1.Simple Applet

```
import java.applet.*;
import java.awt.*;
/*<applet code=simpleapplet width=200 height=200>
  </applet>*/

public class simpleapplet extends Applet
{
        public void init()
        {
                setBackground(Color.green);
        }


        public void paint(Graphics g)
        {
                g.drawString("SIMPLE APPLET",100,100);
        }
}
```

2.Shapes In Applet

```
import java.applet.*;
import java.awt.*;
/*<applet code=shapesapplet width=500 height=500>  </applet>*/
public class shapesapplet extends Applet
{
        public void init()
        {       setBackground(Color.white);    }
        public void paint(Graphics g)
        {
                g.drawLine(10,490,60,490);
                g.drawString("LINE",20,480);
                g.drawRect(20,410,70,40);
                g.drawString("RECTANGLE",30,400);
                g.fillRect(120,410,70,40);
                g.drawString("FILLED RECTANGLE",120,400);
                g.drawOval(40,300,40,70);
                g.drawString("OVAL",45,300);
                g.fillOval(140,300,40,70);
                g.drawString(" FILLED OVAL",145,300);
        }
}
```

3.Mouse Motions In Applet

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

/*<applet code=mousedemo width=500 height=500>
</applet>*/

public class mousedemo extends Applet implements
MouseListener,MouseMotionListener
{
    String msg;
        public void init()
        {
        addMouseListener(this);
        addMouseMotionListener(this);
        }
        public void mouseClicked(MouseEvent me)
        {       msg="MOUSE CLICKED";
                repaint();      }
        public void mouseEntered(MouseEvent me)
        {       msg="MOUSE ENTERED";
                repaint();}
        public void mouseExited(MouseEvent me)
        {       msg="MOUSE EXITED";
                repaint();}
        public void mousePressed(MouseEvent me)
        {
                msg="MOUSE PRESSED";
                repaint();}
        public void mouseReleased(MouseEvent me)
        {       msg="MOUSE RELEASED";
                repaint();}
        public void mouseDragged(MouseEvent me)
        {       msg="MOUSE DRAGGED";
                repaint();}
        public void mouseMoved(MouseEvent me)
        {       msg="MOUSE MOVED";
                repaint();}
        public void paint(Graphics g)
        {
        g.drawString(msg,100,100);
        }
}
```

4.Arithmetic Operations

```
import java.awt.*;
import java.awt.event.*;
public class arith extends Frame implements ActionListener
{
  Label l1,l2,l3;
  TextField t1,t2,t3;
  Button a,s,m,d,mo;
        arith()
        {
        l1=new Label("Enter 1st Number:");          t1=new TextField();
        l2=new Label("Enter 2nd Number:");          t2=new TextField();
        l3=new Label("The Answer Is:");             t3=new TextField();
        a=new Button("ADDITION");                   a.addActionListener(this);
        s=new Button("SUBTRACTION");                s.addActionListener(this);
        m=new Button("MULTIPLICATION");             m.addActionListener(this);
        d=new Button("DIVISION");                   d.addActionListener(this);
        mo=new Button("MODULUS");                   mo.addActionListener(this);

        addWindowListener(new WindowAdapter()
        {
                public void windowClosing(WindowEvent e)
                {
                dispose();
                }
        });
        add(l1);add(t1);
        add(l2);add(t2);
        add(l3);add(t3);
        add(a);add(s);
        add(m);add(d);
        add(mo);

        setLayout(new GridLayout(6,2,40,40));
        setSize(800,800);
        setVisible(true);
        }
        public void actionPerformed(ActionEvent ae)
        {
                if(ae.getSource()==a)
                {
                        int a,b,c;
                        a=Integer.parseInt(t1.getText());
                        b=Integer.parseInt(t2.getText());
                        c=a+b;
                        t3.setText(Integer.toString(c));
                }
                else if(ae.getSource()==s)
                {
                        int d,e,f;
                        d=Integer.parseInt(t1.getText());
                        e=Integer.parseInt(t2.getText());
                        f=d-e;
                        t3.setText(Integer.toString(f));
                }


                else if(ae.getSource()==m)
                {
                        int g,h,i;
                        g=Integer.parseInt(t1.getText());
                        h=Integer.parseInt(t2.getText());
                        i=g*h;
                        t3.setText(Integer.toString(i));
                }
                else if(ae.getSource()==d)
                {
                        double j,k,l;
                        j=Double.parseDouble(t1.getText());
                        k=Double.parseDouble(t2.getText());
                        l=j/k;
                        t3.setText(Double.toString(l));
                }
                else if(ae.getSource()==mo)
                {
                        double n,o,p;
                        n=Double.parseDouble(t1.getText());
                        o=Double.parseDouble(t2.getText());
                        p=n%o;
                        t3.setText(Double.toString(p));
                }
        }
        public static void main(String args[])
        {
                arith obj=new arith();
        }
}
```

5.Factorial

