JAVA PROGRAMMING LAB MANUAL

Year & Semester: II&II

REGULATION-R18

Prepared By: D.Ramreddy
M.Pavan kumar

Verified By:

COMPUTERSCIENCE AND ENGINEERING

VAAGDEVI ENGINEERING COLLEGE

BOLLIKUNTA, WARANGAL-506005

CS308ES: JAVA programming LAB

B.Tech. II Year II Sem. **L T P C**

0032

Course Objectives:

To write programs using abstract classes.
To write programs for solving real world problems using java collection frame work.To write multithreaded programs.
To write GUI programs using swing controls in Java.To introduce java compiler and eclipse platform.
To impart hands on experience with java programming.

Course Outcomes:

Able to write programs for solving real world problems using java collection framework.

Able to write programs using abstract classes.Able to write multithreaded programs.
Able to write GUI programs using swing controls in Java.

Note:

1. Use Linux and My SQL for the Lab Experiments. Though not mandatory, encouragethe use of Eclipse platform.
2. The list suggests the minimum program set. Hence, the concerned staff is requested toadd more problems to the list as needed.

List of Experiments:

1. Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

2. Write a Java program that works as a simple calculator. Use a grid layout to arrange

buttons for the digits and for the +, -,*, % operations. Add a text field to display theresult. Handle any possible exceptions like divided by zero.

3. A) Develop an applet in Java that displays a simple message.

   b) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.

4. Write a Java program that creates a user interface to perform integer divisions. Theuser enters two numbers in the text fields, Num1 and Num2. The division of Num1and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display theexception in a message dialog box.

5. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third threadwill print the value of cube of the number.

6. Write a Java program for the following:
   i) Create a doubly linked list of elements.
   ii) Delete a given element from the above list.
   iii)Display the contents of the list after deletion.

7. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially, there is no message shown.

8. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.

10. Write a Java program that handles all mouse events and shows the event name at thecenter of the window when a mouse event is fired (Use Adapter classes).

11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other valuefrom the hash table (hint: use hash tables).

12. Write a Java program that correctly implements the producer – consumer problemusing the concept of inter thread communication.

13. Write a Java program to list all the files in a directory including the files present in allits subdirectories.

14. Write a Java program that implements Quick sort algorithm for sorting a list of namesin ascending order

15. Write a Java program that implements Bubble sort algorithm for sorting in descendingorder and also shows the number of interchanges occurred for the given set of integers.

REFERENCE BOOKS

1. Java for Programmers, P. J. Deitel and H. M. Deitel, 10th Edition Pearson education.
2. Thinking in Java, Bruce Eckel, Pearson Education.
3. Java Programming, D. S. Malik and P. S. Nair, Cengage Learning.
4. Core Java, Volume 1, 9th edition, Cay S. Horstmann and G Cornell, Pearso

1. Use Eclipse or Net bean platform and acquaint with the various menus. Create a test project, add a test class, and run it. See how you can use auto suggestions, auto fill. Try code formatter and code refactoring like renaming variables, methods, and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.

gedit Eclipse.java

```
class EvenOrOdd
{
   public static void main(String args[])
    {
      int n=Integer.parseInt(args[0]);
      for(int i=0;i<=n;i++)
        {
            if(i%2==0)
              System.out.println("it is even number"+i);
           else
              System.out.println("it is odd number"+i);

        }
    }
}
```

OUTPUT
javac
Eclipse.javajava
EvenOrOdd

```
it is even number0
it is odd number1
it is even number2
it is odd number3
it is even number4
it is odd number5
it is even number6
it is odd number7
it is even number8
it is odd number9
it is even number10
it is odd number11
it is even number12
it is odd number13
it is even number14
it is odd number15
```

2. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +, -,*, % operations. Add a text field to display the result. Handle any possible exceptions like divided by zero.

Gedit Calculator.java

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

public class Calculator implements ActionListener
{
   int c,n;
   String s1,s2,s3,s4,s5;
   Frame f;
   Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b10,b11,b12,b13,b14,b15,b16,b17;
   Panel p;
   TextField tf;
   GridLayout g;
   Calculator()
   {
      f = new Frame("My calculator");
      p = new Panel();
      f.setLayout(new FlowLayout());
      b1 = new Button("0");
      b1.addActionListener(this);
      b2 = new Button("1");
      b2.addActionListener(this);
      b3 = new Button("2");
      b3.addActionListener(this);
       b4 = new Button("3");
      b4.addActionListener(this);
       b5 = new Button("4");
      b5.addActionListener(this);
       b6 = new Button("5");
      b6.addActionListener(this);
      b7 = new Button("6");
      b7.addActionListener(this);
       b8 = new Button("7");
      b8.addActionListener(this);
      b9 = new Button("8");
      b9.addActionListener(this);
```

```
     b10 = new Button("9");
    b10.addActionListener(this);
    b11 = new Button("+");
    b11.addActionListener(this);
    b12 = new Button("-");
    b12.addActionListener(this);
    b13 = new Button("*");
    b13.addActionListener(this);
    b14 = new Button("/");
    b14.addActionListener(this);
    b15 = new Button("%");
    b15.addActionListener(this);
    b16 = new Button("=");
    b16.addActionListener(this);
    b17 = new Button("C");
    b17.addActionListener(this);
    tf = new TextField(20);
    f.add(tf);
    g = new GridLayout(4,4,10,20);
    p.setLayout(g);
     p.add(b1);
    p.add(b2);
    p.add(b3);
    p.add(b4);
    p.add(b5);
    p.add(b6);
    p.add(b7);
    p.add(b8);
    p.add(b9);
    p.add(b10);
    p.add(b11);
    p.add(b12);
    p.add(b13);
    p.add(b14);
    p.add(b15);
    p.add(b16);
    p.add(b17);
    f.add(p);
    f.setSize(300,300);
    f.setVisible(true);
}
```

