Program to demonstrate Arrays

```
 class ArrDemo
         { public static void main(String[]
          args)
           { int a[][] = new
            int[3][3]; int i,j,k=0;

            for(i=0; i<3; i++)
             for(j=0; j<3; j++)
              { a[i][j] = k;
k++; }          for(i=0;
i<3; i++)
              { for(j=0; j<3; j++)
               System.out.print(a[i][j] + "
               "); System.out.println();
              }
            }
         }
```

OUTPUT:

```
0 1 2
3 4 5
6 7 8
```

Program to demonstrate class and objects

```
class Box
{
 double l,w,h;

public class Boxdemo
{ public static void main(String[]
 args)
  {
   Box b1 = new
   Box(); Box b2 =
   new Box(); b1.l=30;
   b1.h=20; b1.w=10;
   b2.l=50; b2.h=30;
   b2.w=40;

   System.out.println("BOX b1 Details:");
      System.out.println("Height" + b1.h);
      System.out.println("Width" + b1.w);
   System.out.println("Length" + b1.l);


   System.out.println("BOX b2 Details");
   System.out.println("Height" + b2.h);
   System.out.println("Width" + b2.w);
   System.out.println("Length" + b2.l);
  }
}
}
```

Output:

```
 BOX b1 Details:
Height 20.0
        Width 10.0
        Length 30.0
        BOX b2 Details
        Height 30.0
        Width 40.0
        Length 50.0
```

Program To Demostrate Class With Methods

```
class Rectangle
{         double          h,w;          void
 setValues(double x, double y)
  {
   h=x;
   w=y;
  }

 void area(double x,double y)
  {
   setValues(x,y);
   System.out.println("Area of rectangle is " + (h*w));
  }

 double perimeter(double x, double y)
  { setValues(x,y);
   return(2*(h+w))
   ;
  }
} public class
Demo
{ public static void main(String[]
 args)
   {
     Rectangle r = new Rectangle();
     r.area(30,20);
     double p = r.perimeter(30,20);
     System.out.println("Perimeter is " + p);
   }
}
```

Output:

```
Area of rectangle is 600.0
Perimeter is 100.0
```

Program to demonstrate Objects as Parameters to Methods

```
class Test
{ int x; void
 increment(Test o)
  {
   o.x=x*10;
  }
}


public class Demo1
{ public static void main(String
 args[])
  {
   Test t = new Test();
   t.x=10;
   Test t1 = new Test();
   t1.x=20;

   System.out.println("t.x=" + t.x);
   System.out.println("t1.x=" + t1.x);
   t.increment(t1);
   System.out.println("After calling increment() t1.x=" + t1.x);
  }
}
```

Output:

```
t. x=10
t1.x=20
After calling increment() t1.x=100
```

Program to demonstrate Constructor Overloading

```
class Student
          {    int    rollno;    String
           name;                double
           m1,m2,m3,avg;

           Student()
            { rollno=1;
             name="xyz"
             ; m1=75;
             m2=80;
             m3=70;
             avg=0;
            }
            Student(int r, String n, double m11, double m12, double m13)
            { rollno=r;
             name=n;
             m1 =
             m11; m2
             = m12;
             m3 =
             m13; avg
             = 0;
            }              double
calculate()
            {
             avg = (m1+m2+m3)/3;
             return(avg);
            }
void details()
            {
             System.out.println("Name: " + name);
             System.out.println("Roll No: " + rollno);
             System.out.println("Average: " + calculate());
            }
          }           public
class Demo2
          { public static void main(String[]
           args)
            {
              Student s1 = new Student();
```

```
    Student s2 = new
    Student(); s1.details();
    s2.details();
   }
}
```

Output:

```
Name: xyz
Roll No: 1
Average: 75.0
Name: xyz
Roll No: 1
Average: 75.0
```

Program to implement Stack operations using Interface

```
interface Stack
{   void   push(int
 i); int pop();
}

class Stack1
 int stk[];
 Stack1()
  { stk = new
   int[10];
  }
}

public class StackDemo extends Stack1 implements Stack
{ int
 tos;
 StackDemo()
  { tos=-
   1;
  }

 public void push(int i)
  {
   if(tos>9)
     System.out.println("Stack is full");
   else
     stk[++tos]=i;
  }

 public int pop()
  {
   if(tos<0)
    {
     System.out.println("Stack is empty");
     return 0;
    }
   else
     return stk[tos--];
  }
```

```
            public static void main(String[] args)
             {
               StackDemo s = new StackDemo();

               for(int i=1; i<=9; i++)

                 s.push(i);
 for(int i=0; i<10; i++)
System.out.println(s.pop());
             }
          }
```

Output:

```
9
8
7
6
5
4
3
2
1
Stack is empty
0
```

Program to demonstrate 'this' keyword

```
class ThisDemo
{ int
 x,y;
 ThisDemo(int x,int y)
  {
   this.x=x;
   this.y=y;
  }

 void display()
  {
   System.out.println("x=" + x);
   System.out.println("y=" + y);
  }
}

public class Demo3
{ public static void main(String[]
 args)
   {
    ThisDemo d = new ThisDemo(1,2);
    d.display();
   }
}
```

Output:

```
x=1 y=2
```

Program to demonstrate 'static' keyword

```
class StatDemo
{   static   int
 x;   int     y;
 static
  {
   x=4;
   System.out.println("x=" + x);
  }

 static void display()
  {
   x=x*2;
   StatDemo d = new StatDemo();
   d.y=10;
   System.out.println("y=" + d.y);
  }
}

class StaticDemo
{ public static void main(String[]
 args)
  {
   StatDemo d1 = new StatDemo();
   d1.y=10;
   System.out.println("d1.y=" + d1.y);
   System.out.println("x=" + StatDemo.x);
   StatDemo.display();
  }
}
```

OUTPUT:

```
x=4
d1.y=10
x=4
y=10
```

Program to demonstrate Command Line Arguments

```
class Demo5
{ public static void main(String[]
 args)
  {
   String str = args[0]; int i =
   Integer.parseInt(args[1]); float f =
   Float.parseFloat(args[2]);
   System.out.println("Adding " + i + " and " + f + " gives " + (f+i));
   System.out.println("Multiplying " + i + " and " + f + " gives " +
   (f*i));
   System.out.println("String " + str + " length is " + str.length());
   System.out.println("For String " + str + " character at e is" +
   str.charAt(3));
  }
}
```

Output:

```
D:\SREE\RECORD>java Demo5 Hello 10 20.1
Adding 10 and 20.1 gives 30.1
Multiplying 10 and 20.1 gives 201.0
String Hello length is 5
For String Hello character at e is l
```

Program to demonstrate inner class

```
class Outer
{ int
 ox;
 Outer(
 )
  { ox =
   10;
  }

 class inner
  { int
   ix;
   inner(
   )
    { ox =
     30; ix =
     40;
    }

   void display()
    {
     System.out.println();
     System.out.println("Inner Class display");
     System.out.println("ox=" + ox);
     System.out.println("ix=" + ix);
    }
   }

 void displayOuter()
  {
   System.out.println("Outer class
   display"); System.out.println("ox=" +
   ox); inner k = new inner(); k.display();
  } public static void main(String[]
 args)
  {
   Outer l = new Outer();
   l.displayOuter();
  }
}
```

Output:

```
Outer class display
ox=10

Inner Class display ox=30 ix=40
```

Program to demonstrate Inheritance

```
class Inher1
{
 int i;
 Inher1()
  { i =
   10;
  }

 Inher1(int i)
  { this.i =
   i;
  }

 void show()
  {
   System.out.println("In Superclass i = " + i);
  }
}

public class Inher2 extends Inher1
{ int k;
 Inher2(
 )
  { i =
   10; k =
   20;
  }

 Inher2(int i, int k)
  { this.i =
   i; this.k =
   k;
  }

 void showK()

  {
   show();
   System.out.println("k=" + k);
  }

 public static void main(String[] args)
  {
   Inher2 i = new Inher2();
   Inher2 i1 = new Inher2(30,40);

   i. showK();
   i1.showK();
  }
}
```

Output:

```
  In Superclass i = 10
  k=20
In Superclass i = 30 k=40
```

Program to demonstrate 'super' keyword

```
class Figure
{ int
 l,b;
 Figure()
  {
   l=10;
   b=5;
   System.out.println("Super class constructor");
  }

 Figure(int i, int j)
  {
   System.out.println("Super class
   Constructor"); l=i; b=j;
  }

 void area()
  {
  System.out.println("There is no area for the figure"); }
}

class Cube extends Figure
{ int h;
 Cube(
 )
  {
   l=5;
   b=7;
   h=9;
  }

 Cube(int l, int b, int h)
  {
   super(l,b);
   this.h=h;
  }

 void area()
```

```
   {
   System.out.println("area=" + (l*b));
   }

  void volume()
   {
     System.out.println("volume=" + (l*b*h));
   }
}
public class SuperDemo extends Figure
{ void
  area()
   {
   System.out.println("Area of rectangle" + (l*b)); }

  void perimeter()
   {
    System.out.println("Perimeter of rectangle" + (2*(l+b)));
   }

  SuperDemo(int l, int b)
   {
    super(l,b);
   }

  public static void main(String[] args)
   {
    Figure f = new Figure(30,40);
    Cube c = new Cube(40,50,60);
    SuperDemo s = new SuperDemo(30,20);
    Figure f1; f.area();
    c.area();
    s.area();
    c.volume();
    s.perimeter();

    f1=f;
    f1.area();
    f1=c;
    f1.area();
   f1=s;
   f1.area();
  }
}
   System.out.println("volume=" + (l*b*h));
  }
}


public class SuperDemo extends Figure
{ void
 area()
  {
  System.out.println("Area of rectangle" + (l*b)); }

 void perimeter()
  {
   System.out.println("Perimeter of rectangle" + (2*(l+b)));
  }

 SuperDemo(int l, int b)
  {
   super(l,b);
  }

 public static void main(String[] args)
  {
   Figure f = new Figure(30,40);
   Cube c = new Cube(40,50,60);
   SuperDemo s = new SuperDemo(30,20);
   Figure f1; f.area();
   c.area();
   s.area();
   c.volume();
   s.perimeter();

   f1=f;
   f1.area();
   f1=c;
   f1.area();
   f1=s;
   f1.area();
  }
}
```

