount;

        System.out.println("deposit completed... ");

        notify();

    }

}

class Que6 {

    public static void main(String args[]) {

        final Customer c = new Customer();

        new Thread() {

            public void run() {

                c.withdraw(5000);

            }

        }.start();

        new Thread() {

            public void run() {

                c.deposit(10000);

            }

        }.start();

    }

}

Q – 7) Write a multithreaded application bus / railway ticket reservation system.

class Seat {

    static int availableSeats = 10;

    synchronized void reserveSeat(int requestedSeats)

    {

        System.out.println(" ");

        System.out.println(Thread.currentThread().getName() + " entered.");

        System.out.println("Available seats : " + availableSeats + " Requested setas : " + requestedSeats);

        if (availableSeats >= requestedSeats) {

            System.out.println("Seat Available. Reserve now.");

            try {

                Thread.sleep(100);

            } catch (InterruptedException e) {

                System.out.println("Thread interrupted");

            }

            System.out.println(requestedSeats + " seats reserved.");

            availableSeats = availableSeats - requestedSeats;

        } else {

            System.out.println("Requested seats not available."

            );

        }

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

    }

}

class Book extends Thread {

    Seat reserve;

    int requestedSeats;

    public Book(Seat reserve, int requestedSeats) {

        this.reserve = reserve;

        this.requestedSeats = requestedSeats;

    }

    @Override

    public void run()

    {

        reserve.reserveSeat(requestedSeats);

    }

}

class Que7 {

    public static void main(String[] args) {

        Seat reserve = new Seat();

        Book thread1 = new Book(reserve, 2);

        thread1.start();

        Book thread2 = new Book(reserve, 3);

        thread2.start();

        Book thread3 = new Book(reserve, 3);

        thread3.start();

        Book thread4 = new Book(reserve, 3);

        thread4.start();

    }

}

Q – 8) Write a Java program that creates three threads. First thread displays “Hello!” every one second, the second thread displays “Wear Mask !” every two seconds and “Use Sanitizer !” every 5 seconds.

import java.lang.\*;

class first extends Thread {

    @Override

    public void run() {

        try {

            for (int i = 0; i < 5; i++) {

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

                Thread.sleep(1000);

            }

        } catch (InterruptedException e) {

            System.out.println(e.getMessage());

        }

    }

}

class Second extends Thread {

    @Override

    public void run() {

        try {

            for (int i = 0; i < 5; i++) {

                System.out.println("Wear Mask !");

                Thread.sleep(2000);

            }

        } catch (InterruptedException e) {

            System.out.println(e.getMessage());

        }

    }

}

class Third extends Thread {

    @Override

    public void run() {

        try {

            for (int i = 0; i < 5; i++) {

                System.out.println("Use Sanitizer !");

                Thread.sleep(5000);

            }

        } catch (InterruptedException e) {

            System.out.println(e.getMessage());

        }

    }

}

public class Que8 {

    public static void main(String[] args) throws InterruptedException {

        first t = new first();

        Second t1 = new Second();

        Third t2 = new Third();

        t.start();