```
 public void actionPerformed(ActionEvent e)
    {
       if(e.getSource()==b1)
         {
           s3 = tf.getText();
            s4 = "0";
            s5 = s3+s4;
            tf.setText(s5);
         }
     if(e.getSource()==b2)
        {
            s3 = tf.getText();
            s4 = "1";
            s5 = s3+s4;
            tf.setText(s5);
        }
if(e.getSource()==b3)
   {
       s3 = tf.getText();
       s4 = "2";
        s5 = s3+s4;
      tf.setText(s5);
   }
if(e.getSource()==b4)
  {
     s3 = tf.getText();
    s4 = "3";
    s5 = s3+s4;
    tf.setText(s5);
  }
if(e.getSource()==b5)
 {
    s3 = tf.getText();
    s4 = "4";
    s5 = s3+s4;

    tf.setText(s5);
 }

if(e.getSource()==b6)
```

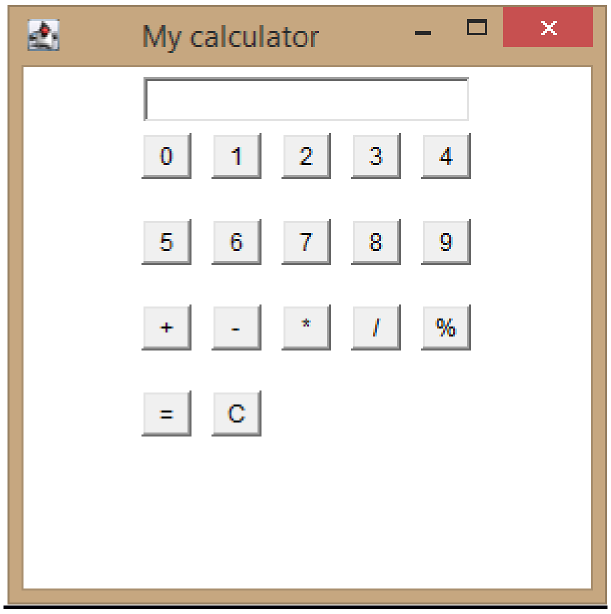
```
{
   s3 = tf.getText();
   s4 = "5";
   s5 = s3+s4;
   tf.setText(s5);
}
if(e.getSource()==b7)
{
   s3 = tf.getText();
   s4 = "6";
   s5 = s3+s4;
   tf.setText(s5);
}
if(e.getSource()==b8)
{
   s3 = tf.getText();
   s4 = "7";
   s5 = s3+s4;
   tf.setText(s5);
}
if(e.getSource()==b9)
{
   s3 = tf.getText();
   s4 = "8";
   s5 = s3+s4;
   tf.setText(s5);
}
if(e.getSource()==b10)
{
   s3 = tf.getText();
   s4 = "9";
   s5 = s3+s4;
   tf.setText(s5);
}
if(e.getSource()==b11)
{
   s1 = tf.getText();

   tf.setText("");
   c=1;
```

```
}
if(e.getSource()==b12)
{
    s1 = tf.getText();
    tf.setText("");
    c=2;

}
if(e.getSource()==b13)
{
    s1 = tf.getText();
    tf.setText("");
     c=3;

}
if(e.getSource()==b14)
{
    s1 = tf.getText();
    tf.setText("");
    c=4;

}
if(e.getSource()==b15)
{
    s1 = tf.getText();
    tf.setText("");
    c=5;
}
if(e.getSource()==b16)
{
    s2 = tf.getText();
    if(c==1)
     {
        n = Integer.parseInt(s1)+Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
     }
    else if(c==2)
      {
        n = Integer.parseInt(s1)-Integer.parseInt(s2);
        tf.setText(String.valueOf(n));
```

```
            }
          else if(c==3)
            {
             n = Integer.parseInt(s1)*Integer.parseInt(s2);
            tf.setText(String.valueOf(n));
            }
          if(c==4)
            {
            try
              {
                int p=Integer.parseInt(s2);
               if(p!=0)
                 {
                   n = Integer.parseInt(s1)/Integer.parseInt(s2);
                   tf.setText(String.valueOf(n));
                 }
               else
                   tf.setText("infinite");
               }
            catch(Exception i)
              {
              }
             }
          if(c==5)
              {
                  n = Integer.parseInt(s1)%Integer.parseInt(s2);
                   tf.setText(String.valueOf(n));
              }
      }
      if(e.getSource()==b17)
      {
          tf.setText("");
      }
   }

   public static void main(String[] args)
     {
      Calculator v = new Calculator();
     }
}
```

Output

```
javac Calculator.java
java Calculator
```



3. A) Develop an applet in Java that displays a simple message.

B)Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named “Compute” is clicked.

A) Develop an applet in Java that displays a simple message.

<u>Gedit simpleapplet.java</u>

```
import java.applet.*;

import java.awt.*;
public class simpleapplet extends Applet

{
        public void paint(Graphics g)
       {
            g.drawString("hello",50,50);
       }

}
```

Ouput

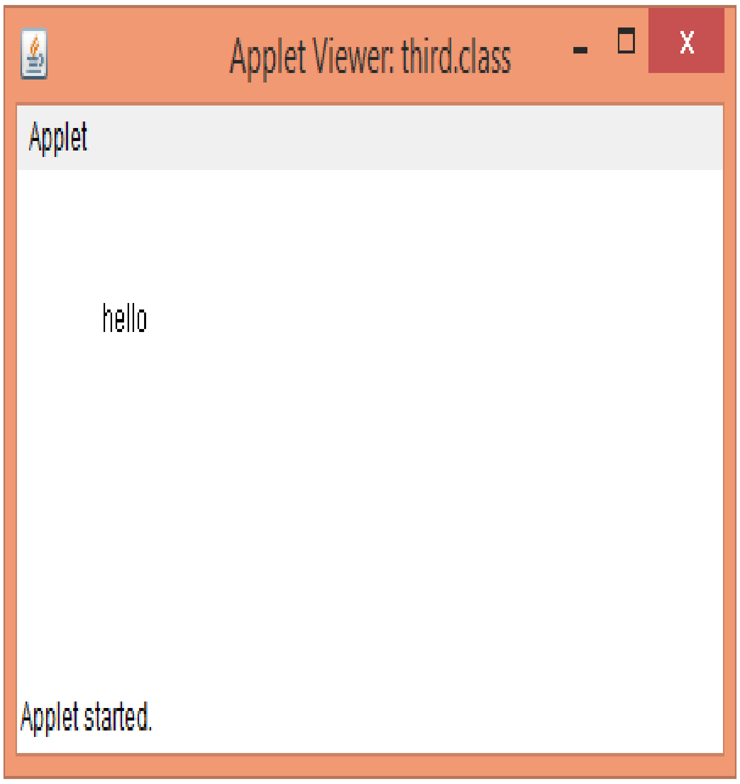
javac

simpleapplet.javagedit

simpleapplet.html

```
<html>
<applet code="simpleapplet.class" height=500 width=700>
</applet>
</html>
```

appletviewer simpleapplet.html



```
hello
```

```
Applet started.
```

B) Develop an applet in Java that receives an integer in one text field, and computes its factorial Value and returns it in another text field, when the button named "Compute" is clicked.

Gedit IntegerApplet.java

