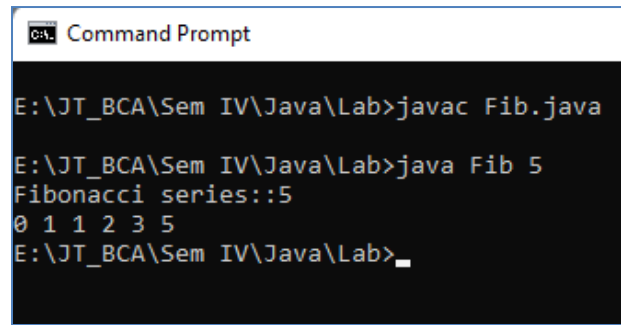


### 1. Display Fibonacci series up to n terms using command line arguments.

```
import java.lang.*;
class Fib
{
    public static void main(String[] args)
    {
        int fib1=0,fib2=1,fib3,n;
        n=Integer.parseInt(args[0]);
        System.out.println("Fibonacci series::"+n);
        System.out.print(fib1+" "+fib2);
        for (int i=2;i<=n ;i++ )
        {
            fib3=fib1+fib2;
            System.out.print(" "+fib3);
            fib1=fib2;
            fib2=fib3;
        }
    }
}
```

#### Output



```
C:\> Command Prompt

E:\JT_BCA\Sem IV\Java\Lab>javac Fib.java

E:\JT_BCA\Sem IV\Java\Lab>java Fib 5
Fibonacci series::5
0 1 1 2 3 5
E:\JT_BCA\Sem IV\Java\Lab>_
```

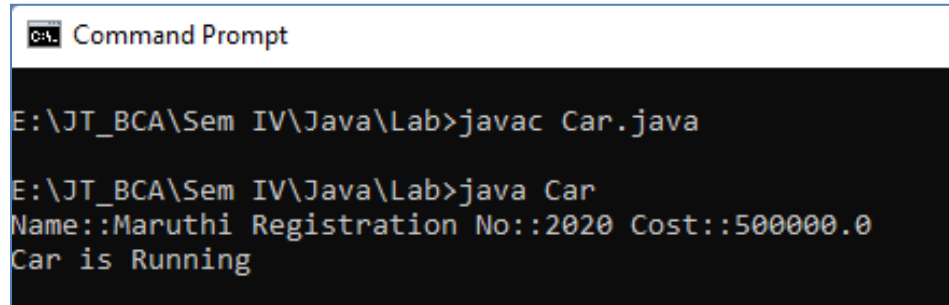
### 2. Demonstrate a single inheritance.

```
class Vehicle
{
    String name;
    int regNo;
    double cost;
    public Vehicle(String name,int regNo,double cost)
    {
        this.name=name;
        this.regNo=regNo;
        this.cost=cost;
    }
    public void run()
    {
        System.out.println("Vehicle is Running");
    }
    public void display()
    {
        System.out.println("Name::"+name+" "+"Registration
No::"+regNo+" "+"Cost::"+cost);
    }
}
class Car extends Vehicle
{
    public Car(String name,int regNo,double cost)
    {
        super(name,regNo,cost);
    }
    public void display()
    {
        super.display();
    }
    public void run()
    {
        System.out.println("Car is Running");
    }

    public static void main(String[] args)
    {
        Car c=new Car("Maruthi",2020,500000);
        c.display();
        c.run();
    }
}
```

```
}  
}
```

Output



```
Command Prompt  
  
E:\JT_BCA\Sem IV\Java\Lab>javac Car.java  
  
E:\JT_BCA\Sem IV\Java\Lab>java Car  
Name::Maruthi Registration No::2020 Cost::500000.0  
Car is Running
```

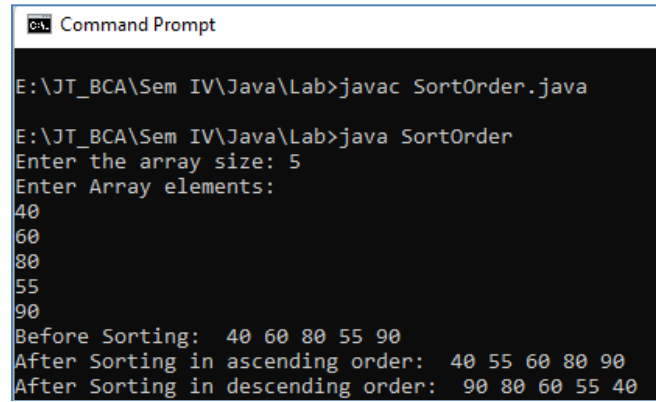
### 3. Sort n elements using an array.

```
import java.util.*;
class SortOrder
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        System.out.print("Enter the array size: ");
        int size=sc.nextInt();
        int arr[]=new int[size];
        System.out.println("Enter Array elements: ");
        for(int i=0;i<size;i++)
        {
            arr[i]=sc.nextInt();
        }
        System.out.print("Before Sorting: ");
        for(int i=0;i<size;i++)
        {
            System.out.print(" "+arr[i]);
        }

        for(int i=0;i<size;i++)
        {
            int temp;
            for(int j=i+1;j<size;j++)
            {
                if(arr[i]>arr[j])
                {
                    temp=arr[i];
                    arr[i]=arr[j];
                    arr[j]=temp;
                }
            }
        }
        System.out.println();
        System.out.print("After Sorting in ascending order: ");
        for(int i=0;i<size;i++)
        {
            System.out.print(" "+arr[i]);
        }
        System.out.println();
        System.out.print("After Sorting in descending order: ");
        for(int i=size-1;i>=0;i--)
        {
            System.out.print(" "+arr[i]);
        }
    }
}
```

```
}  
}
```