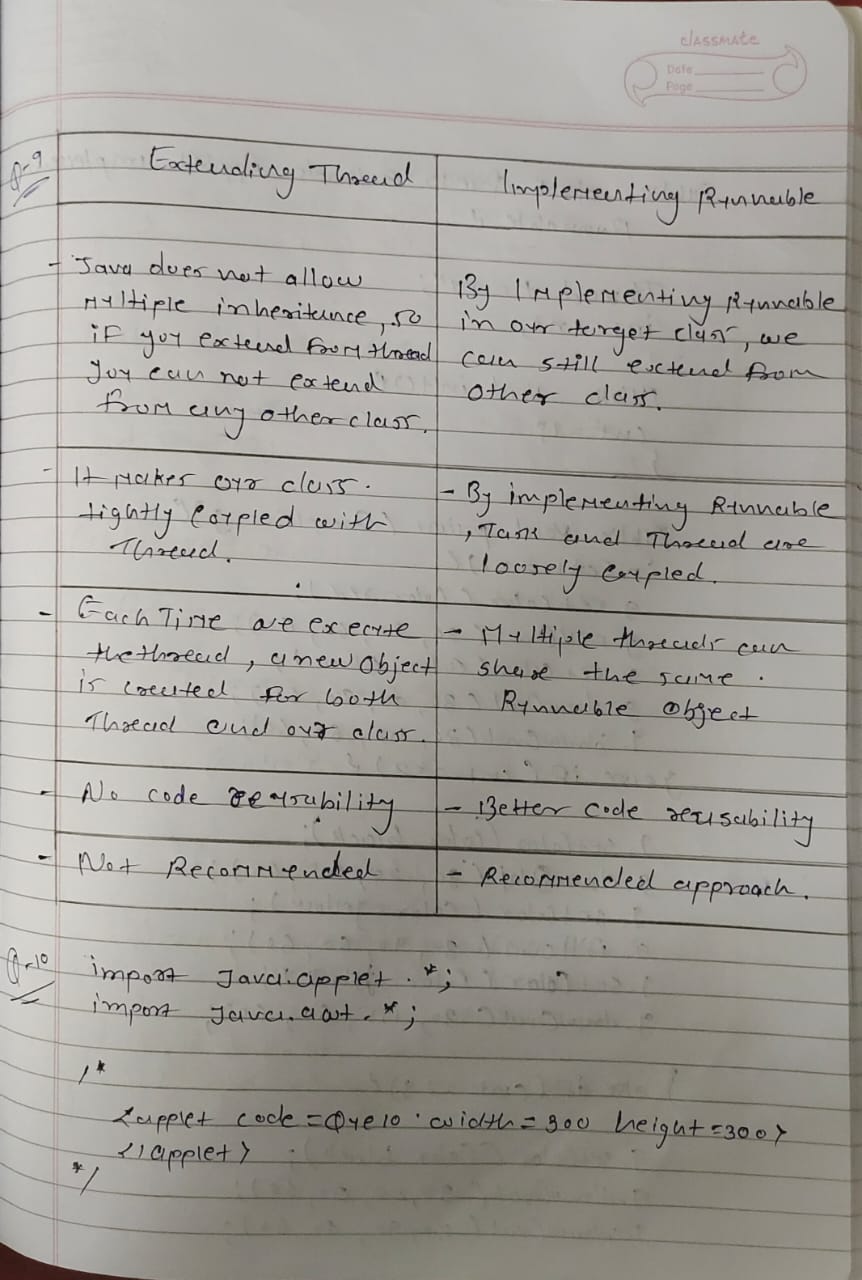
        t1.start();

        t2.start();

    }

}

Q – 9) Write the difference between Extending thread and implementing runnable?



Q – 10) Write a threaded applet which will display circle with different colours.

Colour will change after 1 second. Accept radius of the circle as

Parameter

(i.e. Write a Multithreaded application which demonstrate Traffic Signal).

import java.applet.\*;

import java.awt.\*;

/\*

    <applet code=Que10 width=300 height=300>

    </applet>

\*/

public class Que10 extends Applet implements Runnable {

  Thread t = null;

  int cnt;

  public void init() {

    cnt = 1;

  }

  public void paint(Graphics g) {

    if (cnt == 1) {

      g.setColor(Color.red);

      g.fillOval(30, 20, 60, 60);

      g.setColor(Color.black);

      g.drawOval(30, 90, 60, 60);

      g.drawOval(30, 155, 60, 60);

    } else if (cnt == 2) {

      g.setColor(Color.black);

      g.drawOval(30, 20, 60, 60);

      g.setColor(Color.yellow);

      g.fillOval(30, 90, 60, 60);

      g.setColor(Color.black);

      g.drawOval(30, 155, 60, 60);

    } else if (cnt == 3) {

      g.setColor(Color.black);

      g.drawOval(30, 20, 60, 60);

      g.drawOval(30, 90, 60, 60);

      g.setColor(Color.green);

      g.fillOval(30, 155, 60, 60);

    }

  }

  public void start() {

    t = new Thread(this, "Que9");

    t.start();

  }

  public void run() {

    for (;;) {

      try {

        if (cnt == 3) {

          cnt = 1;