```
import java.awt.*;
 import java.awt.event.*;
public class IntegerApplet extends java.applet.Applet implements ActionListener
{
TextField t1,t2;
Label  l1,l2,l3;
Button b1;
int fact=1,n,i;
IntegerApplet e;
    public void init()
    {
      e=this;
      t1=new TextField(10);
      t2=new TextField(10);
      l1=new Label("factorial of a
      number");l2=new Label("enter
      number");
      l3=new Label("result");
      b1=new Button("compute");
      add(l1);
      add(l2);
      add(l3);
      add(t1);
      add(t2);
      add(b1);
      b1.addActionListener(this);
}
public void actionPerformed(ActionEvent ae)
{
       String  str=t1.getText();
       n=Integer.parseInt(str);
       for(i=n;i>1;i--)
{
fact=fact*i;
}
String msg=""+fact;
t2.setText(msg);
 fact=1;
}
}
```

Output

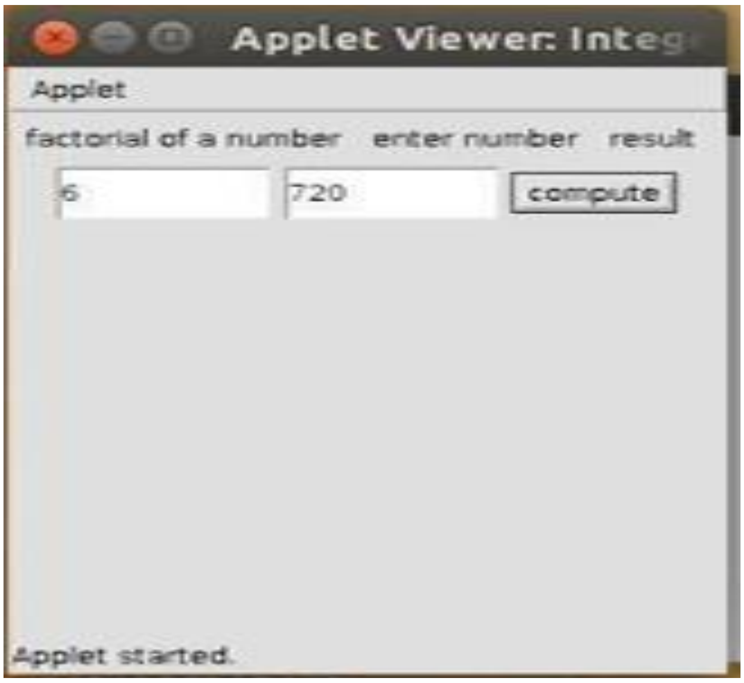
javac

IntegerApplet.javagedit

fact.html

```
<html>
<applet
 code="IntegerApplet.class"
 width=300 height=300>
<param name="t1" value="ram">
<param name="t2" value="98.55">
</applet>
</html>
```

apppletviewer fact.html



```
6
720
Applet started.
```

4.Write a java program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1and Num 2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. Display the exception in a message dialog box.

<u>Gedit Division.java</u>

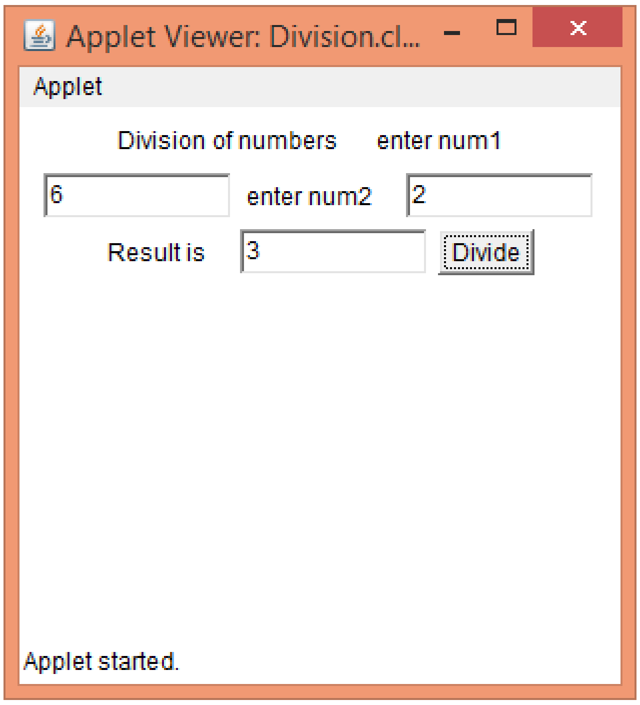
```
import java.awt.*;
import javax.swing.*;
import
 java.applet.Applet;
 import java.awt.event.*;
 public class Division extends Applet implements ActionListener
{
 TextField t1,t2,t3;
 Button b;
 Label  L1,L2,L3,L4;
 String s;
 Division e;
    public void init()
      {
         e=this;
        t1=new TextField(10);
        t2=new TextField(10);
        t3=new TextField(10);
        L1=new  Label("enter num1");
        L2=new Label("enter num2");
        L3=new Label("Result is");
        L4=new Label("Division of numbers");
        b=new Button("Divide");
        add(L4);
        add(L1);
        add(t1);
        add(L2);
        add(t2);
        add(L3);
```

```
            add(t3);
            add(b);
            b.addActionListener(this);
        }

        public void actionPerformed(ActionEvent ae)
         {
            try
             {
               int num1=Integer.parseInt(t1.getText());
                int num2=Integer.parseInt(t2.getText());
               s=""+(num1/num2);
               t3.setText(s);
             }
           catch(ArithmeticException a)
             {
               JOptionPane.showMessageDialog(null,"Divide by zero");
             }
            catch(NumberFormatException b)
             {
              JOptionPane.showMessageDialog(null,"NumberFormatException");
             }
        }
}
```

output:
javac
Division.javagedit
Division.html

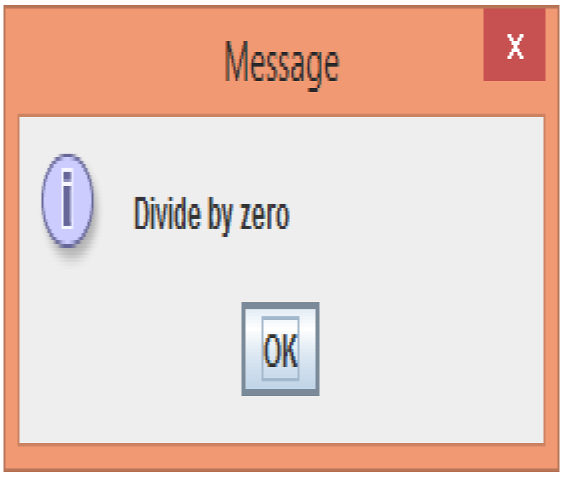
```
<html>
<head>
</head>
<body>
<applet code="Division.class" height=500 width=300></applet>
</body>
</html>
```

<u>appletviewer Division.htm</u>



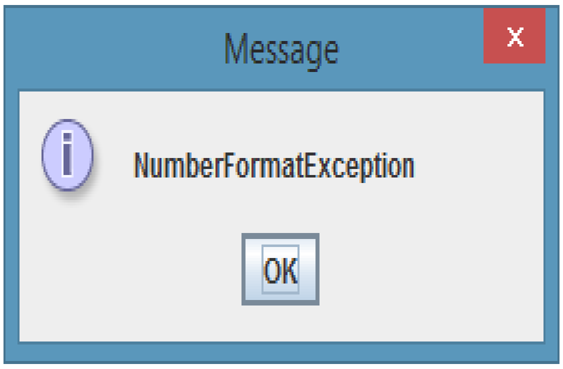
```
Applet Viewer: Division.cl...
Applet
Division of numbers    enter num1
6    enter num2    2
Result is    3    Divide
Applet started.
```

When we entered mun2 as 0 it will gives