```
import java.awt.*;
import java.awt.event.*;
class factorial extends Frame implements ActionListener
{
	Label ti,l1,l2;
	TextField t1,t2;
	Button b1;
	factorial()
	{
		ti=new Label("Find the Factorial of number");
		ti.setBounds(150,40,156,50);
		l1=new Label("Enter the Number :");
		l1.setBounds(55,94,136,22);
		t1=new TextField();
		t1.setBounds(257,94,43,22);
		b1=new Button("FIND");
		b1.setBounds(358,94,70,22);
		b1.addActionListener(this);
		l2=new Label("Answer is :");
		l2.setBounds(55,176,136,22);
		t2=new TextField();
		t2.setBounds(257,176,43,22);

		addWindowListener(new WindowAdapter()
		{
			public void windowClosing(WindowEvent e)
			{
				dispose();
			}
		});
		add(ti);
		add(l1);
		add(t1);
		add(b1);
		add(l2);
		add(t2);
		setLayout(null);
		setSize(500,300);
		setVisible(true);
	}
	public void actionPerformed(ActionEvent ae)
	{
		if(ae.getSource()==b1)
		{
			int c,i,fact=1;
			c=Integer.parseInt(t1.getText());
			for(i=1;i<=c;i++)
			{
				fact=fact*i;
			}
			t2.setText(""+fact);

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

6.Positive or Negative

```
import java.awt.*;
import java.awt.event.*;
class posorneg extends Frame implements ActionListener
{
	Label l1,l2,l3;
	TextField t1;
	Button b1;
	posorneg()
	{
		l1=new Label("Enter the Number  :");
		t1=new TextField();
		b1=new Button("CHECK");
		b1.addActionListener(this);
		l2=new Label("This Number is :");
		l3=new Label();

	addWindowListener(new WindowAdapter()
		{
		public void windowClosing(WindowEvent e)
		{
			dispose();
		}
		});

		add(l1);
		add(t1);
		add(b1);
		add(l2);
		add(l3);
		setLayout(new GridLayout(2,3,5,5));
		setSize(500,200);
		setVisible(true);
	}
	public void actionPerformed(ActionEvent ae)
	{
		if(ae.getSource()==b1)
		{
			int c;
			c=Integer.parseInt(t1.getText());
			if(c>0)
			{
				l3.setText("Positive");
			}
```

```
                    else if(c<0)
                    {
                            l3.setText("Negative");
                    }
                    else
                    {
                            l3.setText("Zero");
                    }
            }
    }
    public static void main(String args[])
    {
            posorneg ch=new posorneg();
    }
}
```

7.Odd Or Even

```
import java.awt.*;
import java.awt.event.*;
class oddoreven extends Frame implements ActionListener
{
	Label l1,l2,l3;
	TextField t1;
	Button b1;
	oddoreven()
	{
		l1=new Label("Enter the Number  :");
		t1=new TextField();
		b1=new Button("Find");
		b1.addActionListener(this);
		l2=new Label("The Given Number is :");
		l3=new Label();

		addWindowListener(new WindowAdapter()
		{
			public void windowClosing(WindowEvent e)
			{
				dispose();
			}
		});

		add(l1);
		add(t1);
		add(b1);
		add(l2);
		add(l3);
		setTitle("check the given number is Odd or Even");
		setLayout(new GridLayout(2,3,40,40));
		setSize(500,200);
		setVisible(true);
	}
	public void actionPerformed(ActionEvent ae)
	{
		if(ae.getSource()==b1)
		{
			int c;
			c=Integer.parseInt(t1.getText());
			if(c % 2 != 0)
			{
				l3.setText("odd");
				}
				else if(c == 0)
				{
					l3.setText("zero");
				}
				else
				{
					l3.setText("even");
				}
			}
		}
		public static void main(String args[])
		{
			oddoreven ch=new oddoreven();
		}
}
```