        } else {

          cnt++;

        }

        repaint();

        Thread.sleep(5000);

      } catch (InterruptedException e) {

        System.out.println(e);

      }

    }

  }

}

Q – 11) Write a Multithreaded application which demonstrate Bouncing Ball Applet.

import java.applet.\*;

import java.awt.\*;

import java.awt.event.\*;

/\*<applet code="Que11" width="500" height="500">

 </applet>\*/

public class Que11 extends Applet implements ActionListener {

  Button btn;

  Ball b1, b2;

  int cntr = 0;

  public void init() {

    btn = new Button("Start");

    add(btn);

    btn.addActionListener(this);

  }

  public void paint(Graphics g) {

    try {

      g.setColor(Color.RED);

      g.fillOval(b1.x, b1.y, b1.w, b1.h);

      g.setColor(Color.GREEN);

      g.fillOval(b2.x, b2.y, b2.w, b2.h);

    } catch (NullPointerException e) {}

  }

  public void actionPerformed(ActionEvent ae) {

    if (cntr == 0) {

      b1 = new Ball(0, 0, 50, 50);

      cntr++;

    } else if (cntr == 1) {

      b2 = new Ball(75, 0, 50, 50);

      cntr++;

    } else {

      btn.setEnabled(false);

    }

  }

  class Ball implements Runnable {

    Thread t = null;

    int x, y, w, h, incr\_y, incr\_x;

    Ball(int x, int y, int w, int h) {

      this.x = x;

      this.y = y;

      this.w = w;

      this.h = h;

      incr\_y = 1;

      incr\_x = 1;

      t = new Thread(this);

      t.start();

    }

    public void run() {

      while (true) {

        y = y + incr\_y;

        x = x + incr\_x;

        if ((y == getSize().height - h) || (y == 0)) incr\_y = (-1) \* incr\_y;

        if (x == getSize().width - w || x == 0) incr\_x = (-1) \* incr\_x;

        try {

          t.sleep(10);

        } catch (InterruptedException e) {}

        repaint();

      }

    }

  }

}

Q – 12) Write a Java program to accept a file name from command prompt. If the file exists, then display number of words and lines in that file using FileReader class.

import java.io.\*;

public class Que12 {

  public static void main(String[] args) {

    BufferedReader reader = null;

    int chCount = 0;

    int woCount = 0;

    int liCount = 0;

    try {

      reader = new BufferedReader(new FileReader(args[0]));

      String currLine = reader.readLine();

      while (currLine != null) {

        liCount++;

        String[] words = currLine.split(" ");

        woCount = woCount + words.length;

        for (String word : words) {

          chCount = chCount + word.length();

        }

        currLine = reader.readLine();

      }

      System.out.println(

        "Number Of Chars In " + args[0] + " File : " + chCount

      );

      System.out.println(

        "Number Of Words In " + args[0] + " File : " + woCount

      );

      System.out.println(

        "Number Of Lines In " + args[0] + " File : " + liCount

      );

    } catch (Exception e) {

      System.out.println("File not found ! Please Give Proper file name.");

    }

  }

}

**FILE HANDLING**

Q – 1) Write a java program to display the contents of the file in reverse order.

import java.io.\*;

import java.util.\*;

public class Fq1 {

  public static void main(String[] args) throws FileNotFoundException {

    File f = new File("rohan.txt");

    Scanner input = new Scanner(f);

    String result = "";

    while (input.hasNextLine()) {

      String fjala = input.next();

      for (int i = fjala.length() - 1; i >= 0; i--) {

        result += fjala.charAt(i);

      }

    }

    input.close();

    System.out.print(result += " ");

  }

}

Q – 2) Write a Java Program to append the content of File f1(abc.txt) to File f2(xyz.txt)

import java.io.\*;

public class Fq2 {

  public static void main(String[] args) {

    try {

      FileReader fr = new FileReader("rohan.txt");

      FileWriter fw = new FileWriter("rohan1.txt");

      String str = " ";

      int i;

      while ((i = fr.read()) != -1) {

        str += (char) i;

      }

      fw.write(str);

      fr.close();

      fw.close();

    } catch (Exception e) {

      System.out.println("There are some IOException");