```
Message
Divide by zero
OK
```

When we entered non integer values, it throws NumberFormatException



NumberFormatException

5. Write a Java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.

<u>Gedit Mthread.java</u>

```
import java.lang.*;
import java.util.*;
class even implements Runnable
{
   public int x;
   public even (int x)
    {
       this.x=x;
    }
   public void run()
    {
      System.out.println("thread name:even thread and"+x+"is even number and squareof"+x+"is:"+x*x);
    }
 }
 class odd implements Runnable
 {
     public int x;
     public odd(int x)
     {
       this.x=x;
     }
     public void run()
     {
       System.out.println("thread name:oddthread and"+x+"is odd number and cubeof"+x+"is:"+x*x*x);
     }
}
class A extends Thread
{
   public String tname;
   public Random r;
  public Thread t1,t2;
     public A(String s)
      {
        tname=s;
      }
      public void run()
     {
          int num=10;
          r= new Random();
         try
          {
             for(int i=0;i<50;i++)
              {
                num=r.nextInt(100);
                System.out.println("main thread and generated number is "+num);
               if(num%2==0)
                 {
                   t1=new Thread(new even(num));
                   t1.start();
                 }
                else
                   {
                       t2 = new Thread(new odd(num));
                       t2.start();
                   }
                   Thread.sleep(1000);

                    System.out.println("      ");
              }
           }
          catch(Exception e)
          {
            System.out.println(e.getMessage());
          }
     }
}
class Mthread
  {
      public static void main(String args[])
        {
          A a=new A("one");
          a.start();
        }
     }
```

output:
javac Mthread.java
java Mthread

```
main thread and generated number is 2
thread name:even thread and2is even number and square of2is:4
--------

main thread and generated number is 85
thread name:odd thread and85is odd number and cube of85is:614125
--------

main thread and generated number is 92
thread name:even thread and92is even number and square of92is:8464
--------

main thread and generated number is 24
thread name:even thread and24is even number and square of24is:576
--------

main thread and generated number is 74
thread name:even thread and74is even number and square of74is:5476
--------

main thread and generated number is 4
thread name:even thread and4is even number and square of4is:16
--------

main thread and generated number is 59
thread name:odd thread and59is odd number and cube of59is:205379

main thread and generated number is 83
thread name:odd thread and83is odd number and cube of83is:571787
--------

main thread and generated number is 23
thread name:odd thread and23is odd number and cube of23is:12167
--------

main thread and generated number is 24
thread name:even thread and24is even number and square of24is:576
--------
```

```
main thread and generated number is 88
thread name:even thread and88is even number and square of88is:7744
--------

main thread and generated number is 62
thread name:even thread and62is even number and square of62is:3844
--------

main thread and generated number is 75
thread name:odd thread and75is odd number and cube of75is:421875
--------

main thread and generated number is 30
thread name:even thread and30is even number and square of30is:900
--------
```

6. Write a Java program for the following:

i) Create a doubly linked list of elements.

ii)Delete a given element from the above list.

iii)Display the contents of the list after deletion.

<u>Gedit doublylinkedlist.java</u>

```
import java.io.*;
import java.util.Scanner;
class linkedlist
{
    int data;
     linkedlist prev;
    linkedlist next;
    linkedlist(int value)
     {
       this.data = value;
     }
     void display()
      {
        System.out.println(data);
      }
}
class linked
{
    public linkedlist fstnode, lastnode;

    linked()
    {
        fstnode = null;

         lastnode = null;
    }
```

```
  /* Insert node at the beginning or create linked list */
    void insert_front(int value)
      {
         linkedlist node = new linkedlist(value);
         if(fstnode == null)
          {
              node.prev = node.next = null;
              fstnode =lastnode = node;
             System.out.println("Lined list created successfully!");
          }
         else
          {
             node.prev = null;

              node.next = fstnode;

             fstnode.prev = node;

             fstnode = node;

             System.out.println("Node Inserted at the front of the Linked list!");

          }
    }

/* Insert node at the end or create linked list */
void insert_end(int value)
 {
    linkedlist node = new linkedlist(value);
    if(fstnode == null)
    {
      node.prev = node.next = null;
      fstnode = lastnode = node;
      System.out.println("Lined list created  successfully!");
    }
    else
    {
         node.next = null;
        node.prev = lastnode;
        lastnode.next = node;
        lastnode = node;
         System.out.println("Node Inserted at the end of the Linked list!");
    }
```

```
/* Delete node from linked list */
  void delete()
  {
     int count = 0, number, i;
     linkedlist node, node1, node2;
     Scanner input = new Scanner(System.in);
     for(node = fstnode; node != null; node = node.next)count++;
     display();
     node = node1 = node2 = fstnode;
     System.out.println(count+" nodes available here!");
     System.out.println("Enter the node number which youwant to delete from ascending order list:");
     number = Integer.parseInt(input.nextLine());
     if(number != 1)
      {
       if(number < count && number > 0)
        {
          for(i = 2; i <= number;i++)
            node = node.next;
          for(i = 2; i <= number-1; i++)
            node1 = node1.next;
          for(i = 2; i <= number+1; i++)
            node2 = node2.next;node2.prev = node1;
            node1.next = node2;
            node.prev = null;
            node.next = null;node = null;
        }
        else if(number == count)
         {
           node = lastnode;
           lastnode = node.prev;
           lastnode.next = null;
           node = null;
         }
else

System.out.println("Invalid node number!\n");