8.Check Box [Language known]

```
import java.awt.*;
import java.awt.event.*;
class language extends Frame implements ActionListener
{
	Panel p1,p2;
	Label l1,l2,l3,l4,l5,l6,l7,l8,l9,l0;
	TextField t1;
	Button b1;
	Checkbox c1,c2,c3,c4;
	double pi,ye,ra,sim,com;

	language()
	{
		p1=new Panel();
		p2=new Panel();

		l1=new Label("CHECK BOX");
		l1.setBounds(178,10,120,30);

		l2=new Label("ENTER YOUR NAME:");
		l2.setBounds(10,44,120,22);

		t1=new TextField();
		t1.setBounds(140,44,90,22);

		c1=new Checkbox("C");
		c1.setBounds(178,72,95,22);

		c2=new Checkbox("C++");
		c2.setBounds(178,105,95,22);

		c3=new Checkbox("JAVA");
		c3.setBounds(178,139,95,22);


		c4=new Checkbox("PHP");
		c4.setBounds(178,167,85,22);

		b1=new Button("SUBMIT");
		b1.setBounds(250,120,85,22);
		b1.addActionListener(this);

		l3=new Label("You Selected...");
		l3.setBounds(190,10,95,22);
		l4=new Label("Your name:");
		l4.setBounds(10,55,75,30);

		l5=new Label();
		l5.setBounds(120,55,75,22);

		l6=new Label("Languages you have selected...");
		l6.setBounds(10,99,190,40);

		l7=new Label();
		l7.setBounds(221,99,62,22);

		l8=new Label();
		l8.setBounds(221,128,62,22);

		l9=new Label();
		l9.setBounds(221,161,62,22);

		l0=new Label();
		l0.setBounds(221,189,62,22);

		addWindowListener(new WindowAdapter()
		{
			public void windowClosing(WindowEvent e)
			{
				dispose();
			}
		});
		add(p1);		add(p2);
		p1.add(l1);		p1.add(l2);
		p1.add(t1);		p1.add(c1);
		p1.add(c2);		p1.add(c3);
		p1.add(c4);		p1.add(b1);
		p2.add(l3);		p2.add(l4);
		p2.add(l5);		p2.add(l6);
		p2.add(l7);		p2.add(l8);
		p2.add(l9);		p2.add(l0);
		setTitle("CHECKBOX");
		setLayout(new GridLayout(2,1,10,10));
		p1.setLayout(null);
		p2.setLayout(null);
		setSize(500,500);
		setVisible(true);
	}
	public void actionPerformed(ActionEvent en)
	{
		if(en.getSource()==b1)
		{
			String name;
			name=t1.getText();
			l5.setText(name);
			if(c1.getState()==true)
			{
				l7.setText(c1.getLabel());
			}
			if(c2.getState()==true)
			{
				l8.setText(c2.getLabel());
			}
			if(c3.getState()==true)
			{
				l9.setText(c3.getLabel());
			}
			if(c4.getState()==true)
			{
				l0.setText(c4.getLabel());
			}
			if(c1.getState()==false)
			{
				l7.setText("");
			}
			if(c2.getState()==false)
			{
				l8.setText("");
			}
			if(c3.getState()==false)
			{
				l9.setText("");
			}
			if(c4.getState()==false)
			{
				l0.setText("");
			}
		}
	}
	public static void main(String args[])
	{		language ch =new language();		}
}
```

9.Simple And Compound Interest [Check Box Group]

```
import java.awt.*;
import java.awt.event.*;
class simpandcomp extends Frame implements ItemListener
{
	Panel p1,p2;
	Label l1,l2,l3,l4,l5,l6,ans;
	TextField t1,t2,t3;
	CheckboxGroup in;
	Checkbox c1,c2;
	double pi,ye,ra,sim,com;
	simpandcomp()
	{
		p1=new Panel();
		p2=new Panel();

		in=new CheckboxGroup();

		l1=new Label("Enter the Principal Amount:");

		t1=new TextField();

		l2=new Label("Enter the Number of Years:");

		t2=new TextField();

		l3=new Label("Enter the Rate of Interest:");

		t3=new TextField();

		l4=new Label();

		l5=new Label();

		l6=new Label();

		ans=new Label();

		c1=new Checkbox("Simple Interest",in,false);
		c1.addItemListener(this);
		c2=new Checkbox("Compound Interest",in,false);
		c2.addItemListener(this);

		addWindowListener(new WindowAdapter()
        {
                public void windowClosing(WindowEvent e)
                {
                        dispose();
                }
        });

        add(p1);
        add(p2);
        p1.add(l1);
        p1.add(t1);
        p1.add(l2);
        p1.add(t2);
        p1.add(l3);
        p1.add(t3);
        p1.add(c1);
        p1.add(c2);
        p2.add(l4);
        p2.add(l5);
        p2.add(l6);
        p2.add(ans);

        setTitle("Find The Simple and Compound Interest");

        setLayout(new GridLayout(2,1,10,10));
        p1.setLayout(new GridLayout(4,2,10,10));
        p2.setLayout(new GridLayout(2,2,10,10));
        setSize(520,520);
        setVisible(true);
}
public void itemStateChanged(ItemEvent en)
{
        pi=Double.parseDouble(t1.getText());
        ye=Double.parseDouble(t2.getText());
        ra=Double.parseDouble(t3.getText());

        sim=(pi*ye*ra)/100;
        com=pi*Math.pow(1.0+ra/100.0,ye)-pi;

        if(c1.getState()==true)
        {
                l4.setText("The Principle Amount:"+pi);
                l5.setText("The Number of years:"+ye);
                l6.setText("The rate of Interest:"+ra);
                ans.setText("Simple Interest:"+sim);
                }
                if(c2.getState()==true)
                {
                        l4.setText("The Principle Amount:"+pi);
                        l5.setText("The Number of years:"+ye);
                        l6.setText("The rate of Interest:"+ra);
                        ans.setText("Simple Interest:"+com);
                }
        }
        public static void main(String args[])
        {
                simpandcomp ch =new simpandcomp();
        }
}
```