### Output

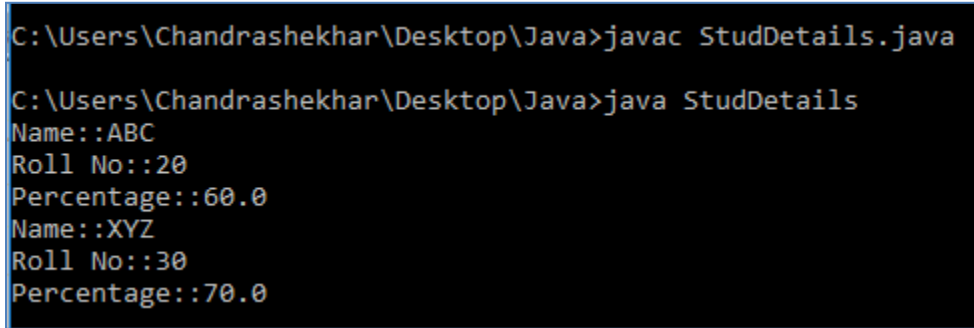


```
Command Prompt  
E:\JT_BCA\Sem IV\Java\Lab>javac SortOrder.java  
E:\JT_BCA\Sem IV\Java\Lab>java SortOrder  
Enter the array size: 5  
Enter Array elements:  
40  
60  
80  
55  
90  
Before Sorting: 40 60 80 55 90  
After Sorting in ascending order: 40 55 60 80 90  
After Sorting in descending order: 90 80 60 55 40
```

### 4 Implement constructor overloading by passing different number of parameter of different types.

```
import java.lang.*;
class StudDetails
{
    String name;
    int roll;
    double per;
    public StudDetails()
    {
        name="ABC";
        roll=20;
        per=60.00;
    }
    public StudDetails(String name,int roll, double per)
    {
        this.name=name;
        this.roll=roll;
        this.per=per;
    }
    public void display()
    {
        System.out.println("Name::"+name+"\n"+"Roll
No::"+roll+"\n"+"Percentage::"+per);
    }
    public static void main(String[] args)
    {
        StudDetails s1=new StudDetails();
        s1.display();
        StudDetails s2=new StudDetails("XYZ", 30,70.00);
        s2.display();
    }
}
```

### Output:

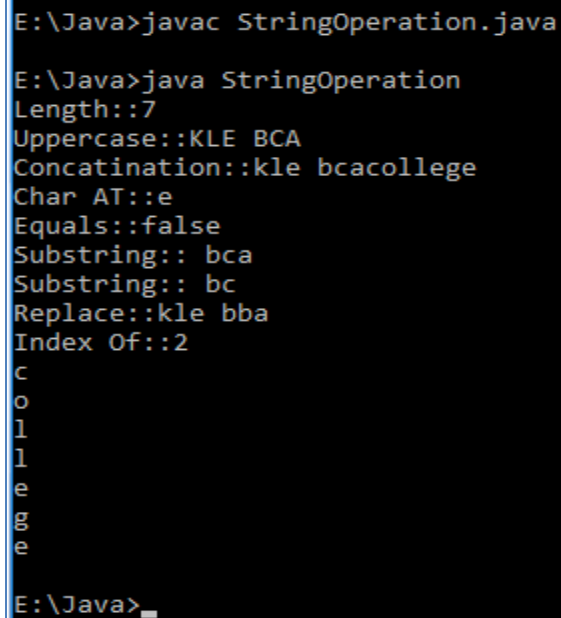


```
C:\Users\Chandrashekhar\Desktop\Java>javac StudDetails.java
C:\Users\Chandrashekhar\Desktop\Java>java StudDetails
Name::ABC
Roll No::20
Percentage::60.0
Name::XYZ
Roll No::30
Percentage::70.0
```