Output:

```
Super class Constructor
Super class Constructor
Super class Constructor
There is no area for the
figure area=2000 Area of
rectangle600
volume=120000
Perimeter of rectangle100
There is no area for the
figure area=2000 Area of
rectangle600
```

Program to demonstrate Abstract class

```
          abstract class Figure
          { int l,b; abstract
           void area(); void
           showlb()
            {
             System.out.println("l=" + l + "b=" + b);
            }
          }

          public class AbstractDemo extends Figure
          { int
           h;
           AbstractDemo(int l,int b, int h)
            {
             this.l=l;
             this.b=b
             ;
             this.h=h
             ;
            }
void area()
            {
            System.out.println("Area of cube "+ (l*b*h));
            }

           public static void main(String[] args)
            {
             AbstractDemo a = new AbstractDemo(10,20,30);
             a.area();
             a.showlbh();
            }           void
showlbh()
            {
             showlb();
             System.out.println("h=" + h);
            }
          }
```

Output:

```
Area of cube
6000 l=10b=20
h=30
```

Program to demonstrate Packages

```
 package mypack;

         class Employee
         { int
          eid;
          String name; double
          hra,da,sal,basic;
          Employee(int id,String n, double hr, double da)
           { eid=id;
            name=n;
            hra=hr;
            basic=3000.50
            ;

            this.da=da;
            sal=50;
           } void
          calculate()
           {
            sal=hra+basic+da;
            System.out.println("empid: " + eid);
            System.out.println("Name: " + name);
            System.out.println("HRA: " + hra);
            System.out.println("DA: " + da);
            System.out.println("Basic: " + basic);
            System.out.println("Sal: " + sal);
           }             public static void
main(String[] argS)
           {
            Employee e = new Employee(1,"abc",30.1,3000);
            e.calculate();
           }
         }
```

Output:

```
empid1
Nameabc
HRA: 30.1
```

```
DA: 3000.0
Basic: 3000.5
Sal: 6030.6
```

Program to demonstrate Interface

```
interface myintf
{   int   x=10;   void
 call();             void
 callMe(int x);
}

public class InterfaceDemo implements myintf
{ public void
 call()
  {
   System.out.println("This is implemented method");
  }

 public void callMe(int x)
  {
   System.out.println("Second method is implemented" + x);
  }

 public static void main(String[] args)
  {
   InterfaceDemo d = new InterfaceDemo();
   d.call();
   d.callMe(4);
   System.out.println("x=" + x);
  }
}
```

Output:

```
This is implemented method
Second method is implemented4
x=10
```

Program to demonstrate Exception Handling

```
class ExcepDemo
{
public static void main(String[] args)
{
int a=0;
int d=30;
try
{
System.out.println("a=" + a +" d=" + d);
d=d/a;
System.out.println("This will not be executed");
}
catch(ArithmeticException e)
{
System.out.println("Exception raised "+ e);
a=5;
d=d/a;
System.out.println("This will be executed");
}
System.out.println("This also will be executed");
}
}
```

Output:

```
a=0 d=30
Exception raised java.lang.ArithmeticException: / by zero
This will be executed
This also will be executed
24
```

Program to demonstrate MultiCatch

```
class Multicatch
{
public static void main(String[] args)
{
int a=args.length;
int d=40;
int c[]={1,2,3,4};
try
{
if(a==0)
d=d/a;
if(a==1)
d=d/(a-a);
if(a==2)
System.out.println("Value of c[7] = " + c[7]);
System.out.println("This will never be executed");
}
catch(ArithmeticException e)
{
System.out.println(e);
}
catch(ArrayIndexOutOfBoundsException e)
{
System.out.println("Array problem" + e);
}
catch(Exception e)
{
System.out.println("All kinds of exceptions " + e);
}
System.out.println("This will be executed");
}
}
```

Output:

```
java.lang.ArithmeticException: / by zero
This will be executed
```

Program to demonstrate 'throws' clause and finally block

```
class ThrowDemo
{
void throwDemometh()throws Exception
{
try
{
System.out.println("This is within ThrowDemo");
throw new NullPointerException("From my code");
}
catch(Exception e)
{
System.out.println(e);
}
}
public static void main(String[] args)
{
ThrowDemo d1 = new ThrowDemo();
try
{
d1.throwDemometh();
}
catch(Exception e)
{
System.out.println("Recaught" + e);
}
finally
{
System.out.println("In finally");
}
}
}
```

OUTPUT:

```
This is within ThrowDemo
java.lang.NullPointerException: From my code In finally
```

Program to demonstrate User Defined Exceptions

```
class BankExcep extends Exception
{
String str;
BankExcep(String str)
{
this.str=str;
}
public String toString()
{
return str+ "Exception raised as amt<500";
}
}
public class UserExcep
{
double amt;
UserExcep()
{
amt=500;
}
void withdraw(double d)throws BankExcep
{
if((amt-d)<500)
throw new BankExcep("In Withdraw");
else
{
amt=amt-d;
System.out.println("Amount Withdrawn is " + d);
}
}
void deposit(double dep)
{
amt=amt+dep;
System.out.println("Amount deposited. Updated amount is
" + amt);
}
public static void main(String[] args)
{
UserExcep d = new UserExcep();
d.deposit(1000);
try
{
d.withdraw(500);
27
d.withdraw(2000);
}
catch(BankExcep e)
{
System.out.println(e);
}
}
}
```

Output:

```
Amount deposited. Updated amount is 1500.0
Amount Withdrawn is 500.0
In WithdrawException raised as amt<500
```

THREADS

Program to demonstrate Threads(Main and Child Threads)

```
class Childth extends Thread
{
String str;
Childth(String str)
  { super(str);
   this.str=str
   ; start();
  }

public void run()
  { for(int i=65; i<91;
   i++)
    {
     System.out.println((char)i);
     try
      {
       Thread.sleep(500);
      }


     catch(InterruptedException e)
      {
       System.out.println(e);
      }
     }
   } public class
MainThread
{ public static void main(String[]
  argS)
    {
     System.out.println("In Main");
     Childth c = new Childth("child");

     for(int i=0; i<25; i++)
      {
       System.out.println("Main" + i);
       try
      {
       Thread.sleep(1000);
      }

     catch(Exception e)
      {
       System.out.println(e);

      }
     System.out.println("Main existing");
    }
  }
}
```

Output:

```
In Main
Main0
A
B
Main existing
Main1
C
D
Main existing
E
Main2
F
Main existing
Main3
G
H
Main existing
Main4
I
J
Main existing
```

Program to demonstrate Thread using Runnable Interface

```
class chrun implements Runnable
{      Thread      t;
 chrun(String str)
  { t=new
   Thread(this,str);
   t.start();
  }

 public void run()
  { for(int i=10; i<36;
   i++)

    {
     System.out.println((char)i + "child");
     try
      {
       Thread.sleep(500); }
     catch(InterruptedException
     e)
      {
       System.out.println(e);
      }
    }
  }
}



public class RunnableDemo
{ public static void main(String[]
 args)
  {
   System.out.println("Entered main");
   chrun c = new chrun("child");

   for(int i=0; i<25; i++)
    {
     System.out.println("Main: " + i);
     try
```

```
      {
       Thread.sleep(1000);
      }

     catch(InterruptedException e)
      {

       System.out.println(e);
      }
     }
   }
}
```

Output:

```
Main: 0

child
♂child
Main: 1
♀child
child
Main: 2
♫child
☼child
Main: 3
►child
◄child
Main: 4
```

Program to demonstrate Join and isalive

```
class Childth implements Runnable
{ int
 x;
 Thread t;
 Childth(String str, int x)
  {
   this.x=x;

   t=new Thread(this,str);
   t.start();
  }

 public void run()
  {
   if(x==1)
   world();
   if(x==2)
   hello();
   if(x==3)
   hai();
  }

 void world()
  {
  try
   { for(int i=0; i<5;
    i++)
     {
      System.out.println("world");
      Thread.sleep(100);
     }
   } catch(InterruptedException

   e)

   {
    System.out.println(e);
   }
  }
 void hello()
  {
  try
   {
    for(int i=0; i<5; i++)
     {
     System.out.println("hello");
     Thread.sleep(100);

     }
   }

  catch(InterruptedException e)
   {
    System.out.println(e);
   }
  }

 void hai()
  {
   try
   { for(int i=0; i<5;
    i++)
     {
     System.out.println("hai");
     Thread.sleep(50);
     }
   } catch(InterruptedException
  e)
   {
    System.out.println(e);
   }
  }
}
public class JoinDemo
{ public static void main(String[]
 argS)
  {
   Childth c1 = new Childth("One",1);
   Childth c2 = new Childth("Two",2);
   Childth c3 = new Childth("Three", 3);
   System.out.println("c1 is alive: " + c1.t.isAlive());
   System.out.println("c2 is alive: " + c2.t.isAlive());
   System.out.println("c3 is alive: " + c3.t.isAlive());

   try
    {
     c1.t.join();
     c2.t.join();
     c3.t.join();
    }

   catch(Exception e)
    {
     System.out.println(e);
    }
    System.out.println("Now c1 is alive: " + c1.t.isAlive());
    System.out.println("Now c2 is alive: " + c2.t.isAlive());
    System.out.println("Now c3 is alive: " + c3.t.isAlive());
  }
}
```