10.Scroll Bars

```
import java.awt.*;
import java.awt.event.*;
class scrollbar extends Frame implements AdjustmentListener
{
        Panel p1,p2;
        Scrollbar re,gre,bl;
        scrollbar()
        {
                p1=new Panel();
                p2=new Panel();

      re=new Scrollbar(Scrollbar.HORIZONTAL, 0, 5, 0, 255);
      re.setBackground(Color.RED);
      re.setBlockIncrement(5);
      re.addAdjustmentListener(this);



      gre=new Scrollbar(Scrollbar.HORIZONTAL, 0, 5, 0, 255);
      gre.setBackground(Color.GREEN);
      gre.setBlockIncrement(5);
      gre.addAdjustmentListener(this);

        bl=new Scrollbar(Scrollbar.HORIZONTAL, 0, 5, 0, 255);
      bl.setBackground(Color.BLUE);
      bl.setBlockIncrement(5);
      bl.addAdjustmentListener(this);

                addWindowListener(new WindowAdapter()
                {
                        public void windowClosing(WindowEvent en)
                        {
                                dispose();
                        }
                });

                add(p1);
                add(p2);

                p1.add(re);
                p1.add(gre);
                p1.add(bl);
                p1.setLayout(new GridLayout(3,1,70,70));
                p2.setLayout(new GridLayout(1,1));

                setLayout(new GridLayout(2,1,10,10));
                setSize(500,500);
                setVisible(true);
        }
    public void adjustmentValueChanged(AdjustmentEvent e)
    {

        int red, blue, green;
        red = green = blue = 0;
        if (e.getAdjustable() == re)
        {
                red = re.getValue();
                blue = bl.getValue();
                green = gre.getValue();
        }
        if (e.getAdjustable() == bl)
        {
                red = re.getValue();
                blue = bl.getValue();
                green = gre.getValue();

        }
        if (e.getAdjustable() == gre)
        {
          red = re.getValue();
                blue = bl.getValue();
                green = gre.getValue();
        }

        Color newcolor = new Color(red, green, blue);
        p2.setBackground(newcolor);
    }

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

11.Choice & List

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

class choiceandlist extends Frame implements ActionListener
{
        Panel p1,p2;
        Choice ch;
        List list;
        Label l1,l2,l3,l4,l5;
        TextArea l6;
        Button b1;

        choiceandlist()
        {
                p1=new Panel();
                p2=new Panel();

                list=new List();
                ch=new Choice();

                l1=new Label("SELECT FROM THE CHOICE");
                l2=new Label("SELECT FROM THE LIST ");
                l3=new Label("Choice:");
                l4=new Label("List :");
                l5=new Label();
                l6=new TextArea();

                b1=new Button("Submit");
                b1.addActionListener(this);

                ch.add("TAMIL");
                ch.add("ENGLISH");
                ch.add("HINDHI");
                ch.add("MALAYALAM");
                ch.add("TELUNGU");

                list.add("TAMIL");
                list.add("ENGLISH");
                list.add("HINDHI");
                list.add("MALAYALAM");
                list.add("TELUNGU");

                add(p1);
```

```
		add(p2);
		p1.add(l1);
		p1.add(l2);
		p1.add(ch);
		p1.add(list);
		p1.add(b1);
		p2.add(l3);
		p2.add(l5);
		p2.add(l4);
		p2.add(l6);

		addWindowListener(new WindowAdapter()
		{
			public void windowClosing(WindowEvent e)
			{
				dispose();
			}
		});

		list.setMultipleMode(true);

		p1.setLayout(new GridLayout(3,2,10,10));
		p2.setLayout(new GridLayout(2,2,10,10));

		setLayout(new GridLayout(2,1,10,10));
		setSize(500,500);
		setVisible(true);
	}
	public void actionPerformed(ActionEvent en)
	{
		if(en.getSource()==b1)
		{
			l5.setText(ch.getItem(ch.getSelectedIndex()));
			String[] item=list.getSelectedItems();
			if(item.length==0)
			{
			l6.setText("NO ITEMS CAN BE SELECTED");
			}

			for(String l:item)
			{
			l6.repaint();
			l6.append(l+",\n");
			}
```

```
        }


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

12.Text Area

```
import java.awt.*;
import java.awt.event.*;
class textarea extends Frame implements ActionListener
{