### 5 Demonstrate String methods.

```
import java.lang.*;
class StringOperation
{
    public static void main(String[] args)
    {
        String s1="kle bca";
        String s2="college";
        System.out.println("Length::"+s1.length());
        System.out.println("Uppercase::"+s1.toUpperCase());
        System.out.println("Concatination::"+s1.concat(s2));
        System.out.println("Char AT::"+s1.charAt(2));
        System.out.println("Equals::"+s1.equals(s2));
        System.out.println("Substring::"+s1.substring(3));
        System.out.println("Substring::"+s1.substring(3,6));
        System.out.println("Replace::"+s1.replace('c','b'));
        System.out.println("Index Of::"+s2.indexOf('l'));
        char a[]=s2.toCharArray();
        for (int i=0;i<a.length;i++ )
        {
            System.out.println(a[i]);
        }
    }
}
```

### Output:

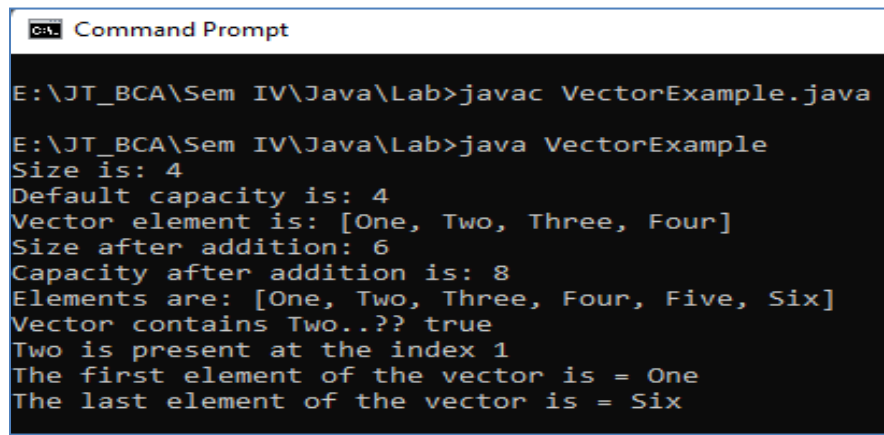


```
E:\Java>javac StringOperation.java
E:\Java>java StringOperation
Length::7
Uppercase::KLE BCA
Concatination::kle bcacollege
Char AT::e
Equals::false
Substring:: bca
Substring:: bc
Replace::kle bba
Index Of::2
c
o
l
l
e
g
e
E:\Java>
```

### 6 Demonstrate vector methods

```
import java.util.*;
public class VectorExample{
    public static void main(String args[]) {
        Vector<String> vec = new Vector<String>(4);
        vec.add("One");
        vec.add("Two");
        vec.add("Three");
        vec.add("Four");
        System.out.println("Size is: "+vec.size());
        System.out.println("Default capacity is: "+vec.capacity());
        System.out.println("Vector element is: "+vec);
        vec.add("Five");
        vec.add("Six");
        System.out.println("Size after addition: "+vec.size());
        System.out.println("Capacity after addition is: "+vec.capacity());
        //Display Vector elements again
        System.out.println("Elements are: "+vec);
        System.out.println("Vector contains Two..?? "+vec.contains("Two"));
        System.out.println("Two is present at the index "+vec.indexOf("Two"));
        //Get the first element
        System.out.println("The first element of the vector is "+vec.firstElement());
        //Get the last element
        System.out.println("The last element of the vector is "+vec.lastElement());
    }
}
```

#### Output:



```
Command Prompt

E:\JT_BCA\Sem IV\Java\Lab>javac VectorExample.java

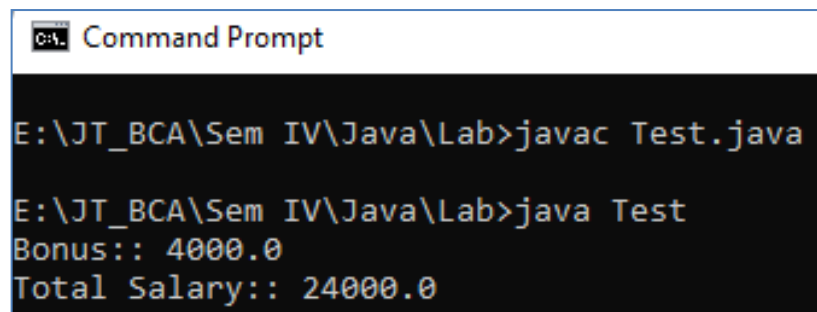
E:\JT_BCA\Sem IV\Java\Lab>java VectorExample
Size is: 4
Default capacity is: 4
Vector element is: [One, Two, Three, Four]
Size after addition: 6
Capacity after addition is: 8
Elements are: [One, Two, Three, Four, Five, Six]
Vector contains Two..?? true
Two is present at the index 1
The first element of the vector is = One
The last element of the vector is = Six
```