Output:

```
c1 is alive:
true c2 is
alive: true c3
is alive: true
world hello hai
hai world hello
hai hai world
hello
hai
world
hello
world
hello
Now c1 is alive: false
Now c2 is alive: false
Now c3 is alive: false
```

Program to demonstrate Synchronization

```
class Shared
{ synchronized public void call(String
 str)
  {
  System.out.print("[" + str);
  try
   {
     Thread.sleep(1000);
   }
  catch(InterruptedException e)
   {
     System.out.println(e);
   }
  System.out.println("]");
  }
}

class Threads extends Thread
{
 String str;
 Shared q;

 Threads(Shared s,String str)
  {
   this.str = str;
   q=s;
   start();
  }

 public void run()
  {
   q.call(str);
  }
}

class SyncDemo
{ public static void main(String[]
 argS)
```

```
  {
   Shared s = new Shared();
   Threads t1 = new Threads(s,"Hello");
   Threads t2 = new Threads(s,"World");
   try
    {
     t1.join();
     t2.join();
    }

   catch(Exception e)
    {
     System.out.println(e);
    }
  }
}
```

Output:

```
[Hello]
[World]
```

Program to demonstrate Interthread Communication

```
          class Q
          {
          int   x;  boolean   vs   =
          false; synchronized int
          get()
           {
            if(!vs)
             {
              try
              {
               wait();
              }
catch(InterruptedException e)
              {
               System.out.println(e);
              }
             }
            System.out.println("got: " + x);
            notify();
            vs=false;
            return x;
           }          synchronized void
put(int x)
           {
            if(vs
            ) {
            try
              {
               wait();
              } catch(Exception
             e)
              {
              System.out.println(e);
              }
              }

            this.x=x;

            System.out.println("put:x " + x);
            notify();
            vs=true;
           }
          }
           class Consumer extends Thread
          {
           Q      q;
           Consumer
           (Q q)
            {
             this.q=q;
             start();
            }            public
void run()
            {
            while(true)
              {
              q.get();
              }
            }
          }         class Producer
extends Thread
          {
           Q      q;
           Producer
           (Q q) { this.q=q;
            start(); } public
            void run()
            { int i=0;
            while(true
             )
              {
              q.put(i++);
              }
            }
          }


         class InterThread
          { public static void main(String[]
           args)
            {
           Q q = new Q(); Consumer c =
           new Consumer(q);
```

```
  Producer p = new Producer(q);

   try
    {
      c.join();
      p.join(); }
   catch(Exception
   e)
    {
    System.out.println(e);
    }
   }
}
```

Output:

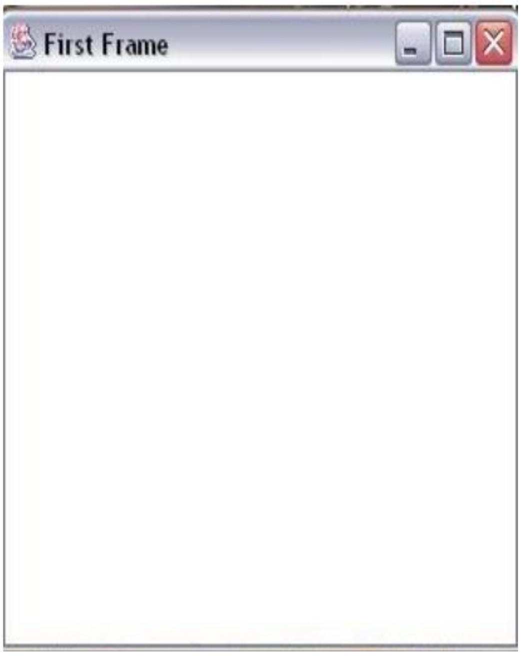
```
got: 527 put:x
528 got: 528
put:x 529
got: 529
put:x 530
got: 530
put:x 531
got: 531
put:x 532
got: 532
put:x 533
got: 533
put:x 534
got: 534
put:x 535
got: 535
put:x 536
got: 536
put:x 537
got: 537
put:x 538
got: 538
put:x 539
got: 539
```

AWT AND EVENT HANDLING

Program to create Frame.

```
import java.awt.*;

        class MyFrame extends Frame
        {
         MyFrame(String str)
          { super(str);
           setSize(300,300)
           ;
           setVisible(true);
          }

         public static void main(String[] args)
          { new MyFrame("First
           Frame");
          }

         public boolean handleEvent(Event e)
          {
            if(e.id==Event.WINDOW_DESTROY)
             {
              System.exit(0);
             } return
            super.handleEvent(e);
          }
        }
```

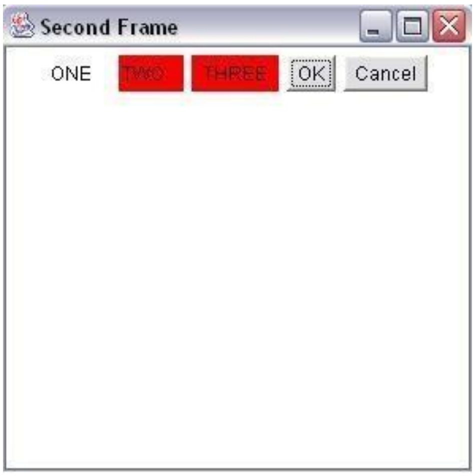
OUTPUT:


First Frame

Program to demonstrate Frame with Labels and Buttons

```
import java.awt.*;

class FrameDemo extends Frame
{
 Label l1,l2,l3; Button
 b1,b2;
 FrameDemo()
  {
   super("Second Frame");
   l1=new Label("ONE");
   l2=new Label("TWO");
   l3=new Label("THREE",Label.CENTER);
   l2.setBackground(Color.red);
   l3.setBackground(Color.red);
   b1=new Button("OK");
   b2=new Button("Cancel");
   setLayout(new FlowLayout());
   add(l1); add(l2); add(l3); add(b1);
  add(b2); setSize(300,300);
  setVisible(true); } public static void
  main(String[] args)
  {
   new FrameDemo();
  }


  public boolean handleEvent(Event e)
  {
   if(e.id==Event.WINDOW_DESTROY)
     {
      System.exit(0);
     } return
   super.handleEvent(e);
  }
}
```

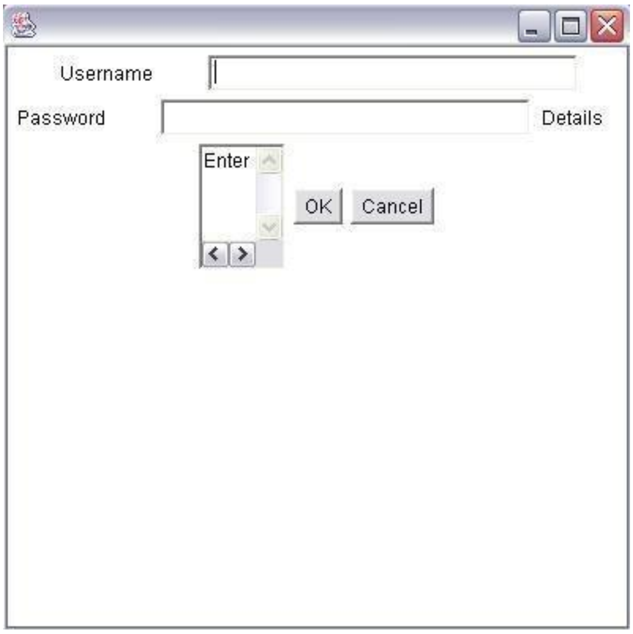
OUTPUT:



Program to demonstrate Frames using button , label, textfield.

```
import java.awt.*; class
FrameDemo5 extends Frame
{
 Label l,u,p,d;
 Button o,c;
 TextField tf1,tf2;
 TextArea ta;
 FrameDemo5()
  {
   u=new                  ");
   Label("Usernam
        "); e p=new
   Label("Password
   d=new
   Label("Details");
   o=new
   Button("OK");
   c=new
   Button("Cancel")
   ; tf1=new
   TextField(30);
   tf2=new
   TextField(30);
   ta=new TextArea("Enter your Address",4,5);
   tf2.setEchoChar('*');
   setLayout(new FlowLayout());

   add(u); add(tf1);
   add(p); add(tf2);
   add(d); add(ta);
   add(o); add(c);
   setSize(400,400)
   ;
   setVisible(true);
  } public static void main(String[]
 args)

   new FrameDemo5();

  {
  } public boolean handleEvent(Event
 e)
  {
    if(e.id==Event.WINDOW_DESTROY)

     {
      System.exit(0);
     } return
   super.handleEvent(e);
  }
}
```

OUTPUT:

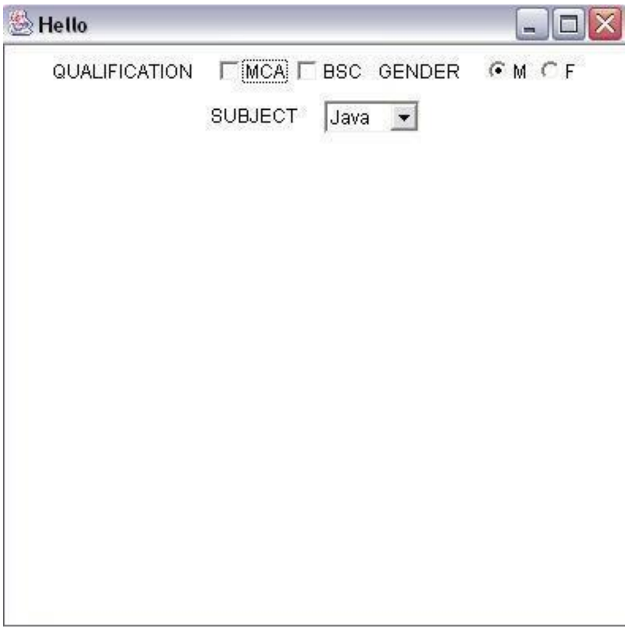


Program to demonstrate Frame with CheckBox and CheckBoxGroup

```
import java.awt.*;

class FrameDemo3 extends Frame
{
 Label l1,l2,l3;
 Checkbox cb1,cb2,cb3,cb4;
 CheckboxGroup cbg;
 Choice c;

 FrameDemo3()
  {
   super("Hello");
   l1=new
   Label("QUALIFICATION");
   l2=new Label("GENDER"); l3=new
   Label("SUBJECT"); cb1=new
   Checkbox("MCA"); cb2=new
   Checkbox("BSC"); cbg=new
   CheckboxGroup(); cb3=new
   Checkbox("M",true,cbg); cb4=new
   Checkbox("F",false,cbg); c=new
   Choice(); c.add("Java");
   c.add("C++");
   c.add("C");
   c.add("Cobol");
   setLayout(new FlowLayout());
   add(l1);
   add(cb1);
   add(cb2);
   add(l2);
   add(cb3);
   add(cb4);
   add(l3); add(c);
   setSize(400,400)
   ;
   setVisible(true);
  }


  {
```