}

else

 {
    node = fstnode;
    fstnode = node.next;
    fstnode.prev = null;
    node = null;
 }

            System.out.println("Node has been deleted  successfully!\n");

}

/* Display linked list */
void display()
  {
     linkedlist node = fstnode;
     System.out.println("List of node in Ascending order!");
     while(node != null)
        {
          node.display();
          node = node.next;
        }
       node = lastnode;
       System.out.println("List of node in Descending order!");
       while(node != null)
        {
                      node.display(); node =
                      node.prev;
        }
  }
}
```

```
 class doublylinkedlist
 {
    public static void main(String args[ ])
      {
       linked list = new linked();
       Scanner input = new Scanner(System.in);
       int op = 0;
       while(op != 5)
       {
         System.out.println("1. Insert at front  2. Insertat back 3. Delete 4. Display 5. Exit");
         System.out.println("Enter your choice:");
       op = Integer.parseInt(input.nextLine());
       switch(op)
        {
         case 1:
                System.out.println("Enter the positive value for Linked list:");
               list.insert_front(Integer.parseInt(input.nextLine()));
                break;
         case 2:
                System.out.println("Enternthe positive value for Linked list:");
                 list.insert_end(Integer.parseInt(input.nextLine()));
                break;
          case 3:
                list.delete();
                break;
          case 4:
                list.display();
                break;
          case 5:
                System.out.println("Bye Bye!");
                System.exit(0);
                break;
       default:
       System.out.println("Invalid choice!");
       }
      }
    }
}
```

Output:

javac doublylinkedlist.javajava doublylinkedlist

```
1. Insert at front 2. Insert at back 3. Delete 4. Display 5.
ExitEnter your choice:
1
Enter the positive value for Linked list:
1
Lined list created successfully!
1. Insert at front 2. Insert at back 3. Delete 4. Display 5.
ExitEnter your choice:
2
Enter the positive value for Linked list:

2
Node Inserted at the end of the Linked list!
1. Insert at front 2. Insert at back 3. Delete 4. Display 5.
ExitEnter your choice:
1
Enter the positive value for Linked list:
2
Node Inserted at the front of the Linked list!
1. Insert at front 2. Insert at back 3. Delete 4. Display 5.
ExitEnter your choice:
2
Enter the positive value for Linked list:
3
Node Inserted at the end of the Linked list!
1. Insert at front 2. Insert at back 3. Delete 4. Display 5.
ExitEnter your choice:
4
List of node in Ascending
order!2
```

```
1
2
3
List of node in Descending
order!3
2
1
2
1. Insert at front 2. Insert at back 3. Delete 4. Display 5.
ExitEnter your choice:
4
List of node in Ascending
order!2
1
2
3

List of node in Descending
order!3
2
1
2
1. Insert at front 2. Insert at back 3. Delete 4. Display 5.
Exit
```

7. Write a Java program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green with radio buttons. On selecting a button, an appropriate message with "Stop" or "Ready" or "Go" should appear above the buttons in selected color. Initially, there is no message shown.

<u>Gedit TrafficLight.java</u>

```
import java.applet.Applet;
import java.awt.Checkbox;
import java.awt.CheckboxGroup;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
public class TrafficLight extends Applet implements ItemListener
{
   String msg="";
   Checkbo red,yellow,green;
  CheckboxGroup cg=null; public
  void init()
    {
       cg=new CheckboxGroup();
      Checkbox red=new Checkbox("red",cg,true);
      red.setBackground(Color.red);
      Checkbox yellow=new
      Checkbox("yellow",cg,false);
      yellow.setBackground(Color.yellow);
       Checkbox green=new Checkbox("green",cg,false);
      green.setBackground(Color.green);
      add(red);
      add(yellow);
      add(green);
      red.addItemListener(this);
      yellow.addItemListener(this);
      green.addItemListener(this);
  }
public void itemStateChanged(ItemEvent ie)
 {

   repaint();
 }
public void paint(Graphics g)
 {
   Checkbox chk=cg.getSelectedCheckbox();
  g.drawString(chk.getLabel()+" Is selected",101,70); }
 }
```

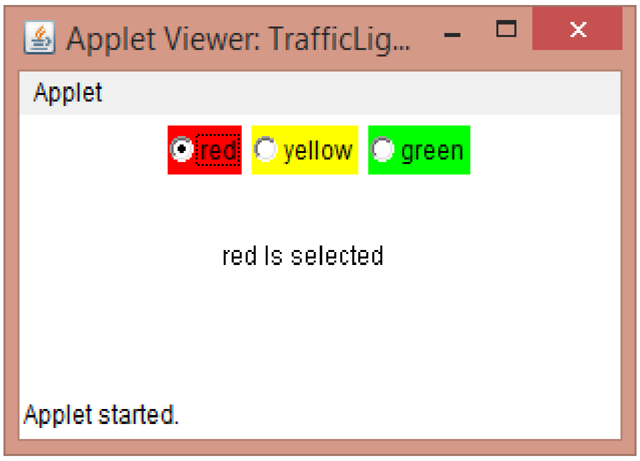
output:

javac TrafficLight.java

gedit tr.html

```
<html>
<head>
</head>
<body>
<applet code="TrafficLight.class" height=500 width=300></applet>
</body>
</html>
```

appletviewer tr.html



red Is selected

Applet started.

8. Write a Java program to create an abstract class named Shape that contains two integers and an empty method named print Area (). Provide three classes named Rectangle, Triangle, and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.

<u>Gedit DemoShape.java</u>

```
abstract class Shape
{
    int a,b;
    abstract void area();
}
class Rectangle extends Shape
{
    void area()
    {
            a=10;b=20;
            System.out.println("Rectangle has four sides");
            System.out.println("recctangle area is"+ a*b);
    }
}
class Triangle extends Shape

{
    void area()
    {
            a=30;b=40;
            System.out.println("Triangle has three sides");
            System.out.println("triangle are is"+ (0.5*a*b));
    }
}
class Circle extends Shape
{
    void area()
    {
            a=50;
            System.out.println("Circle has one radius");
            System.out.println("circle area is"+(3.14*a*a));
    }
}
public class DemoShape

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

```
                Rectangle r=new Rectangle();
                Triangle t=new Triangle();
                Circle c=new Circle();
                r.area();
                t.area();
                c.area();
        }
}
```

Output:

javac DemoShape.java

java DemoShape

```
-

Rectangle has four sides

recctangle area is200

Triangle has three sides

triangle are is600.0

Circle has one radius

circle area is7850.0
```

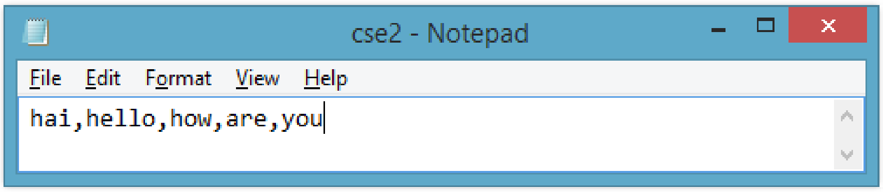
9. Suppose that a table named Table.txt is stored in a text file. The first line in the file is the header, and the remaining lines correspond to rows in the table. The elements are separated by commas. Write a java program to display the table using Labels in Grid Layout.