### 7 Demonstrate concept of interface

```
interface Dept
{
    int bpay=20000;
    double bonus(double b);
}
class Sales implements Dept
{
    public double bonus(double b)
    {
        System.out.println("Bonus:: "+(bpay*b));
        return(bpay+(bpay*b));
    }
}
class Test
{
    public static void main(String args[])
    {
        Sales s=new Sales();
        System.out.println("Total Salary:: "+(s.bonus(0.2)));
    }
}
```

### Output:



```
C:\ Command Prompt

E:\JT_BCA\Sem IV\Java\Lab>javac Test.java

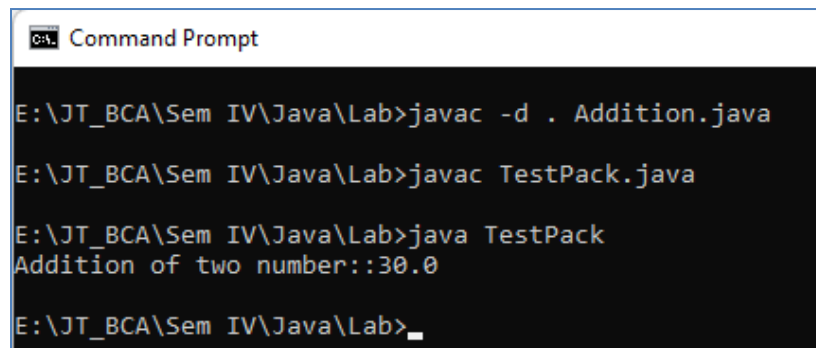
E:\JT_BCA\Sem IV\Java\Lab>java Test
Bonus:: 4000.0
Total Salary:: 24000.0
```

### 8 Demonstrate concept of creating, accessing and using a package.

```
package mypack;
public class Addition
{
    public double a,b;
    public Addition(double a, double b)
    {
        this.a=a;
        this.b=b;
    }
    public double sum()
    {
        return (a+b);
    }
}

import mypack.Addition;
public class TestPack
{
    public static void main(String[] args)
    {
        Addition a=new Addition(10,20);
        System.out.println("Addition of two number::"+a.sum());
    }
}
```

### Output:



```
Command Prompt

E:\JT_BCA\Sem IV\Java\Lab>javac -d . Addition.java

E:\JT_BCA\Sem IV\Java\Lab>javac TestPack.java

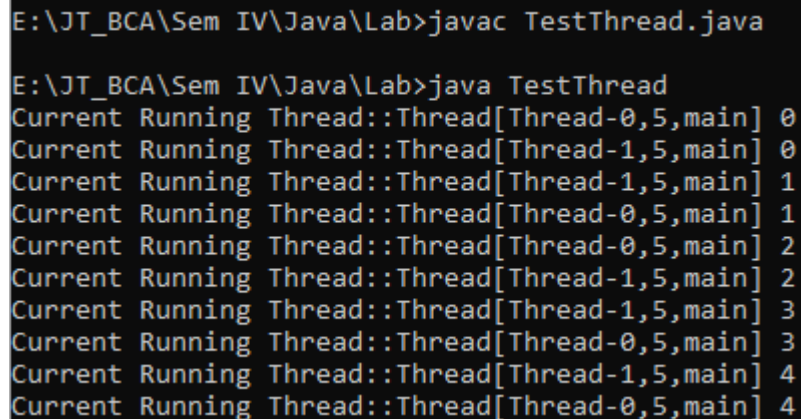
E:\JT_BCA\Sem IV\Java\Lab>java TestPack
Addition of two number::30.0

E:\JT_BCA\Sem IV\Java\Lab>_
```

### 9 Demonstrate multithreaded programming.

```
import java.lang.*;
class MultiThread extends Thread
{
    public void run()
    {
        try
        {
            for (int i=0;i<5 ;i++ )
            {
                Thread.sleep(100);
                System.out.println("Current Running
Thread::"+Thread.currentThread()+" "+i);
            }
        }
        catch(Exception e)
        {
            System.out.println(e.getMessage());
        }
    }
}
class TestThread
{
    public static void main(String[] args)
    {
        MultiThread m1=new MultiThread();
        MultiThread m2=new MultiThread();
        m1.start();
        m2.start();
    }
}
```

**Output:**



```
E:\JT_BCA\Sem IV\Java\Lab>javac TestThread.java
E:\JT_BCA\Sem IV\Java\Lab>java TestThread
Current Running Thread::Thread[Thread-0,5,main] 0
Current Running Thread::Thread[Thread-1,5,main] 0
Current Running Thread::Thread[Thread-1,5,main] 1
Current Running Thread::Thread[Thread-0,5,main] 1
Current Running Thread::Thread[Thread-0,5,main] 2
Current Running Thread::Thread[Thread-1,5,main] 2
Current Running Thread::Thread[Thread-1,5,main] 3
Current Running Thread::Thread[Thread-0,5,main] 3
Current Running Thread::Thread[Thread-1,5,main] 4
Current Running Thread::Thread[Thread-0,5,main] 4
```