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

   new FrameDemo3();
  }


 public boolean handleEvent(Event e)
  {


   if(e.id==Event.WINDOW_DESTROY)
     {
      System.exit(0);
     } return
   super.handleEvent(e);
  }
}
```

OUTPUT:



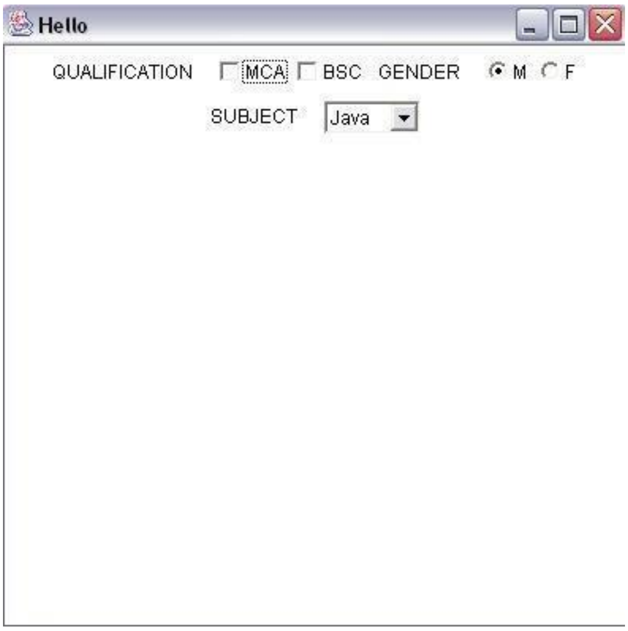
QUALIFICATION □ MCA □ BSC GENDER ● M ○ F

SUBJECT Java ▾

Program to demonstrate Complex Frame(Checkboxes)

```
import java.awt.*; class
FrameDemo4 extends Frame
{
 Label la;
 Checkbox o,w,d,l,s,j,cp,c,p; CheckboxGroup
 cbg1,cbg2;
 FrameDemo4()
  {   super("Hello");   la=new   Label("Skill
   Set");  o=new  Checkbox("O.S");  s=new
   Checkbox("Software");           cbg1=new
   CheckboxGroup();                    w=new
   Checkbox("Windows",true,cbg1); d=new
   Checkbox("Dos",false,cbg1);         l=new
   Checkbox("Linux",false,cbg1);      p=new
   Checkbox("Unix",false,cbg1); cbg2=new
   CheckboxGroup();                     j=new
   Checkbox("Java",true,cbg2);       cp=new
   Checkbox("C++",    false,cbg2);    c=new
   Checkbox("Cobol",false,cbg2);
   setLayout(new FlowLayout());
  add(la); add(o);
  add(w); add(d);
  add(l); add(p);
  add(s); add(j);
  add(cp); add(c);
  setSize(400,400)
  ;
  setVisible(true);
  } public static void main(String[]
 args)
  {
  new FrameDemo4();
  } public boolean handleEvent(Event
 e)

  if(e.id==Event.WINDOW_DESTROY)
   {
    System.exit(0);

  {
     }

    if(e.id==Event.WINDOW_DESTROY)
      {
       System.exit(0);
      } return
    super.handleEvent(e);
   }
}
```

OUTPUT:



QUALIFICATION □ MCA □ BSC GENDER ● M ○ F

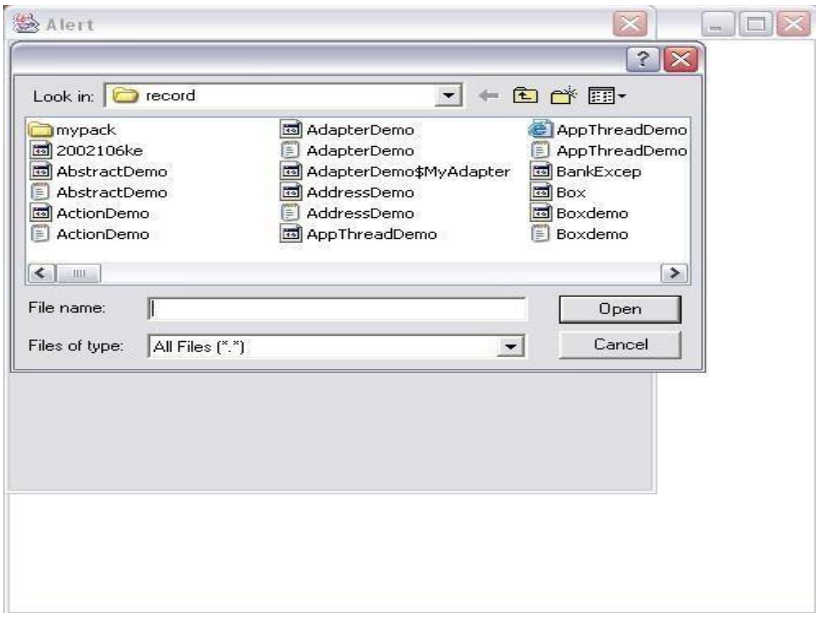
SUBJECT Java

Program to demonstrate DialogBox

```
import java.awt.*;

class DialogDemo extends Dialog
{
 Label l;
 Button b1,b2;
 DialogDemo(Frame f, String title)
  {
   super(f,title);
   l=new Label("Illegal
   Operation"); b1=new
   Button("Ok"); b2=new
   Button("Cancel"); setLayout(new
   FlowLayout());
   add(l); add(b1);
   add(b2);
   setSize(400,400)
   ;
  }
}

class FramesDemo extends Frame
{
 Scrollbar vs,hs;
 FramesDemo()
  {   vs=new   Scrollbar(Scrollbar.VERTICAL);
    hs=new
    Scrollbar(Scrollbar.HORIZONTAL);
    setLayout(new FlowLayout());
   add(vs); add(hs);
   setVisible(true);
   setSize(500,500)
   ;
   try
    {
     Thread.sleep(100);
    }

   catch(Exception e)
```

```
              {
               System.out.println(e);
              }

             DialogDemo d = new DialogDemo(this,"Alert");
             d.setVisible(true); try
              {
               Thread.sleep(1000);
              } catch(Exception
              e)
              {
               System.out.println(e);
              }

           FileDialog fd = new FileDialog(this);
       fd.setVisible(true);
            } public static void
       main(String[] args)
            {
             new FramesDemo();
            }

            public boolean handleEvent(Event e)
             {
             if(e.id==Event.WINDOW_DESTROY)
              {
               System.exit(0);
              } return
             super.handleEvent(e);
             }
             }
```

OUTPUT:



Look in: record

mypack
2002106ke
AbstractDemo
AbstractDemo
ActionDemo
ActionDemo
AdapterDemo
AdapterDemo
AdapterDemo$MyAdapter
AddressDemo
AddressDemo
AppThreadDemo
AppThreadDemo
AppThreadDemo
BankExcep
Box
Boxdemo
Boxdemo

File name:

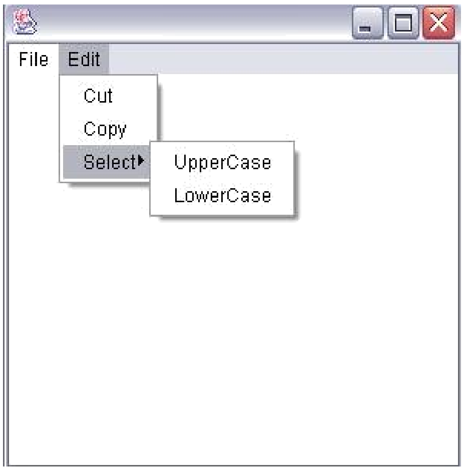
Files of type: All Files (*.*)

Program to demonstrate Menus

```
import java.awt.*;

  class MenuDemo extends Frame
  {
   Menu f,e,s;
   MenuItem o,c,cu,co,u,l;
   CheckboxMenuItem fo;
   MenuBar mb;
   MenuDemo()
    { f=new Menu("File"); e=new
     Menu("Edit"); s=new
     Menu("Select"); o=new
     MenuItem("Open"); c=new
     MenuItem("Close"); cu=new
     MenuItem("Cut"); co=new
     MenuItem("Copy"); u=new
     MenuItem("UpperCase"); l=new
     MenuItem("LowerCase");
     fo=new CheckboxMenuItem("Font");
     f.add(o);
     f.add(c);
     e.add(cu);
     e.add(co);
     s.add(u);
     s.add(l);
     e.add(s);
     mb=new MenuBar();
     mb.add(f);
     mb.add(e);
     setMenuBar(mb);
     setVisible(true);
     setSize(300,300);
    }

   public static void main(String[] argS)
    {
     new MenuDemo();
    }
```

```
 public boolean handleEvent(Event e)
  {
   if(e.id==Event.WINDOW_DESTROY)
     {
      System.exit(0);
     } return
   super.handleEvent(e);
  }
}
```

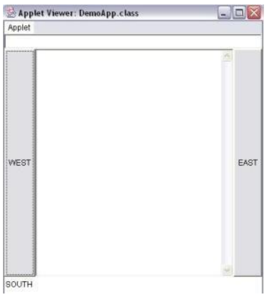
<u>OUTPUT:</u>



Program to demonstrate Border Layout

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

public class DemoApp extends Applet
{
 Button b1,b2;
 Label l1;
 TextField tf;
 TextArea ta;

 public void init()
  {
   b1=new Button("WEST");
   b2=new Button("EAST");
   l1=new Label("SOUTH");
   tf=new TextField(20); ta=new
   TextArea(5,4); setLayout(new
   BorderLayout());
  }

 public void start()
  {
   add(b1,BorderLayout.WEST);
   add(b2,BorderLayout.EAST);
   add(l1,BorderLayout.SOUTH);
   add(tf,BorderLayout.NORTH);
   add(ta);
  }
}
   <HTML>
<applet code="DemoApp.class" height=400
width=400>
</applet>
</HTML>
```