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;
import java.io.*;
public class Table1 extends JFrame
{
int i=0;
int j=0,k=0;
Object data[][]=new Object[5][4];
Object list[][]=new Object[5][4];
JButton save;
JTable table1; FileInputStream fis;
DataInputStream dis;
 public Table1()
 {
                //String d= "pavan kumar";
                Container con=getContentPane(); con.setLayout(new BorderLayout());
                final String[] colHeads={"Name","Roll Number","Department","Percentage"};
                try
                {
                      String s=JOptionPane.showInputDialog("Enter the File name present in the current
                directory");
                      FileInputStream fis=new FileInputStream(s);
                     //DataInputStream dis = new DataInputStream(fis);
                       BufferedReader dis=new BufferedReader(new InputStreamReader(fis));
                          while ((d=dis.readLine())!=null)
                          {
                                    StringTokenizer st1=new StringTokenizer("");
                                    while (st1.hasMoreTokens())
                                    {
                                             for (j=0;j<4;j++)
                                             {
                                                      data[i][j]=st1.nextToken();
                                                      System.out.println(data[i][j]);
                                             }
                                             i++;
                                    }
```

```
                            System.out.println ("   ");
                            dis.close();
                    }
            }
    catch (Exception e)
            {
                    System.out.println ("Exception raised" +e.toString());
            }
            table1=new JTable(data,colHeads);
            int v=ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
            int h=ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
            JScrollPane scroll=new JScrollPane(table1,v,h); con.add(scroll,BorderLayout.CENTER);
    }
    public static void main(String args[])
    {
        Table1 t=new Table1();
        t.setBackground(Color.green);
        t.setTitle("Display Data");
        t.setSize(500,300);
        t.setVisible(true);
        t.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    }
}
```

<u>Output:-</u>

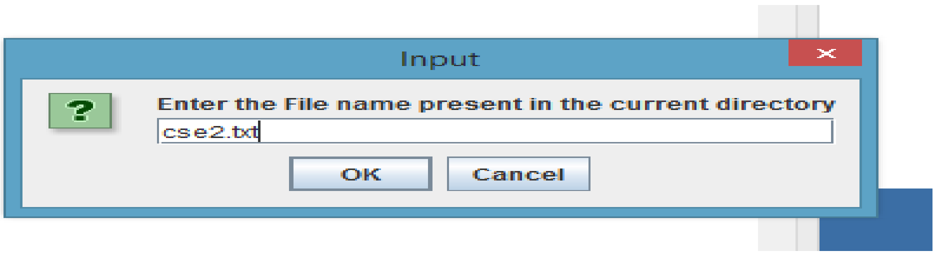
NOTE: Before going to compilation we have to create the one text file like below



```
hai,hello,how,are,you
```

Compile:javac Table1.java

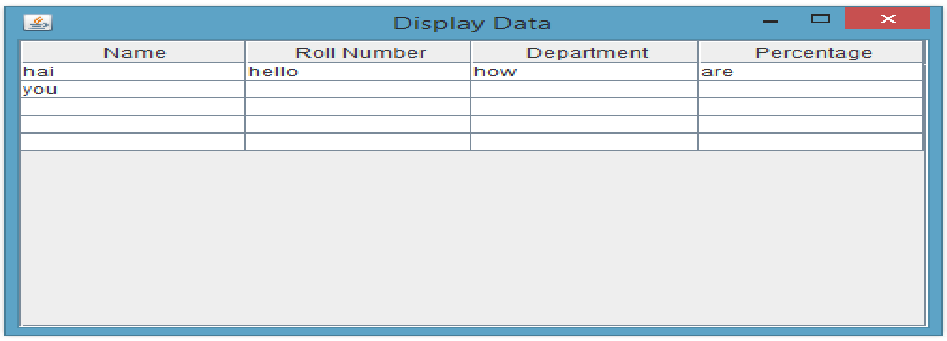
Run:java Table1


Input

Enter the File name present in the current directory

cse2.txt

Here when we submit the ok button it will read the data from cse2.txt file as shown above figure


Display Data

Name	Roll Number	Department	Percentage
hai	hello	how	are
you			

10. Write a Java program that handles all mouse events and shows the event name at thecenter of the window when a mouse event is fired (Use Adapter classes).

<u>Gedit MouseDemo.java</u>

-

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class MouseDemo extends Applet implements MouseListener,MouseMotionListener
{
  int mx=0;
  int my =0;
  String msg="";
  public void init()
  {
         addMouseListener(this);
         addMouseMotionListener(this);
  }
  public void mouseClicked(MouseEvent me)

  {
         mx=20;
         my =40;
         msg="Mouse Clicked";
```

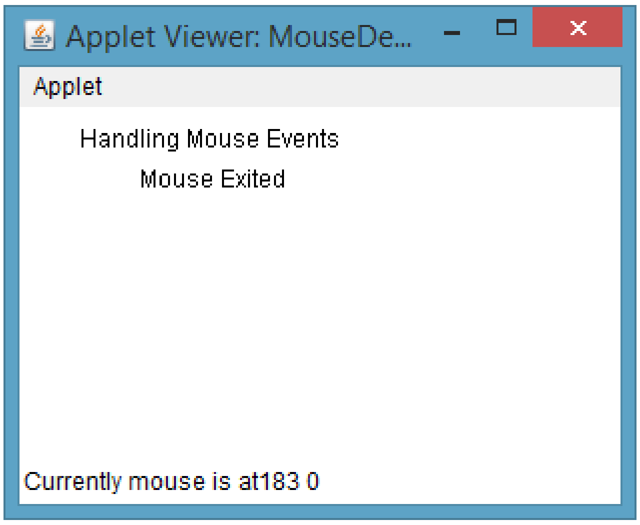
```
        repaint();
}
public void mousePressed(MouseEvent me)
{
        mx=30;
        my =60;
        msg="Mouse Pressed";
        repaint();
}
public void mouseReleased(MouseEvent me)
{
        mx=30;
        my =60;
        msg="Mouse Released";
        repaint();
}
public void mouseEntered(MouseEvent me)
{
        mx=40;
        my =80;
        msg="Mouse Entered";
        repaint();
}
public void mouseExited(MouseEvent me)
{
        mx=40;
        my =80;
        msg="Mouse Exited";
        repaint();
}
public void mouseDragged(MouseEvent me)
{
        mx=me.getX();
        my =me.getY();
        showStatus("Currently mouse dragged"+mx+" "+my);
        repaint();
}
public void mouseMoved(MouseEvent me)
{
        mx=me.getX();
        my =me.getY();
        showStatus("Currently mouse is at"+mx+" "+my);
        repaint();
}
```

```
  public void paint(Graphics g)
  {
            g.drawString("Handling Mouse Events",30,20);
            g.drawString(msg,60,40);
  }
}
```