	Panel p1,p2;
	Label l1,l2;
	TextArea t1;
	TextField t2,t3;
	Button b1,b2;
	textarea()
	{
		p1=new Panel();
		p2=new Panel();

		l1=new Label("Enter the Text      :");
		l2=new Label("Enter the Location :");

		t1=new TextArea();

		t2=new TextField();
		t3=new TextField();

		b1=new Button("Append");
		b1.addActionListener(this);
		b2=new Button("Insert");
		b2.addActionListener(this);

		addWindowListener(new WindowAdapter()
		{
			public void windowClosing(WindowEvent e)
			{
				dispose();
			}
		});

		add(p1);
		add(p2);
		p1.add(t1);
		p2.add(l1);
		p2.add(t2);
		p2.add(l2);
		p2.add(t3);
		p2.add(b1);
		p2.add(b2);

		p1.setLayout(new GridLayout(1,1));
		p2.setLayout(new GridLayout(3,2,40,40));

		setLayout(new GridLayout(2,1,10,10));
		setTitle("TEXT AREA EXAMPLE PROGRAM");
		setSize(500,500);
		setVisible(true);
	}
	public void actionPerformed(ActionEvent en)
	{
		String g,h;int a;
		if(en.getSource()==b1)
		{
			g=t1.getText();
			h=t2.getText();
			t1.append(h);
		}
		if(en.getSource()==b2)
		{
			g=t1.getText();
			h=t2.getText();
			a=Integer.parseInt(t3.getText());
			t1.insert(h,a);
		}
	}
	public static void main(String args[])
	{
		textarea e=new textarea();
	}
}
```

13.Menu Bar

```
import java.awt.*;
import java.awt.event.*;
public class menubar extends Frame implements ActionListener
{
        Label l1,l2;
        MenuBar mb;
        MenuItem newfile,window,open,save,saveas, cut,copy,paste,closetag,opentag;
        Menu file,edit,submenu;
        menubar()
        {
                l1=new Label("selected menu is:");
                l2=new Label();
                mb=new MenuBar();
                file=new Menu("File");
                edit=new Menu("Edit");
                submenu=new Menu("tag");
                closetag=new MenuItem("closetag");
                closetag.addActionListener(this);
                opentag=new MenuItem("opentag");
                opentag.addActionListener(this);
                cut=new MenuItem("Cut");
                cut.addActionListener(this);
                copy=new MenuItem("Copy");
                copy.addActionListener(this);
                paste=new MenuItem("paste");
                paste.addActionListener(this);
                newfile=new MenuItem("NewFile");
                newfile.addActionListener(this);
                window=new MenuItem("Window");
                window.addActionListener(this);
                open=new MenuItem("Open");
                open.addActionListener(this);
                save=new MenuItem("Save");
                save.addActionListener(this);saveas=new MenuItem("SaveAs");
                saveas.addActionListener(this);

        addWindowListener(new WindowAdapter ()
        {
                public void windowClosing(WindowEvent e)
                {
                        dispose();
                }
        });
        add(l1);
        add(l2);
        file.add(newfile);
        file.add(window);
        file.add(open);
        file.add(save);
        file.add(saveas);
        edit.add(cut);edit.add(copy);
        edit.add(paste);
        setMenuBar(mb);
        mb.add(file);
        mb.add(edit);
        submenu.add(closetag);
        submenu.add(opentag);
        edit.add(submenu);
        setSize(500,500);
        setVisible(true);
        setLayout(null);
    }
    public void actionPerformed(ActionEvent ae)
    {
        if(ae.getSource()==newfile)
        {
            l2.setText(newfile.getLabel());
        }
        if(ae.getSource()==window)
        {
            l2.setText(window.getLabel());
        }
        if(ae.getSource()==open)
        {
            l2.setText(open.getLabel());
        }
        if(ae.getSource()==save)
        {
            l2.setText(save.getLabel());
        }
        if(ae.getSource()==saveas)
        {
            l2.setText(saveas.getLabel());
        }
        if(ae.getSource()==cut)
        {
            l2.setText(cut.getLabel());
        }
```

```
		if(ae.getSource()==copy)
		{
			l2.setText(copy.getLabel());
		}
		if(ae.getSource()==paste)
		{
			l2.setText(paste.getLabel());
		}
		if(ae.getSource()==closetag)
		{
			l2.setText(closetag.getLabel());
		}
		if(ae.getSource()==opentag)
		{
			l2.setText(opentag.getLabel());}
		}

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

14.Dialog Box

```
import java.awt.*;
import java.awt.event.*;
public class dialogdemo extends Frame implements ActionListener
{
	Dialog d;
	Button b;

	dialogdemo()
	{
	d=new Dialog(this,"Dialog Example");
	d.setLayout(new FlowLayout());
	b=new Button("ok");
	b.addActionListener(this);
	d.add(new Label("click button to continue"));
	d.add(b);
	d.setSize(150,150);
	d.setVisible(true);
	setSize(400,400);



		addWindowListener(new WindowAdapter()
		{
			public void windowClosing(WindowEvent ea)
			{
				dispose();
			}
		});

	setVisible(true);
	}

	public void actionPerformed(ActionEvent e)
	{
		d.setVisible(false);
	}

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

15.File Dialog Box