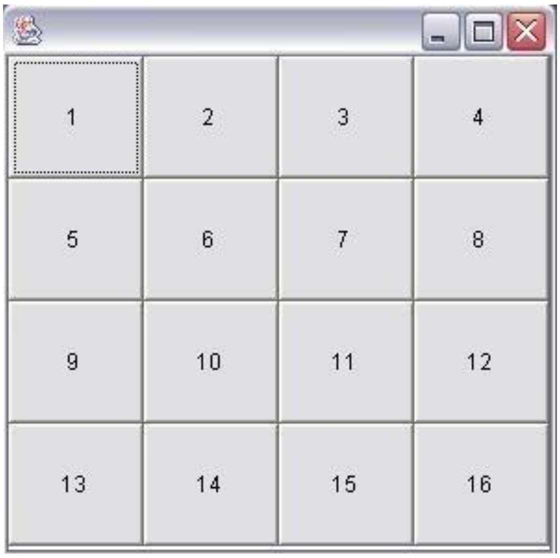
OUTPUT



Program to demonstrate Grid Layout

```
import java.awt.*;

class GridDemo extends Frame
{
 Button b[];
 GridDemo()
  {          b=new          Button[16];
   setLayout(new
   GridLayout(4,4));

   for(int i=0; i<16; i++)
    { b[i]=new Button((i+1)+"
     ");
     add(b[i]);
    }

   setVisible(true);
   setSize(300,300);
  }

 public static void main(String[] args)
  {
   new GridDemo();
  }

 public boolean handleEvent(Event e)
  {
   if(e.id==Event.WINDOW_DESTROY)
    {
     System.exit(0);
    } return
   super.handleEvent(e);
  }
}
```

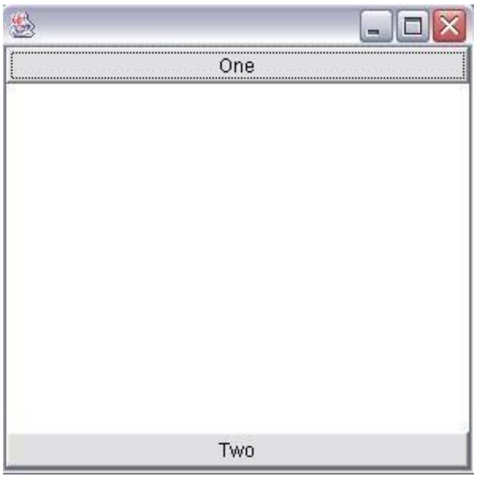
OUTPUT:



Program to demonstrate Implementation of Action Listener

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

class ActionDemo extends Frame
implements ActionListener {
Button b1,b2;
 Label l;
 ActionDemo()
  {     b1=new     Button("One");
   b2=new Button("Two"); l=new
   Label("                        ");
   add(b1,BorderLayout.NORTH)
   ;
   add(b2,BorderLayout.SOUTH)
   ;
   add(l); setSize(300,300);
   setVisible(true);
   b1.addActionListener(this)
   ;
   b2.addActionListener(this)
   ;
  }

 public void actionPerformed(ActionEvent ae)
  {
   if(ae.getSource()==b1)
    {
     l.setText("One");
    }

   if(ae.getSource()==b2)
    {
     l.setText("Two");
    }
  }

 public static void main(String[] argS)
  {
   new ActionDemo();
  }

 public boolean handleEvent(Event e)
  {
   if(e.id==Event.WINDOW_DESTROY)
     {
      System.exit(0);
     }

   return super.handleEvent(e);
  }
}
```

OUTPUT:

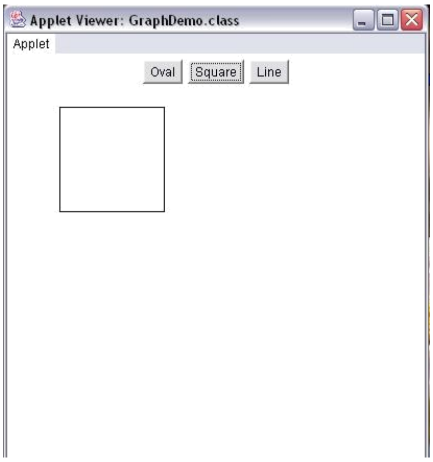


Program to graphs using Action Listener

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

public class GraphDemo extends Applet
implements ActionListener {
 Button o,s,l;
 int k;

 public void start()
  { o=new
   Button("Oval");

   s=new Button("Square");
   l=new Button("Line");
   add(o);
   add(s);
   add(l);
   k=0;
   o.addActionListener(this);
   s.addActionListener(this);
   l.addActionListener(this);
  }

 public void actionPerformed(ActionEvent e)
  {
   if(e.getSource()==o)
    { k=1;
     repaint()
     ;
    }

   if(e.getSource()==s)
    { k=2;
     repaint()
     ;
    }

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

```
    { k=3;
     repaint()
     ; }
  } public void paint(Graphics
 g)
  {
   if(k==1)
   g.drawOval(10,20,50,60);
   if(k==2)
   g.drawRect(50,50,100,100);
   if(k==3)
   g.drawLine(80,20,160,90);
  }
}
```

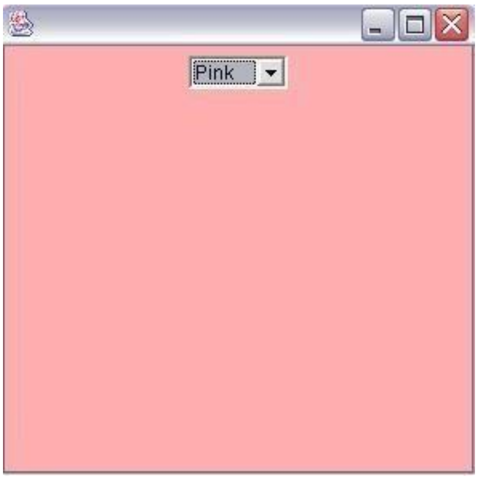
OUTPUT:



Program to demonstrate implementation of ItemListener

```
import java.awt.*; import
java.awt.event.*; public class
ItemDemo extends Frame
implements ItemListener
{
 Choice c;
 Panel p;
 ItemDemo()
  { c=new
   Choice();
   c.add("Red");
   c.add("Green");
   c.add("Blue");
   c.add("Pink");
   p=new Panel();
   setLayout(new
   GridLayout(1,2)); p.add(c);
   add(p);
   c.addItemListener(this);
   setVisible(true);
   setSize(300,300);
  } public void itemStateChanged(ItemEvent
 e)
  {
   if(c.getSelectedItem().equals("Red"))
   p.setBackground(Color.red);
   if(c.getSelectedItem().equals("Green"))
   p.setBackground(Color.green);
   if(c.getSelectedItem().equals("Blue"))
   p.setBackground(Color.blue);
   if(c.getSelectedItem().equals("Pink"))
   p.setBackground(Color.pink);
  } public static void main(String[]
 args)
  {
   new ItemDemo();
  } public boolean handleEvent(Event
 e)
  {
    if(e.id==Event.WINDOW_DESTROY)
     {
      System.exit(0);
     }

   return super.handleEvent(e);
  }
}
```

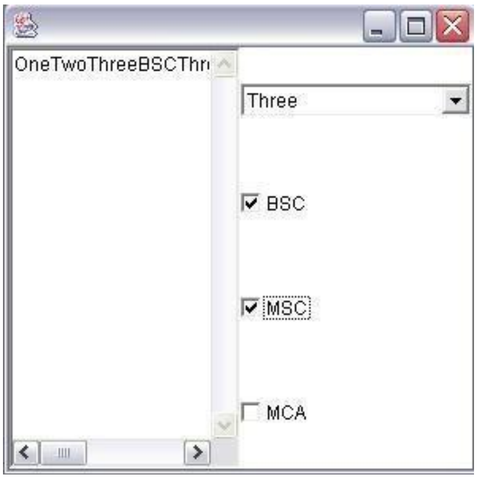
OUTPUT:



Program to show the use of Item Listener

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

public class ItemDemo2 extends Frame
implements ItemListener {
 Choice c;
 Checkbox bsc,msc,mca;
 TextArea ta;
 Panel p;
 ItemDemo2()
  { setLayout(new
   GridLayout(1,2)); c=new
   Choice(); c.add("One");
   c.add("Two");
   c.add("Three");         bsc=new
   Checkbox("BSC");   msc=new
   Checkbox("MSC");  mca=new
   Checkbox("MCA");     p=new
   Panel();
   p.setLayout(new GridLayout(4,1));
   p.add(c);
   p.add(bsc);
   p.add(msc);
   p.add(mca); ta=new
   TextArea(15,30);
   add(ta);
   add(p);
   c.addItemListener(this);
   bsc.addItemListener(this);
   msc.addItemListener(this);
   mca.addItemListener(this);
   setVisible(true);
   setSize(300,300);
  } public void itemStateChanged(ItemEvent
 e)
  { if(e.getSource()==c)
   ta.append(c.getSelectedItem());
   if(c.getSelectedItem().equals("One"))
   ; ta.append("One");
   if(c.getSelectedItem().equals("Two"));
   ta.append("Two");
   if(c.getSelectedItem().equals("Threee"));
   ta.append("Three"); if(bsc.getState())


   ta.append("BSC");
   if(msc.getState())
   ta.append("MSC");
   if(mca.getState())
   ta.append("MCA");
  }

 public static void main(String[] args)
  {
   new ItemDemo2();
  }

 public boolean handleEvent(Event e)
  {
   if(e.id==Event.WINDOW_DESTROY)
     {
      System.exit(0);
     } return
   super.handleEvent(e);
  }
}
```