### 10 Write a program to implement thread priority

```
class A extends Thread
{
    public void run()
    {
        for(int i=1;i<=3;i++)
        {
            System.out.println("i= "+i);
        }
    }
}
class B extends Thread
{
    public void run()
    {
        for(int j=1;j<=3;j++)
        {
            System.out.println("j= "+j);
        }
    }
}
class C extends Thread
{
    public void run()
    {
        for(int k=1;k<=3;k++)
        {
            System.out.println("k= "+k);
        }
    }
}
class TestThread
{
    public static void main(String arg[])
    {
        A a=new A();
        B b=new B();
        C c=new C();
        c.setPriority(9);
        b.setPriority(2);
        a.setPriority(4);
        a.start();
        b.start();
        c.start();
    }
}
```

### Output:

```
C:\Users\Chandrashekhar\Desktop\Java>javac TestThread.java  
C:\Users\Chandrashekhar\Desktop\Java>java TestThread  
k= 1  
j= 1  
i= 1  
j= 2  
k= 2  
j= 3  
i= 2  
k= 3  
i= 3
```

### 11. Create an applet to draw a human face

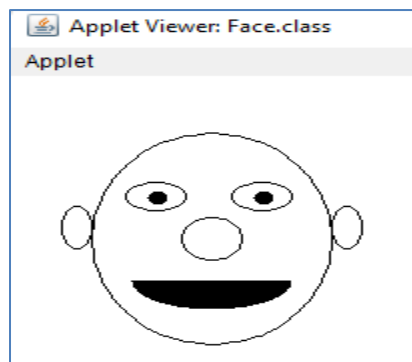
```
import java.applet.*;
import java.awt.*;
public class Face extends Applet
{
    public void paint(Graphics g)
    {
        g.drawOval(40,40,120,150);
        g.drawOval(57,75,30,20);
        g.drawOval(110,75,30,20);
        g.fillOval(68,81,10,10);
        g.fillOval(121,81,10,10);
        g.drawOval(85,100,30,30);
        g.fillArc(60,125,80,40,180,180);
        g.drawOval(25,92,15,30);
        g.drawOval(160,92,15,30);
    }
}
```

#### HTML FILE

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Face Applet</title>
</head>
<body>
<applet code="Face.class" width="500" height="400"> </applet>
</body>
</html>
```

#### Output:

```
javac Face.java
appletviewer Face.html
```



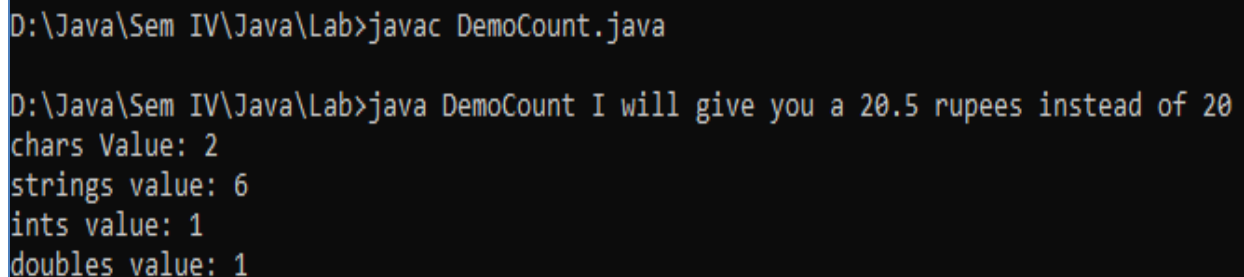
12 Demonstrate simple banner applet

### 13 Program to count number of strings, integers and float values through command line arguments.

```
import java.lang.*;
class DemoCount
{
    public static void main(String[] args)
    {
        int chVal = 0, strVal=0, iVal=0, dVal=0;

        for (int i=0; i<args.length; i++)
        {
            if(args[i].matches("[a-zA-Z]+")){
                if(args[i].length() == 1)
                {
                    chVal = chVal + 1;
                }
                else
                {
                    strVal = strVal + 1;
                }
            }
            if(args[i].matches("[0-9]+"))
            {
                iVal = iVal + 1;
            }
            if(args[i].matches("[+-]?\\d+\\.?(\\d+)?")){
                dVal = dVal + 1;
            }
        }
        dVal = dVal - iVal;
        System.out.println("chars Value: "+ String.valueOf(chVal));
        System.out.println("strings value: "+ String.valueOf(strVal));
        System.out.println("ints value: "+ String.valueOf(iVal));
        System.out.println("doubles value: "+ String.valueOf(dVal));
    }
}
```

#### Output:



```
D:\Java\Sem IV\Java\Lab>javac DemoCount.java

D:\Java\Sem IV\Java\Lab>java DemoCount I will give you a 20.5 rupees instead of 20
chars Value: 2
strings value: 6
ints value: 1
doubles value: 1
```



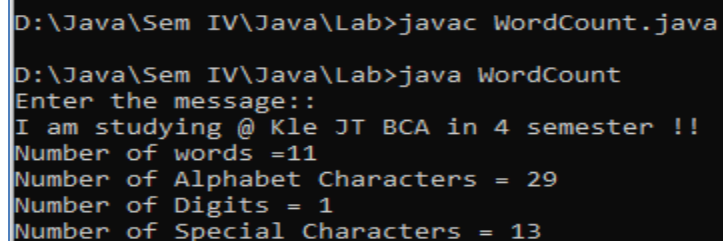
### 14 Program to accept a message from the keyboard and display the no. of words and non alphabetical characters.

```
import java.util.Scanner;
class WordCount
{
    public static void main(String[] args)
    {

        Scanner sc=new Scanner (System.in);
        System.out.println("Enter the message::");
        String line=sc.nextLine();
        char ch;

        int wc=0,alph=0,digi=0,spl=0;
        String words[]=line.split(" ");
        wc=wc+words.length;
        for(int i = 0; i < line.length(); i++)
        {
            ch = line.charAt(i);
            if(ch >= 'a' && ch <= 'z' || ch >= 'A' && ch <= 'Z' )
            {
                alph++;
            }
            else if(ch >= '0' && ch <= '9')
            {
                digi++;
            }
            else
            {
                spl++;
            }
        }
        System.out.println("Number of words =" +wc);
        System.out.println("Number of Alphabet Characters = " + alph);
        System.out.println("Number of Digits = " + digi);
        System.out.println("Number of Special Characters = " + spl);
    }
}
```

### Output



```
D:\Java\Sem IV\Java\Lab>javac WordCount.java
D:\Java\Sem IV\Java\Lab>java WordCount
Enter the message::
I am studying @ Kle JT BCA in 4 semester !!
Number of words =11
Number of Alphabet Characters = 29
Number of Digits = 1
Number of Special Characters = 13
```

### 15 Demonstrate creation of list using an applet.

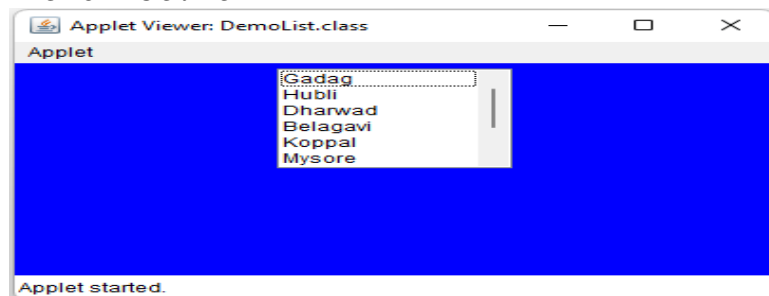
```
import java.applet.Applet;
import java.awt.List;
import java.awt.Color;
public class DemoList extends Applet
{
    public void init() {
        List l = new List(6, true);
        l.add("Gadag");
        l.add("Hubli");
        l.add("Dharwad");
        l.add("Belagavi");
        l.add("Koppal");
        l.add("Mysore");
        l.add("Bangaluru");
        l.add("Haveri");
        l.add("Udupi");
        add(l);
        setBackground(Color.blue);
    }
}
```

#### HTML File

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>List Program</title>
</head>
<body>
    <applet code="DemoList.class" width="400"
height="200"></applet>
</body>
</html>
```