```
import java.awt.*;
import java.awt.event.*;
public class filedemo extends Frame implements ActionListener
{
        Label headerLabel;
        Label statusLabel;
        Panel controlPanel;
        FileDialog filedialog;
        Button lo;


        public filedemo()
        {
                setSize(400,400);
                setLayout(new GridLayout(3,1));

        addWindowListener(new WindowAdapter()
        {
                public void windowClosing(WindowEvent we)
                {
                        dispose();
                }
        });


                headerLabel=new Label();
                headerLabel.setAlignment(Label.CENTER);

                statusLabel=new Label();
                statusLabel.setAlignment(Label.CENTER);

                statusLabel.setSize(350,350);

                lo=new Button("open");
                lo.addActionListener(this);


                controlPanel=new Panel();
                controlPanel.setLayout(new FlowLayout());
                controlPanel.add(lo);
                filedialog=new FileDialog(this,"select file");

                add(headerLabel);
```

```
		add(controlPanel);
		add(statusLabel);
		setVisible(true);

	}


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

	public void actionPerformed(ActionEvent ae)
	{

	headerLabel.setText("Control in Action:fileDialog");
	filedialog.setVisible(true);
	statusLabel.setText("File Selected:"+filedialog.getDirectory()
+filedialog.getFile());
	}
}
```

16.LOG IN (IMAGE ICON)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class login implements ActionListener
{
	JFrame frame;
	JLabel l1,l2;
	JButton b1;
	JTextField t1;
	JPasswordField t2;
	JDialog d;
	login()
	{
		frame=new JFrame("Login Page");
		l1=new JLabel("User Id:");
		l2=new JLabel("Password:");
		t1=new JTextField();
		t2=new JPasswordField();
		b1=new JButton(new ImageIcon("index.png"));
		b1.addActionListener(this);
		d=new JDialog(frame,"LOGGING IN",true);
		frame.add(l1);
		frame.add(t1);
		frame.add(l2);
		frame.add(t2);
		frame.add(b1);
		frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		frame.setLayout(new GridLayout(1,1,20,20));
		frame.setSize(600,600);
		frame.setVisible(true);
	}
	public void actionPerformed(ActionEvent ae)
	{
		if(ae.getSource()==b1)
		{
			d.setLayout(new FlowLayout());
			d.add(new JLabel("Login Successfully"));
			d.setSize(150,150);
			d.setVisible(true);
		}

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

17.SWING USING PRIME NUMBER

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class prime implements ActionListener
{
	JFrame frame;
	JPanel p1,p2;
	JLabel l1,l2;
	JButton b1;
	JTextField t1;
	prime()
	{
		frame=new JFrame();
		p1=new JPanel();
		p2=new JPanel();
		l1=new JLabel("Enter The number:");
		l2=new JLabel();
		b1=new JButton("Find");
		b1.addActionListener(this);
		t1=new JTextField();

			frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
			frame.add(p1);
			frame.add(p2);
			p1.add(l1);
			p1.add(t1);
			p2.add(b1);
			p2.add(l2);
			p1.setLayout(new GridLayout(1,1));
			p2.setLayout(new GridLayout(1,1));
			frame.setLayout(new GridLayout(2,1));
			frame.setSize(400,250);
			frame.setVisible(true);
	}

	public void actionPerformed(ActionEvent ae)
	{
		int i,m=0,num,flag=0;
		num=Integer.parseInt(t1.getText());
		m=num/2;

			if(ae.getSource()==b1)
			{
				if(num==0||num==1)
				{
					l2.setText(m+" Is Not Prime");
```

```
                }
                else
                {
                    for(i=2;i<=m;i++)
                    {
                        if(num%i==0)
                        {
                            flag=1;
                            break;
                        }
                    }
                        if(flag==0)
                        {
                            l2.setText(num+" Is  Prime");
                        }

                        else
                        {
                            l2.setText(num+" Is Not Prime");
                        }
                }

            }
    }


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

18.JTABLE DEMONSTRATION