OUTPUT:



```
OneTwoThreeBSCThr
```

Three

☑ BSC

☑ MSC

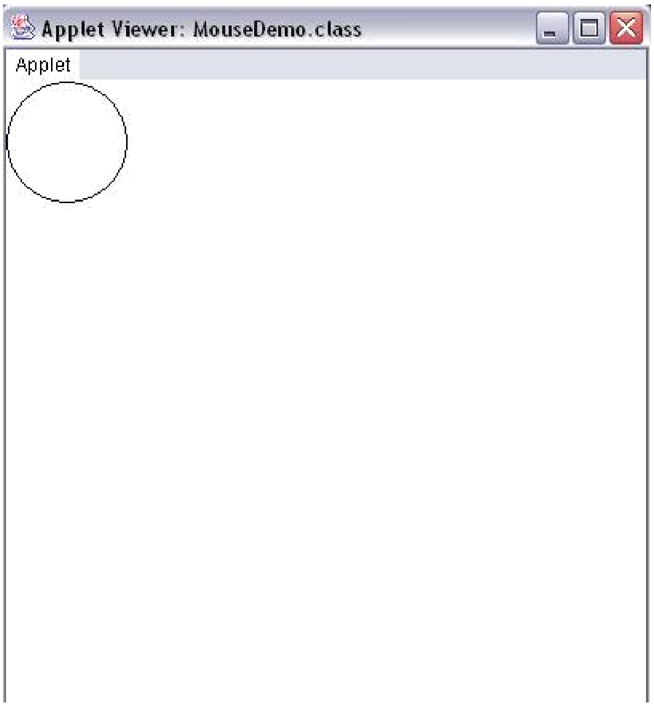
☐ MCA

Program to demonstrate implementation of MouseListener

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

public class MouseDemo extends
Applet implements MouseListener { int
x,y;
 public void start()
  { x=0; y=0;
   addMouseListener(this)
   ;
  } public void mouseClicked(MouseEvent
 e)
  {
   x=e.getX();
   y=e.getY();
   repaint();
  }

 public void mousePressed(MouseEvent e)
  {}

 public void mouseReleased(MouseEvent e)
  {}

 public void mouseEntered(MouseEvent e)
  {}

 public void mouseExited(MouseEvent e)
  {}

 public void paint(Graphics g)
  {
   g.drawOval(x,y,50,50);
  }
}
<HTML>
<applet code="MouseDemo.class" height=400 width=400>
</applet>
```

</HTML>

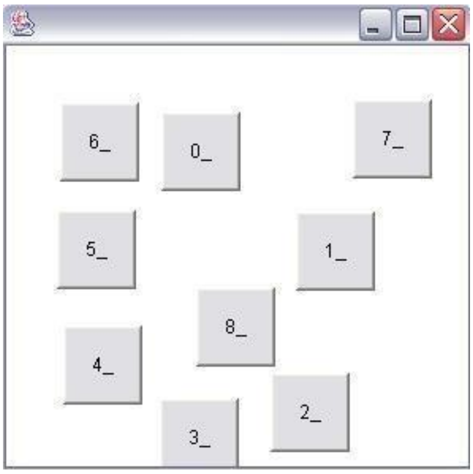
OUTPUT:



Program to show the use of MouseListener and increment the value of the button.

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

        public class MouseDemo1 extends
        Frame implements MouseListener { int
        x,y,i=0;
         MouseDemo1()
          { x=0; y=0;
           addMouseListener(this)
           ; setLayout(null);
           setVisible(true);
           setSize(300,300);
          }

         public void mouseClicked(MouseEvent e)
          {
           Button b1=new
           Button((i++)+"_"); x=e.getX();
           y=e.getY();
           b1.setBounds(x,y,50,50); add(b1);
           validate();
          }           public void
mousePressed(MouseEvent e)
          {}            public void
mouseReleased(MouseEvent e)
          {}

         public void mouseEntered(MouseEvent e)
          {}

         public void mouseExited(MouseEvent e)
          {}            public static void
main(String[] args)
          {
            new MouseDemo1();
          }
         public boolean handleEvent(Event e)
          {
           if(e.id==Event.WINDOW_DESTROY)
        {
         System.exit(0);
        } return
      super.handleEvent(e);
     }
   }
```

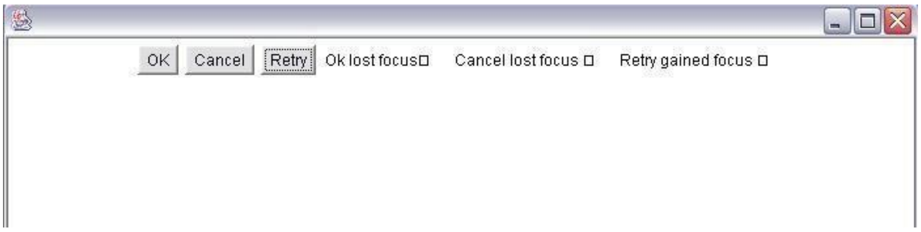
OUTPUT:



Program to demonstrate FocusListener and WindowListener

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

class FocusDemo extends Frame implements
FocusListener,WindowListener {
 Button o,c,r;
 Label l1,l2,l3;
 FocusDemo()
  {                   setLayout(new
   FlowLayout());           o=new
   Button("OK");             c=new
   Button("Cancel");         r=new
   Button("Retry");         l1=new
   Label("Label1");         l2=new
   Label("Label2");         l3=new
   Label("Label3");         add(o);
   add(c);     add(r);     add(l1);
   add(l2); add(l3);
  o.addFocusListener(this);
  c.addFocusListener(this);
  r.addFocusListener(this);
  addWindowListener(this);
  setVisible(true);
  setSize(700,700);
 } public void focusGained(FocusEvent
 fe)
  {  if(fe.getSource()==o)  l1.setText("Ok
   gained                           focus\n");
   if(fe.getSource()==c)
   l2.setText("Cancel gained focus \n");
  if(fe.getSource()==r)
  l3.setText("Retry gained focus \n");
  }

 public void focusLost(FocusEvent fe)
  { if(fe.getSource()==o)
  l1.setText("Ok lost
  focus\n");
  if(fe.getSource()==c)
  l2.setText("Cancel lost focus
  \n"); if(fe.getSource()==r)
    l3.setText("Retry lost focus\n");
 }

public void windowClosing(WindowEvent we)
 {
  setVisible(false);
  System.exit(0);
 }

public void windowClosed(WindowEvent we) {}
public void windowOpened(WindowEvent we) {}
public void windowActivated(WindowEvent we) {}
public void windowDeactivated(WindowEvent we) {}
public void windowIconified(WindowEvent we) {}
public void windowDeiconified(WindowEvent we) {}

public static void main(String[] argS)
 {
  new FocusDemo();
 } public boolean handleEvent(Event

e)

 {
  if(e.id==Event.WINDOW_DESTROY)
    {
     System.exit(0);
    } return
  super.handleEvent(e);
 }
}
```

OUTPUT:



```
Ok lost focus□    Cancel lost focus □    Retry gained focus □
```

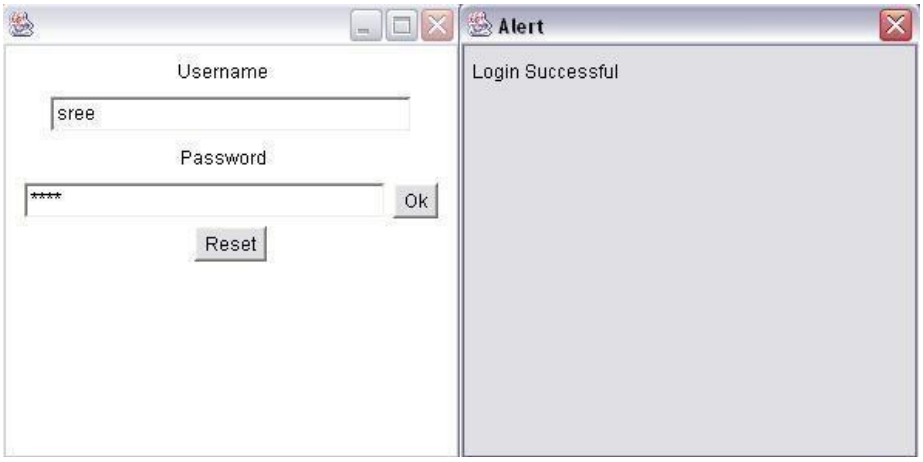
Program to demonstrate Adapter class

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

class AdapterDemo extends Frame
implements ActionListener {
Label u,p;
 Button       o,c;
 TextField tf1,tf2;
 dia d;

 AdapterDemo()
  { u=new Label("Username"); p=new
   Label("Password"); tf1=new TextField(30);
   tf2=new TextField(30); o=new Button("Ok");
   c=new Button("Reset"); tf2.setEchoChar('*');
   setLayout(new FlowLayout()); add(u); add(tf1);
   add(p); add(tf2); add(o); add(c); setVisible(true);
   setSize(300,300);
   o.addActionListener(this);
   c.addActionListener(this);
   addWindowListener(new MyAdapter());
  }

 class MyAdapter extends WindowAdapter
  { public void windowClosing(WindowEvent
   we)
    {
     setVisible(false);
     System.exit(0);
    }
  }

 public void actionPerformed(ActionEvent ae)
  {
   if(ae.getSource()==o)
   {
     if(tf1.getText().equals(tf2.getText())
     ) d=new dia(this,"Login
```

```
      Successful"); else d=new
      dia(this,"Login Denied");
     }
    if(ae.getSource()==c)
     { tf1.setText("
      ");
      tf2.setText("
      ");
     }
   }

  public static void main(String[] args)
   {
    new AdapterDemo();
   } public boolean handleEvent(Event

  e)

   {
    if(e.id==Event.WINDOW_DESTROY)
      {
       System.exit(0);
      } return
    super.handleEvent(e);
   }
}

class dia extends Dialog
{  Label  l;  dia(Frame
  f,String s)
   {
    super(f,"Alert"); l=new
    Label(s); setLayout(new
    FlowLayout());
    add(l);
    setVisible(true);
    setSize(300,300);
    addWindowListener(new winadp());
   }

  class winadp extends WindowAdapter
```

```
        { public void windowClosing(WindowEvent
         we)
          {
           setVisible(false);
            return;
          }
        }
    }
```