#### Output

```
javac DemoList.java
appletviewer DemoList.html
```



### 16 Demonstrate concept of event handling

```
import java.awt.*;
import java.applet.*;
import java.awt.event.*;
public class KeyBoard extends Applet implements KeyListener
{
    String msg=" ";
    public void init()
    {
        addKeyListener(this);
        requestFocus();
    }
    public void paint(Graphics g)
    {
        g.drawString(msg,100,50);
    }
    public void keyTyped(KeyEvent ke)
    {
        msg=String.valueOf((ke.getKeyChar()));
        repaint();
    }

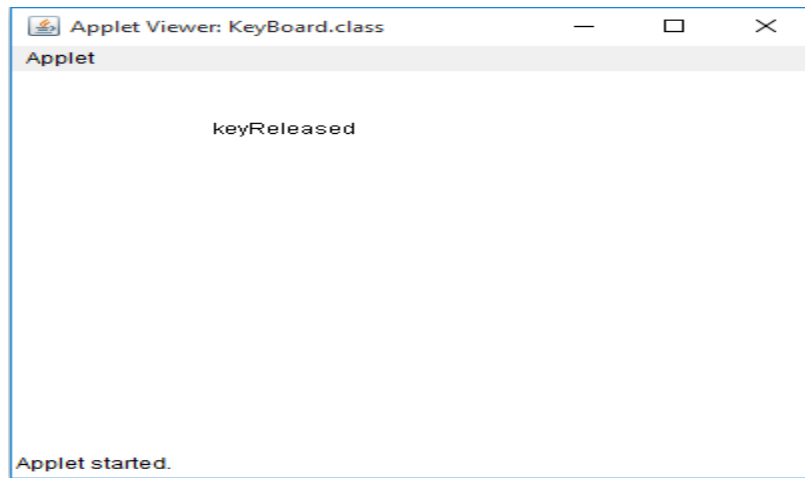
    public void keyPressed(KeyEvent ke)
    {
        msg="Key Pressed";
        repaint();
    }
    public void keyReleased(KeyEvent ke)
    {
        msg="keyReleased";
        repaint();
    }
}
```

#### Html file:

```
<html>
<head>
    <title>Keyboard Event</title>
</head>
<body>
<applet code="KeyBoard.class" width=400 height=300></applet>
</body>
</html>
```

```
javac KeyBoard.java
appletviewer KeyBoard.html
```

Output :



### 17 Program to demonstrate different types of fonts.

```
import java.applet.Applet;
import java.awt.Font;
import java.awt.*;
public class DemoFont extends Applet
{

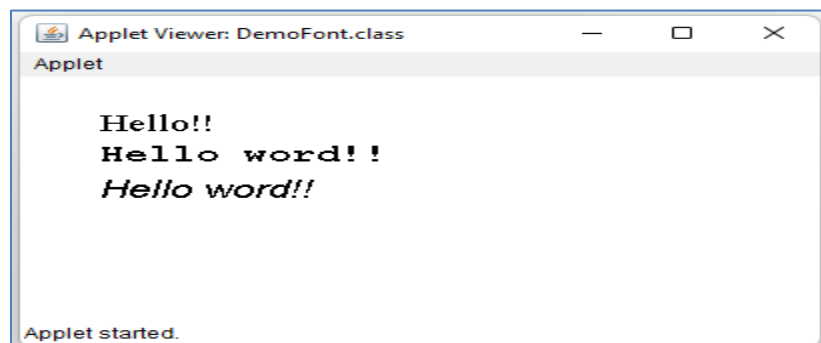
    public void paint(Graphics g)
    {
        Font f1= new Font("SERIF", Font.PLAIN, 20);
        g.setFont(f1);
        g.drawString("Hello!!", 40, 45);
        Font f2= new Font("MONOSPACED", Font.BOLD, 20);
        g.setFont(f2);
        g.drawString("Hello word!!", 40, 70);
        Font f3= new Font("SANS_SERIF", Font.ITALIC, 20);
        g.setFont(f3);
        g.drawString("Hello word!!", 40, 100);
    }
}
```

#### HTML FILE

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>Font Program</title>
</head>
<body>
    <applet code="DemoFont.class" width="400"
height="200"></applet>
</body>
</html>
```

Output:

```
javac DemoFont.java
appletviewer DemoFont.html
```



### 18 Create an applet to tokenize the string.

```
import java.applet.*;
import java.util.StringTokenizer;
import java.awt.*;
public class DemoToken extends Applet
{
    String msg;
    int y=30;
    StringTokenizer st;
    public void init()
    {
        msg=getParameter("abc");
    }
    public void paint(Graphics g)
    {
        st=new StringTokenizer(msg, " ");
        while(st.hasMoreTokens())
        {
            g.drawString(st.nextToken(), 50, y);
            y=y+20;
        }
    }
}
```

### HTML FILE

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>String Token</title>
</head>
<body>
    <applet code="DemoToken.class" width="400" height="300">
        <param name="abc" value="KLE JT BCA">
    </applet>

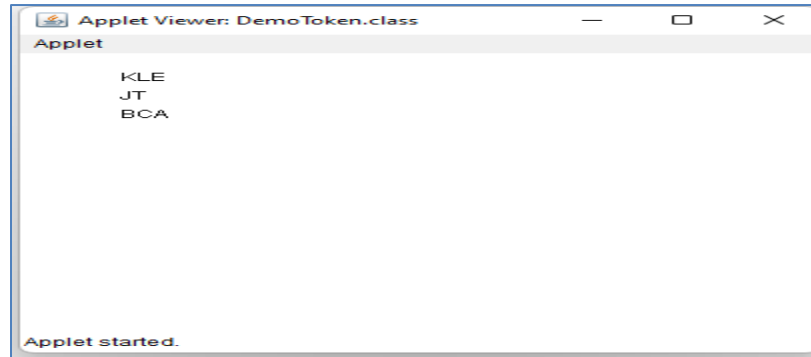
</body>
</html>
```