```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
class table extends JFrame
{
	table()
	{
		setTitle("JTable example");
		String columnNames[]={"Name","Roll Number","college","Department"};
		String data[][]={{"SANTHOSHKUMAR","74","VHNSNC","BCA"},
					{"SARATHIKANNAN","75","VHNSNC","IT"},
					{"SELVAKUMAR","76","ANJAC","COMMERCE"},
					{"SIVANANTHAM","77","ANJAC","COMMERCE"},
					{"SANJAYKUMAR","78","AMC","IT"}};
		JTable j=new JTable(data,columnNames);
		JScrollPane sp=new JScrollPane(j);
		add(sp);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setSize(200,150);
		setVisible(true);
	}
	public static void main(String args[])
	{
		new table();
	}
}
```

19.JTREE DEMONSTRATION

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

public class Tree extends JFrame
{
	JTree tree;
	DefaultMutableTreeNode root,parent1,parent2,parent3,child,child1,child2,child3;
	JTextField tf;
	Tree()
	{
		super("JTree Demo");
		Container cpane=getContentPane();
		root = new DefaultMutableTreeNode("States");
		parent1 = new DefaultMutableTreeNode("Tamil Nadu");
		child = new DefaultMutableTreeNode("Virudhunagar");
		child1 = new DefaultMutableTreeNode("Madurai");
		parent2 = new DefaultMutableTreeNode("pondicheri");
		child2 = new DefaultMutableTreeNode("Goa");
		parent3=new  DefaultMutableTreeNode("kerala");
		child3 = new DefaultMutableTreeNode("kollam");
		parent1.add(child);
		parent1.add(child1);
		parent2.add(child2);
		parent3.add(child3);
		root.add(parent1);
		root.add(parent2);
		root.add(parent3);
		tree = new JTree(root);
		cpane.add(new JScrollPane(tree),BorderLayout.CENTER);
		tf = new JTextField(20);
		cpane.add(tf,BorderLayout.SOUTH);
		tree.addMouseListener(new MouseAdapter()
		{
			public void mouseClicked(MouseEvent me)
			{
				TreePath tp= tree.getPathForLocation(me.getX(),me.getY());

				if(tp!=null)
					tf.setText(tp.toString());
				else
					tf.setText("");
			}
		});
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setSize(400,300);
		setVisible(true);
	}
	public static void main(String args[]) throws Exception
	{
		new Tree();
	}
}
```

20.JSPINNER(DATE OF BIRTH)

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
class spinner extends JFrame implements ChangeListener
{
	JLabel l1,l2,l3,l4,l5;
	JSpinner s,s1,s2;
	spinner()
	{
		super("Select your data of Birth:");
		l1=new JLabel("Date:");
		l2=new JLabel("Month:");
		l3=new JLabel("Year:");
		l4=new JLabel("Your DataOfBirth:");
		l5=new JLabel();
		s=new JSpinner();
		s1=new JSpinner(new SpinnerNumberModel(1,1,31,1));
		s.setValue(2000);
		String
month[]={"January","February","March","April","May","June","July","August","Septembe
r","November","December"};
		s2=new JSpinner(new SpinnerListModel(month));
		s.addChangeListener(this);
		s1.addChangeListener(this);
		s2.addChangeListener(this);
		add(l1);
		add(l2);
		add(l3);
		add(s1);
		add(s2);
		add(s);
		add(l4);
		add(l5);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setLayout(new GridLayout(3,3,10,10));
		setSize(400,150);
		setBackground(Color.blue);
		setVisible(true);
	}
	public void stateChanged(ChangeEvent e)
	{
		String n;

		l5.setText(s1.getValue()+"-"+s2.getValue()+"-"+s.getValue());
	}
	public static void main(String args[]){ new spinner();} }
```

21.JTABBED PANE DEMONSTRATION

```
	import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class tpane extends JFrame implements ActionListener
{
	JTabbedPane tp;
	JPanel p1,p2;
	JLabel l1,l2,l3,l4;
	JTextField t1,t2;
	JButton b1,b2;

	tpane()
	{
		tp=new JTabbedPane();
		p1=new JPanel();
		p2=new JPanel();
		l1=new JLabel("Enter the side of square:");
		l2=new JLabel("Enter the radius of circle:");
		l3=new JLabel();
		l4=new JLabel();
		t1=new JTextField();
		t2=new JTextField();
		b1=new JButton("SQUARE");
		b1.addActionListener(this);
		b2=new JButton("CIRCLE");
		b2.addActionListener(this);

		setDefaultCloseOperation(EXIT_ON_CLOSE);
		setVisible(true);
		setLayout(null);
		setSize(500,500);

		add(p1);
		add(p2);
		p1.add(l1);
		p1.add(t1);
		p1.add(b1);
		p1.add(l3);
		p1.setBackground(Color.red);
		p1.setLayout(new GridLayout(2,2));
		p2.add(l2);
		p2.add(t2);
		p2.add(b2);
		p2.add(l4);
		p2.setBackground(Color.green);
		p2.setLayout(new GridLayout(2,2));
		tp.addTab("SQUARE",p1);
		tp.addTab("CIRCLE",p2);
		tp.setSize(400,400);
		add(tp);
	}
	public void actionPerformed(ActionEvent ae)
	{
		if(ae.getSource()==b1)
		{
			int a=0;
			int A=0;
			a=Integer.parseInt(t1.getText());
			A=a*a;
			l3.setText("The Area Of Square Is:"+A);
		}

		if(ae.getSource()==b2)
		{
			int r=0;
			float A;
			r=Integer.parseInt(t2.getText());
			A=3.14f*r*r;
			l4.setText("The Area Of Circle Is:"+A);
		}