OUTPUT



Username

sree

Password

****

Login Successful

I/O STREAMS

Program to check whether a file exits or not

```
import java.io.*;

      class FileDemo
      { public static void main(String[] args) throws
       Exception
        {
         DataInputStream dis = new DataInputStream(System.in);
         System.out.println("Enter the name of the file");
         String s = dis.readLine();

         File f1 = new File(s);
         if(f1.exists())
         System.out.println("File Exits");
         else
          System.out.println("File does not exist"); }
      }
```

OUTPUT:

```
Enter the name of the file
Demo.java
File Exits
```

Program to read a string using Buffered Reader

```
import java.io.*;

      class BufferDemo
      { public static void main(String args[]) throws
       Exception
        {
         BufferedReader br = new BufferedReader(new
         InputStreamReader(System.in));
         System.out.println("Enter a string");
         String str = br.readLine();
         System.out.println("You entered " + str);
        }
      }
```

OUTPUT:

```
Enter a string
Loyola
You entered loyola
```

Program to demonstrate Serialization

```
 import java.io.*;
      class SerDemo implements Serializable
         { int
          x,y;
          SerDemo(int x,int y)
           {
            this.x=x;
            this.y=y;
           }             public
void display()
           {
            System.out.println("x= " + x);
            System.out.println("y= " + y);
           }
         }           class
SerialDemo
         { public static void main(String[] args) throws
          Exception
           {
            FileOutputStream fos = new FileOutputStream("a.txt");
            ObjectOutputStream oos = new ObjectOutputStream(fos);
            SerDemo s1 = new SerDemo(2,3);
            oos.writeObject(s1);
            oos.flush();
            oos.close();
            FileInputStream fis = new FileInputStream("a.txt");
            ObjectInputStream ois = new ObjectInputStream(fis);
            SerDemo ret = (SerDemo)ois.readObject();
            s1.display();
            ret.display();
           }
         }
```

OUTPUT:

```
x= 2
y= 3
x= 2
y= 3
```

Program to demonstrate Socket programming

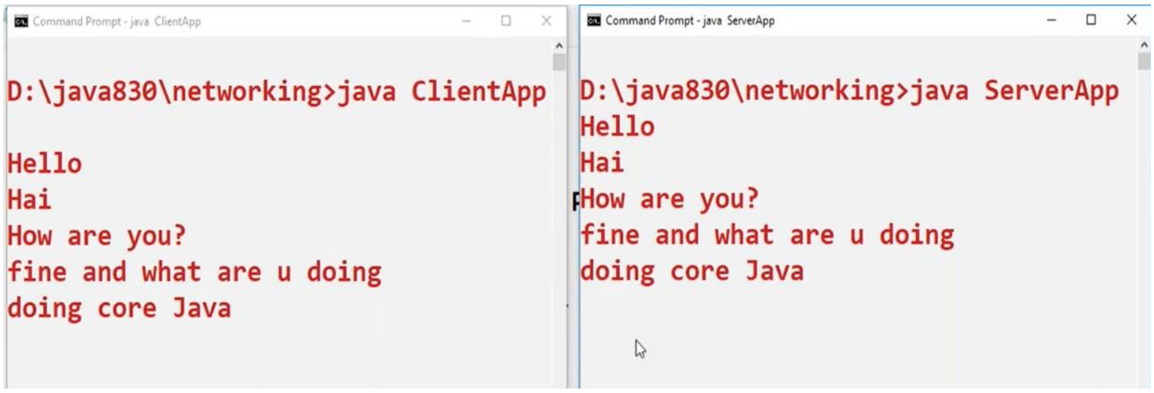
Client side programming:

```
import java.io.*;
import java.net.*;
public class ClientApp {

    public static void main(String[] args) throws Exception {
    Socket s=new Socket("localhost",4444);
    OutputStream os=s.getOutputStream();
    PrintStream ps=new PrintStream(os);
    BufferedReader br1=new BufferedReader(new 
InputStreamReader(System.in));
    InputStream is=s.getInputStream();
    BufferedReader br2=new BufferedReader(new 
InputStreamReader(is));

    while(true) {
        String data1=br1.readLine();
        ps.println(data1);
        String data2=br2.readLine();
        System.out.println(data2);

        if(data1.equals("bye")&&data2.equals("bye")) {
            System.exit(0);
            }
        }
    }
}
```

ServerSide programming

```
import java.io.*;
import java.net.*;
public class ServerApp {

        public static void main(String[] args) throws Exception {
                ServerSocket ss=new ServerSocket(4444);
                Socket s=ss.accept();
                InputStream is=s.getInputStream();
                BufferedReader br1=new BufferedReader(new
InputStreamReader(is));

                OutputStream os=s.getOutputStream();
                PrintStream ps=new PrintStream(os);
                BufferedReader br2=new BufferedReader(new
InputStreamReader(System.in));

                while(true) {
                        String data1=br1.readLine();
                        System.out.println(data1);
                        String data2=br2.readLine();
                        ps.println(data2);

                        if(data1.equals("bye")&&data2.equals("bye")) {
                                System.exit(0);
                        }
                }
        }

}
```

OUTPUT:


Command Prompt - java ClientApp
Command Prompt - java ServerApp

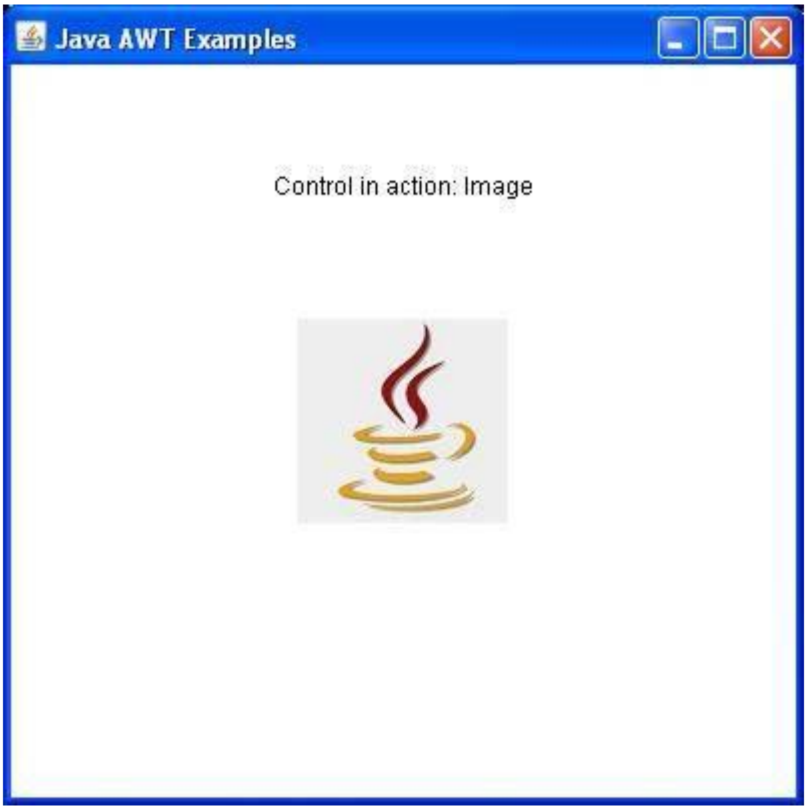
```
D:\java830\networking>java ClientApp

Hello
Hai
How are you?
fine and what are u doing
doing core Java
```

```
D:\java830\networking>java ServerApp
Hello
Hai
How are you?
fine and what are u doing
doing core Java
```