    }

  }

}

Q – 3) Write a Java Program to take charters input from user once user enters ‘q’ quit from the file and display the content added in the file.

import java.io.\*;

import java.util.\*;

import java.util.Scanner;

public class Fq3 {

  public static void main(String[] args) {

    try {

      Scanner sc = new Scanner(System.in);

      String c = sc.nextLine();

      String str = " ";

      for (int i = 0; i < Math.min(c.length(), c.length()); i++) {

        if (c.charAt(i) == 'q') {

          break;

        } else {

          str += c.charAt(i);

        }

      }

      FileWriter obj = new FileWriter("fq3.txt");

      obj.write(str);

      obj.close();

      File obj1 = new File("fq3.txt");

      Scanner reader = new Scanner(obj1);

      while (reader.hasNextLine()) {

        String data = reader.nextLine();

        System.out.println(data);

      }

      reader.close();

    } catch (Exception e) {

      System.out.println("An error occured");

    }

  }

}

Q – 4) Write a Java program to take a file path in a TextField and display the content present in the file in TextArea.

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

public class FileViewDemo extends Frame implements ActionListener

{

    Label l1;

    Button bt;

    TextField tf;

    TextArea ta;

    public FileViewDemo()

    {

        setTitle("ViewFile");

        Panel p1=new Panel();

        p1.setLayout(new FlowLayout());

        ta=new TextArea(10,60);

        p1.add(ta);

        ta.setBackground(Color.yellow);

        ta.setForeground(Color.black);

        Panel p2=new Panel();

        p2.setLayout(new FlowLayout());

            l1=new Label("File name");

        tf=new TextField(" ",12);

        p2.add(l1);

        p2.add(tf);

        tf.setBackground(Color.yellow);

        tf.setForeground(Color.red);

        bt=new Button("View");

        p2.add(bt);

        setLayout(new FlowLayout());

        add(p1);

        add(p2);

        bt.addActionListener(this);

    }

    public void actionPerformed(ActionEvent e)

    {

        if(e.getSource()==bt)

        displayFile();

    }

    void displayFile()

    {

        BufferedReader inFile=null;

        String filename=tf.getText().trim();

        String inLine;

        try

        {

            FileReader fr=new FileReader(filename);

            inFile=new BufferedReader(fr);

            inLine=inFile.readLine();

            boolean firstLine=true;

            while(inLine!=null)

            {

                if(firstLine)

                {

                    ta.append(inLine);

                    firstLine =false;

                }

                else

                {

                    ta.append("\n"+inLine);

                }

                inLine=inFile.readLine();

            }

        }

        catch(FileNotFoundException e)

        {

            System.out.println("File Not Found : "+ filename);

        }

        catch(IOException e)

        {

            System.out.println(e.getMessage());

        }

        finally

        {

            try

            {

                if(inFile!=null)

                {

                    inFile.close();

                }

            }

            catch(IOException e)

            {

                System.out.println("I/O Error encounter.");

            }

        }

    }

    public static void main(String args[])

    {

        FileViewDemo f=new FileViewDemo();

        f.setSize(400,300);

        f.setVisible(true);

    }

}

Q – 5) Write a program to count characters, word and lines in a file . Read the file name from User.

import java.io.\*;

public class Fq5 {

  public static void main(String[] args) {

    BufferedReader reader = null;

    int chCount = 0;

    int woCount = 0;

    int liCount = 0;

    try {

      reader = new BufferedReader(new FileReader(args[0]));

      String currLine = reader.readLine();

      while (currLine != null) {

        liCount++;

        String[] words = currLine.split(" ");

        woCount = woCount + words.length;

        for (String word : words) {

          chCount = chCount + word.length();

        }

        currLine = reader.readLine();

      }

      System.out.println(

        "Number Of Chars In " + args[0] + " File : " + chCount

      );

      System.out.println(

        "Number Of Words In " + args[0] + " File : " + woCount

      );

      System.out.println(

        "Number Of Lines In " + args[0] + " File : " + liCount

      );

    } catch (Exception e) {

      System.out.println("File not found ! Please Give Proper file name.");

    }

  }

}