output:
javac
MouseDemo.javagedit
MouseDemo.html

```
<html>
<head>
</head>
<body>
<applet code="MouseDemo.class" height=500 width=300></applet>
</body>
</html>
```

appletviewer MouseDemo.html



```
Handling Mouse Events
        Mouse Exited

Currently mouse is at183 0
```

11. Write a Java program that loads names and phone numbers from a text file where the data is organized as one line per record and each field in a record are separated by a tab (\t). It takes a name or phone number as input and prints the corresponding other value from the hash table (hint: use hash tables).

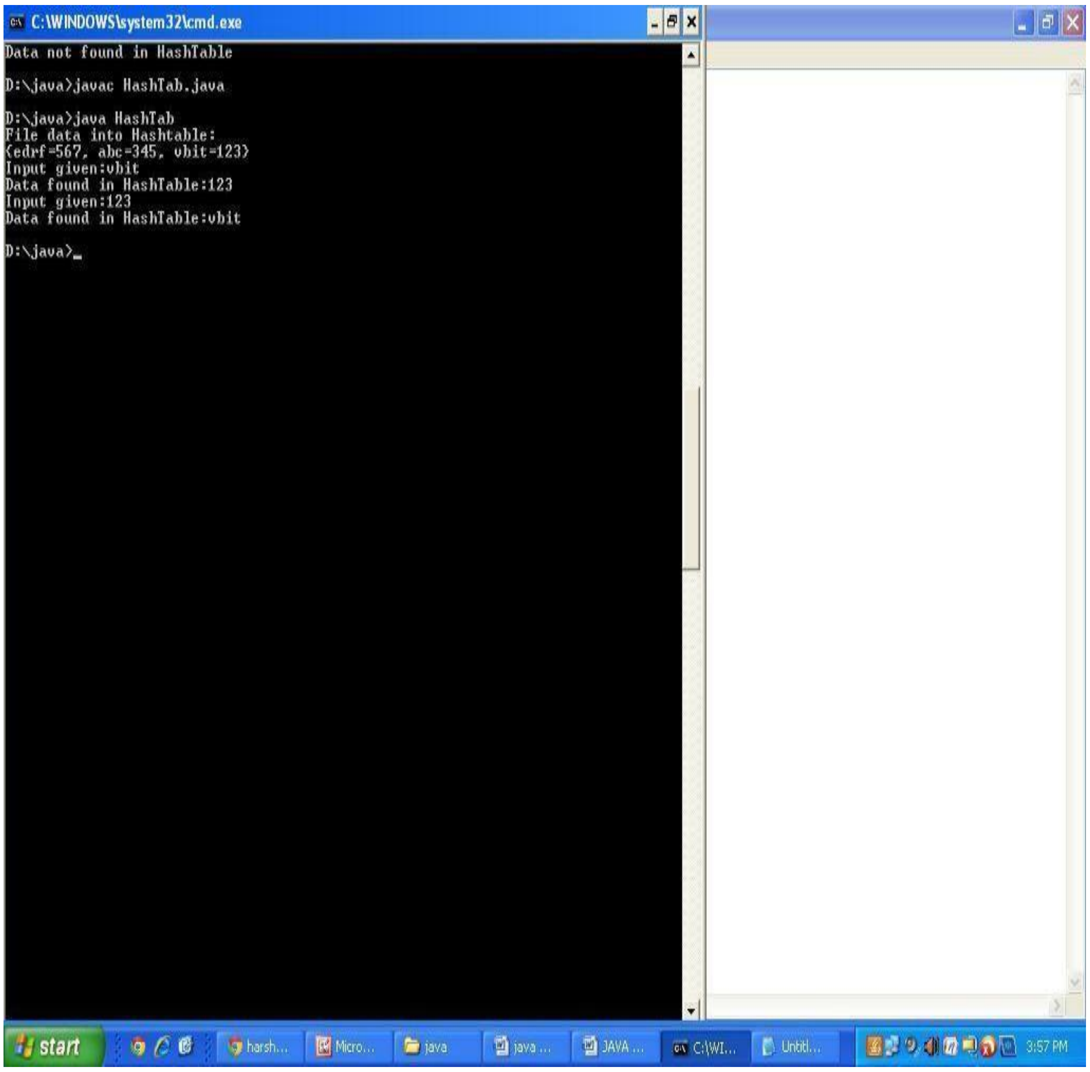
```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.Hashtable;
import java.util.Iterator;
import java.util.Set;
public class HashTab
 {
        public static void main(String[] args)
        {
                HashTab prog11 = new HashTab();
                Hashtable<String, String>hashData = prog11.readFromFile("HashTab.txt");
                System.out.println("File data into Hashtable:\n"+hashData);
                prog11.printTheData(hashData, "vbit");
                 prog11.printTheData(hashData, "123");
                 prog11.printTheData(hashData, " ------------");
        }
        private void printTheData(Hashtable<String, String>hashData, String input)
```

```
    {
                String output = null;
                if(hashData != null)
                {
                    Set<String> keys = hashData.keySet();
                 if(keys.contains(input))
                 {
                            output = hashData.get(input);
                 }
                 else
                 {          Iterator<String> iterator = keys.iterator();
                            while(iterator.hasNext()) {
                                        String key = iterator.next();
                                        String value = hashData.get(key);
                                        if(value.equals(input))
                                        {
                                              output = key;
                                              break;
                                        }
                                   }
                  }
}
                System.out.println("Input given:"+input);
                if(output != null)
                 {
                          System.out.println("Data found in HashTable:"+output);
                  }
                else
                {
                          System.out.println("Data not found in HashTable");
                }
}

private Hashtable<String, String>readFromFile(String fileName)

{

Hashtable<String, String> hashData = new Hashtable<String, String>();


            try
            {     File f = new File("D:\\java\\"+fileName);

                  BufferedReader br = new BufferedReader(new FileReader(f));

                  String line = null;

                  while((line = br.readLine()) != null)

                   {

                      String[] details = line.split("\t");

                     hashData.put(details[0], details[1]);

                   }

            }

           catch (FileNotFoundException e)

           {
             e.printStackTrace();

           }

           catch (IOException e)

           {

             e.printStackTrace();

           }

              return hashData;
          }

}
```