Program Exploring image package

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

public class AwtControlDemo {

   private Frame mainFrame;
   private Label headerLabel;
   private Label statusLabel;
   private Panel controlPanel;

   public AwtControlDemo(){
      prepareGUI();
   }

   public static void main(String[] args){
      AwtControlDemo  awtControlDemo = new AwtControlDemo();
      awtControlDemo.showImageDemo();
   }

   private void prepareGUI(){
      mainFrame = new Frame("Java AWT Examples");
      mainFrame.setSize(400,400);
      mainFrame.setLayout(new GridLayout(3, 1));
      mainFrame.addWindowListener(new WindowAdapter() {
         public void windowClosing(WindowEvent windowEvent){
            System.exit(0);
         }
      });
      headerLabel = new Label();
      headerLabel.setAlignment(Label.CENTER);
      statusLabel = new Label();
      statusLabel.setAlignment(Label.CENTER);
      statusLabel.setSize(350,100);

      controlPanel = new Panel();
      controlPanel.setLayout(new FlowLayout());

      mainFrame.add(headerLabel);
      mainFrame.add(controlPanel);
      mainFrame.add(statusLabel);
      mainFrame.setVisible(true);
   }

   private void showImageDemo(){
      headerLabel.setText("Control in action: Image");

      controlPanel.add(new ImageComponent("resources/java.jpg"));
      mainFrame.setVisible(true);
   }

   class ImageComponent extends Component {

      BufferedImage img;

      public void paint(Graphics g) {
         g.drawImage(img, 0, 0, null);
      }

      public ImageComponent(String path) {
         try {
            img = ImageIO.read(new File(path));
         } catch (IOException e) {
            e.printStackTrace();
         }
      }

      public Dimension getPreferredSize() {
         if (img == null) {
            return new Dimension(100,100);
         } else {
            return new Dimension(img.getWidth(), img.getHeight());
         }
      }
   }
}
```

OUTPUT:



Control in action: Image

Program demonstrating SWING components:

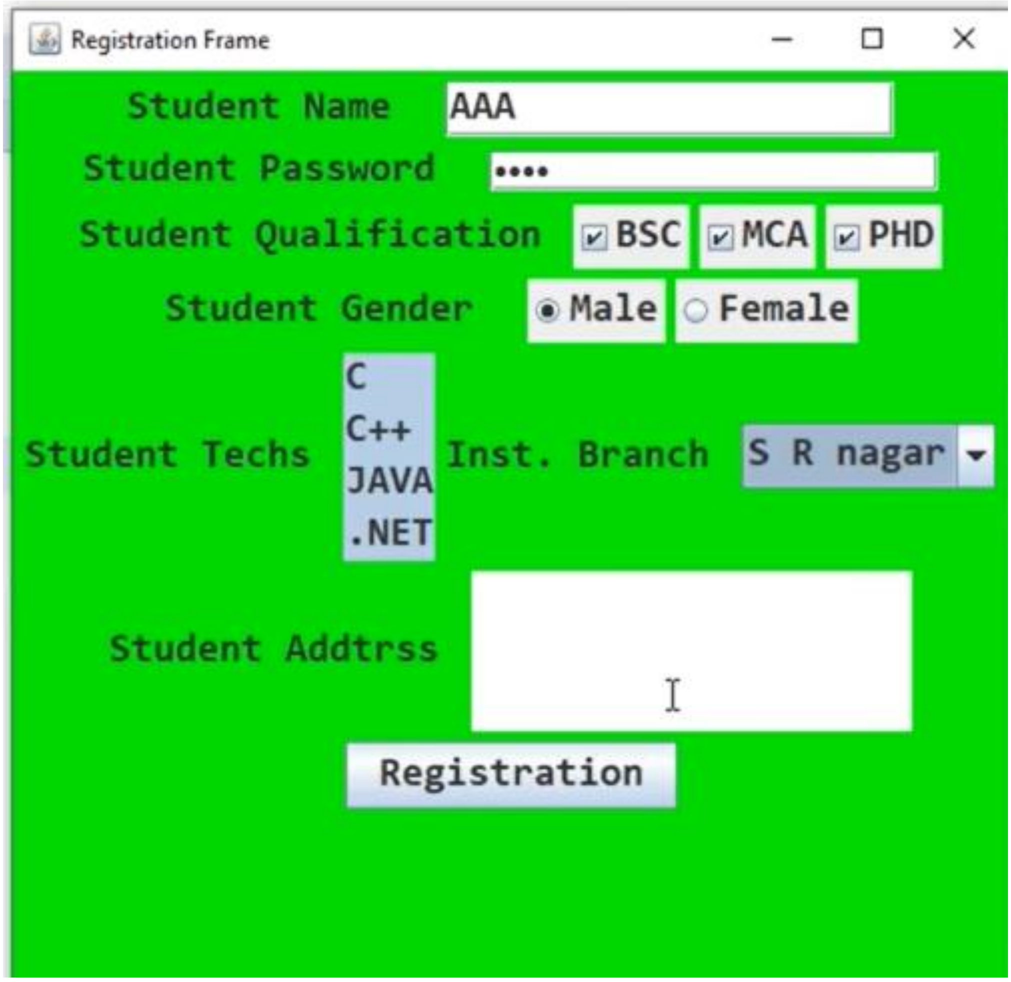
```
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Font;
import java.awt.Container;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JPasswordField;
import javax.swing.JRadioButton;
import javax.swing.JTextArea;
import javax.swing.JTextField;

public class RegistrationFrame extends JFrame implements ActionListener
{
        JLabel l1,l2,l3,l4,l5,l6,l7;
        JTextField tf;
        JPasswordField pf;
        JCheckBox cb1,cb2,cb3;
        JRadioButton rb1,rb2;
        JList l;
        JComboBox cb;
        JTextArea ta;
        JButton b;

        Container c;

        String
uname="",upwd="",uqual="",ugen="",utech="",uprof="",uaddr="";

        public RegistrationFrame(){
                this.setVisible(true);
                this.setSize(500,600);
                this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                c=getContentPane();
                c.setBackground(Color.cyan);
                this.setForeground(Color.magenta);
```

```
c.setLayout(null);
l1=new JLabel("Username");
l1.setBounds(50,100,100,10);
l2=new JLabel("Password");
l2.setBounds(50,150,100,10);
l3=new JLabel("Qualification");
l3.setBounds(50,200,100,10);
l4=new JLabel("Gender");
l4.setBounds(50,250,100,10);
l5=new JLabel("Technologies");
l5.setBounds(50,300,100,10);
l6=new JLabel("Proffession");
l6.setBounds(50,350,100,10);
l7=new JLabel("Address");
l7.setBounds(50,400,100,10);

tf=new JTextField(20);
tf.setBounds(150,90,100,30);
tf.setToolTipText("This is TextField");

pf=new JPasswordField(20);
pf.setBounds(150,140,100,30);
pf.setToolTipText("This is Password Field");

cb1=new JCheckBox("BSc");
cb1.setBounds(150,190,60,30);
cb2=new JCheckBox("MCA");
cb2.setBounds(220,190,60,30);
cb3=new JCheckBox("Phd");
cb3.setBounds(90,190,60,30);

rb1=new JRadioButton("Male");
rb1.setBounds(150,240,80,30);
rb2=new JRadioButton("Female");
rb2.setBounds(250,240,80,30);
ButtonGroup bg=new ButtonGroup();
bg.add(rb1);
bg.add(rb2);

String[] techs= {"C","C++","JAVA","SQL"};
l=new JList(techs);
l.setBounds(150,280,60,60);

String[] prof= {"Student","Business","Teacher"};
```

```
        cb=new JComboBox(prof);
        cb.setBounds(150,340,80,30);

        ta=new JTextArea(5,25);
        ta.setBounds(150,380,100,40);

        b=new JButton("Registration");
        b.setBounds(50,450,110,40);
        b.addActionListener(this);

        c.add(l1);c.add(tf);
        c.add(l2);c.add(pf);
        c.add(l3);c.add(cb1);c.add(cb2);c.add(cb3);
        c.add(l4);c.add(rb1);c.add(rb2);
        c.add(l5);c.add(l);
        c.add(l6);c.add(cb);
        c.add(l7);c.add(ta);
        c.add(b);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        uname=tf.getText();
        upwd=pf.~~getText()~~;
        if(cb1.isSelected()==true) {
            uqual=uqual+cb1.getText()+" ";
        }
        if(cb2.isSelected()==true) {
            uqual=uqual+cb2.getText()+" ";
        }
        if(cb3.isSelected()==true) {
            uqual=uqual+cb3.getText()+" ";
        }
        if(rb1.isSelected()==true) {
            ugen=ugen+rb1.getText()+" ";
        }
        if(rb2.isSelected()==true) {
            ugen=ugen+rb2.getText()+" ";
        }
        Object[] techs=l.~~getSelectedValues()~~;
        for(int i=0;i<techs.length;i++) {
            utech=utech+techs[i];
        }
        uprof=(String)cb.getSelectedItem();
```

```
                uaddr=ta.getText();

                class DisplayFrame extends JFrame{
                        DisplayFrame(){
                                this.setVisible(true);
                                this.setSize(500,500);
                                this.setBackground(Color.pink);
                        }
                        public void paint(Graphics g) {
                                Font f=new Font("arial",Font.BOLD,25);
                                g.setFont(f);
                                g.drawString("UserName
        :"+uname,50,100);
                                g.drawString("Password          :"+upwd,50,100);
                                g.drawString("Qualification :"+uqual,50,100);
                                g.drawString("User Gender       :"+ugen,50,100);
                                g.drawString("Technologies      :"+utech,50,100);
                                g.drawString("Proffession       :"+uprof,50,100);
                                g.drawString("Address           :"+uaddr,50,100);

                        }
                }
                DisplayFrame df=new DisplayFrame();
        }
}
public class Test {

        public static void main(String[] args) {
                RegistrationFrame rf=new RegistrationFrame();

        }

}
```

OUTPUT:



Registration Frame

Student Name AAA

Student Password ••••

Student Qualification ☑ BSC ☑ MCA ☑ PHD

Student Gender ● Male ○ Female

Student Techs C C++ JAVA .NET

Inst. Branch S R nagar

Student Addtrss

Program on object class

```
public class sample {

	public static void main(String[] args) {
		sample s = new sample();
		System.out.println(s.toString());
		Employee e1= new Employee(12,"abcd");
		System.out.println(e1.toString());
		Employee e2= new Employee(15,"xyz");
		System.out.println(e2.toString());

	}
	public String tostring() {
		return"this is sample object";
	}

}
class Employee{
	int empId;
	String ename;
	Employee(int epmId, String ename){
		this.empId=empId;
		this.ename=ename;
	}
	public String toString() {
		return empId+":"+ename;

	}
}
```

OUTPUT:

```
com.harika.core.sample@7637f22
0:abcd
0:xyz
```