### Output:

```
javac DemoToken.java
appletviewer DemoToken.html
```

## JAVA PROGRAMMING LAB

---



### 19 Design a simple calculator using java applets

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

public class Calculator extends Applet implements ActionListener
{
    String msg = "";
    TextField t1, t2, t3;
    Button b1, b2, b3, b4, b5;
    Label l1, l2, l3;
    public void init() {
        l1 = new Label("First Number");
        add(l1);
        t1 = new TextField(15);
        add(t1);
        l2 = new Label("Second Number");
        add(l2);
        t2 = new TextField(15);
        add(t2);
        l3 = new Label("Result");
        add(l3);
        t3 = new TextField(15);
        add(t3);
        b1 = new Button("ADD");
        add(b1);
        b1.addActionListener(this);
        b2 = new Button("SUB");
        add(b2);
        b2.addActionListener(this);
        b3 = new Button("MULT");
        add(b3);
        b3.addActionListener(this);
        b4 = new Button("DIV");
        add(b4);
        b4.addActionListener(this);
        b5 = new Button("CLEAR");
        add(b5);
        b5.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == b1) {
            int x = Integer.parseInt(t1.getText());
            int y = Integer.parseInt(t2.getText());
            int sum = x + y;
            t3.setText(" " + sum);
        }
    }
}
```



```
        if (e.getSource() == b2) {
            int x = Integer.parseInt(t1.getText());
            int y = Integer.parseInt(t2.getText());
            int sub = x - y;
            t3.setText(" " + sub);
        }
        if (e.getSource() == b3) {
            int x = Integer.parseInt(t1.getText());
            int y = Integer.parseInt(t2.getText());
            int mul = x * y;
            t3.setText(" " + mul);
        }
        if (e.getSource() == b4) {
            int x = Integer.parseInt(t1.getText());
            int y = Integer.parseInt(t2.getText());
            int div = x / y;
            t3.setText(" " + div);
        }
        if (e.getSource() == b5) {
            t1.setText("");
            t2.setText("");
            t3.setText("");
        }

        repaint();
    }
}
```

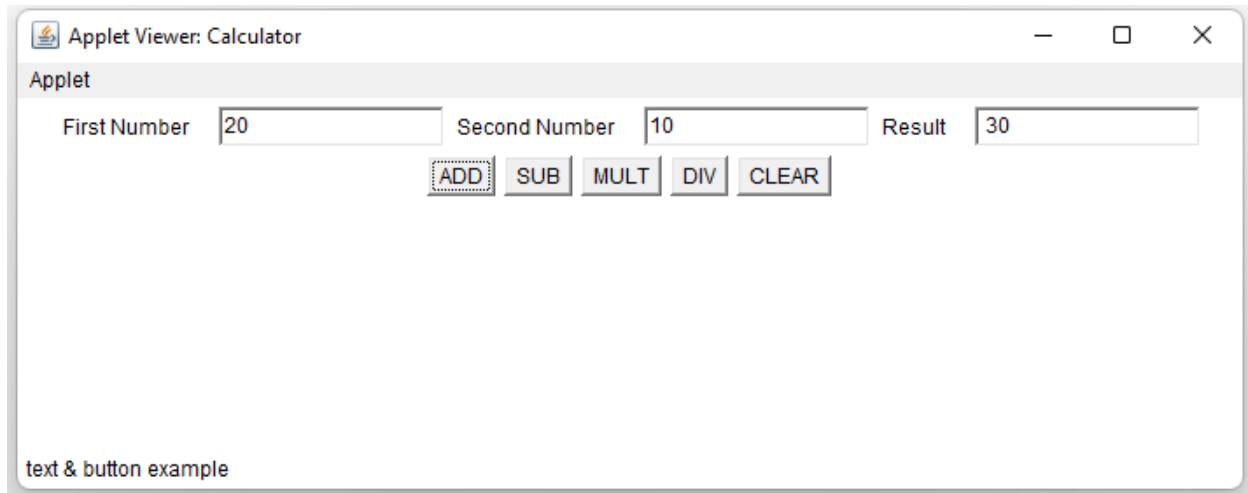
### **HTML FILE**

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1">
    <title>Calculator</title>
</head>
<body>
<applet code="Calculator" width="700" height="200">

</applet>
</body>
</html>
```

### **Output:**

```
javac Calculator.java
appletviewer Calculator.html
```



20 Implement static and dynamic stack using interface using abstract class