HashTab.txt

abd	123
abc	345
edrf	567

Output:-



```
Data not found in HashTable

D:\java>javac HashTab.java

D:\java>java HashTab
File data into Hashtable:
{edrf=567, abc=345, vbit=123}
Input given:vbit
Data found in HashTable:123
Input given:123
Data found in HashTable:vbit

D:\java>_
```

12. Write a Java program that correctly implements the producer – consumer problem usingthe concept of interthread communication.

```
class Communicate
{
	public static void main(String[] args)throws Exception
	{
		//Producer produces some data which Consumer consumes
		Producer obj1=new Producer();
		//pass Producer object to Consumer so that it is then available to consumer
		Consumer obj2=new Consumer(obj1);
		//create 2 threads and attach to Producer and Consumer
		Thread t1=new Thread(obj1);
		Thread t2=new Thread(obj2);
		//Run the threads
		t1.start(); //Producer starts production t2.start(); //Consumer waits
	}
}
class Producer extends Thread
{
	//to add data, we use string buffer object
	StringBuffer sb;
	//dataprodover will be true when data production is over
	boolean dataprodover = false;
	Producer()

	{
		sb=new StringBuffer(); //allot memory
	}

	public void run()
	{
		synchronized(sb)
		{
			//go on appending data (numbers) to string buffer
			for(int i=1; i<=10; i++)
			{
				try
				{
					sb.append(i+" : ");
					Thread.sleep(100);
					System.out.println("appending");
				}
				catch(Exception e)
```

```
                        {
                        }
                    }
                    //data production is over, so notify to Consumer thread sb.notify();
            }
    }

}
class Consumer extends Thread
{
    //create Producer reference to refer to Producer object from Consumer class
    Producer prod;

    Consumer(Producer prod)
    {
            this.prod=prod;
    }

    public void run()
    {
            //if dta productin is not over, sleep for 10 miliseconds and check
            //again, Here there is a time delay fo several milliseconds to receive data
            /*try
            {
            while(!prod.dataprodover)
            Thread.sleep(10);
            }
            catch(Exception e)
            {
            }*/
            //when data production is over, display data of StringBuffer
            System.out.println(prod.sb);
    }
}
```

output:

```
E:\ >javac Communicate.java

E:\ >java Communicate
appending
appending
appending
appending
appending
appending
appending
appending
appending
appending
```

13. Write a Java program to list all the files in a directory including the files present in all itssubdirectories.

```
import java.io.File;
public class Recursion
{
	static void RecursivePrint(File[] arr,int index,int level)
	{
		// terminate condition
		if(index == arr.length)
			return;
		// tabs for internal levels
		for(int i = 0; i < level; i++)
			System.out.print("\t");
		// for files
		if(arr[index].isFile())
			System.out.println(arr[index].getName());

		// for sub-directories
		else if(arr[index].isDirectory())

		{
			System.out.println("[" + arr[index].getName() + "]");

			// recursion for sub-directories
			RecursivePrint(arr[index].listFiles(), 0, level + 1);
		}

		// recursion for main directory
		RecursivePrint(arr,++ index, level);
	}
	// Driver Method
	public static void main(String[] args)
	{
		// Provide full path for directory(change accordingly)
		String maindirpath = "C:\\Users\\Gaurav Miglani\\Desktop\\Test";
		// File object
		File maindir = new File(maindirpath);
		if(maindir.exists() && maindir.isDirectory())
		 {
			// array for files and sub-directories
			// of directory pointed by maindir
			File arr[] = maindir.listFiles();
			System.out.println("**********************************************");
```

```
                System.out.println("Files from main directory : " + maindir);

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

                // Calling recursive method
                RecursivePrint(arr,0,0);
            }
        }
}
```

Output:

```
Files from main directory : C:\Users\Gaurav Miglani\Desktop\Test

**********************************************

Cormen.pdf
**********************************************


Extra-Items.pdf

XYZ.pdf [Docs]
    A.docx
    B.doc C.docx
ABC.pdf
JKL.pdf
[sheets]
    XXX.csv
    YYY.csv
results.pdf
[Resumes]
    [Before2016]
        Resume2015.doc
        Resume2016.doc
        [Before2014]
            Resume2014.doc
    Resume2017.doc
    Resume2017.pdf
        QA.doc

Testing.pdf
```

14. Write a Java program that implements Quick sort algorithm for sorting a list of names inascending order

```
public class sorting
{
    public static void main(String [] input)

    {
            int k=input.length;
            String temp=new String ();
            String names[]=new String [k+1];
            for(int i=0;i<k;i++)
            {
                    names[i]=input[i];
            }
            for(int i=0;i<k;i++)
                    for(int j=i+1;j<k;j++)
                    {
                            if(names[i].compareTo(names[j])>0)
                            {
                                    temp=names[i]; names[i]=names[j]; names[j]=temp;
                            }
                    }
            System.out.println("Sorted order is");
            for(int i=0;i<k;i++)
            {
                    System.out.println(names[i]);
            }
    }
}
```

Output:

```
Sorted order is
niranjan
pavan
ramreddy
```

15. Write a Java program that implements Bubble sort algorithm for sorting in descendingorder and also shows the number of interchanges occurred for the given set of integers.

```
import java.util.Scanner;
public class Bubble
{
    public static void main(String []args)
    {
            int num, i, j, temp;
            Scanner input = new Scanner(System.in);
            System.out.println("Enter the number of integers to sort:");
            num = input.nextInt();
            int array[] = new int[num];
            System.out.println("Enter " + num + " integers: ");
            for (i = 0; i < num; i++)
                    array[i] = input.nextInt();
            for (i = 0; i < ( num - 1 ); i++)
            {
                    for (j = 0; j < num - i - 1; j++)
                    {
                            if (array [j] < array [j+1])
                            {
                                    temp = array [j];
                                    array [j] = array [j+1];
                                    array [j+1] = temp;
                            }
                    }
            }
            System.out.println("Sorted list of integers:");
            for (i = 0; i < num; i++)
                    System.out.println(array [i]);
    }
}
```

Output:

```
Enter the number of integers to sort:
5
Enter 5 integers:
20
45
99
82
16
Sorted list of integers:
```

99
82
45
20
16