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

22.JTABLE EDIT OPERATIONS

```
import javax.swing.*;
import javax.swing.event.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.table.DefaultTableModel;
class table1 extends JFrame implements ActionListener
{
	JTable j;
	DefaultTableModel model;
	JPanel p1,p2;
	JButton b1;
	JLabel l1,l2,l3;
	JTextField t1,t2,t3;
	table1()
	{
		p1=new JPanel();
		p2=new JPanel();
		l1=new JLabel("Enter the row number:");
		l2=new JLabel("Enter the column number:");
		l3=new JLabel("Enter the text:");
		t1=new JTextField();
		t2=new JTextField();
		t3=new JTextField();
		setTitle("Jtable example");
		String columnname[]={"Name","Rollno","Department"};
		String data[][]={{"santhosh","20suca074","IT"},
					{"sarathi","20suca075","IT"},
					{"selva","20suca076","IT"},
					{"selva","20suca077","IT"},
					{"siva","20suca078","IT"},
					{"tamil","20suca079","IT"},
					{"kannan","20suca080","IT"}};
		 model = new DefaultTableModel(data, columnname);
		j=new JTable(model);
		JScrollPane sp=new JScrollPane(j);
		b1=new JButton("Edit");
		b1.addActionListener(this);
		p1.add(sp);
		p1.setLayout(new GridLayout(1,1));
		p2.add(l1);
		p2.add(t1);
		p2.add(l2);
		p2.add(t2);
		p2.add(l3);
		p2.add(t3);
		p2.add(b1);
		p2.setLayout(new GridLayout(4,2));
		add(p1);
		add(p2);
		setLayout(new GridLayout(2,1));
		setSize(500,500);
		setVisible(true);
	}
	public void actionPerformed(ActionEvent ae)
	{
		if(ae.getSource()==b1)
		{
			String value = t3.getText();
			int r,c;
			r=Integer.parseInt(t1.getText());
			c=Integer.parseInt(t2.getText());
	      j.setValueAt(value, r, c);
		}
	}
	public static void main(String args[])
	{
		new table1();
	}
}
```

23.EDIT TABLE

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class main extends JFrame implements ActionListener
{
	JTable j;
	JTextField t1;
	JLabel l1;
	JButton b1;
	main()
	{
		String colname[]={"NAME","ACCOUNT NO","BALANCE"};
		String data[][]={{"VICTOR","suca001","10000"},
					{"STEVEN","suca002","20000"},
					{"THOMAS","suca003","30000"}};
		j=new JTable(data,colname);
		b1=new JButton("Submit");
		b1.addActionListener(this);

		JScrollPane sp=new JScrollPane(j);
		add(sp);
		add(b1);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		Panel p1=new Panel();
		Panel p2=new Panel();

		p1.add(sp);
		p2.add(b1);

		p1.setLayout(new FlowLayout(FlowLayout.LEFT,30,30));
		p2.setLayout(new FlowLayout(FlowLayout.LEFT,20,20));

		add(p1);
		add(p2);

		setLayout(new GridLayout(2,1,10,10));
		setVisible(true);
		setSize(700,700);
	}
	public void actionPerformed(ActionEvent e)
	{
		int r=j.getSelectedRow(),c=j.getSelectedColumn();
		j.editCellAt(r,c);
	}
	public static void main(String args[])
	{
```

```
                new main();
        }
}
```

24.Display current date and time

```
<%@ page import="java.io.*,java.util.*"%>
<html>
<head>
<title>current time & date</title>
</head>
<body>
<center>
<h1>Display current date & time</h1>
</center>
<%
  Date date=new Date();
  out.print("<h2 align =\"center\">"+date.toString()+"</h2>");
%>
</body>
</html>
```

25.Digital clock

```
<%@ page import = "java.io.*,java.util.*" %>

<html>
   <head>
      <title>DIGITAL CLOCK</title>
   </head>
   <body>
      <center>
         <h2>DIGITAL CLOCK</h2>
         <%
            // Set refresh, autoload time as 5 seconds
            response.setIntHeader("Refresh", 1);
            // Get current time
            Calendar calendar = new GregorianCalendar();
            String am_pm;
            int hour = calendar.get(Calendar.HOUR);
            int minute = calendar.get(Calendar.MINUTE);
            int second = calendar.get(Calendar.SECOND);
            if(calendar.get(Calendar.AM_PM) == 0)
               am_pm = "AM";
            else
               am_pm = "PM";
            String CT = hour+":"+ minute +":"+ second +" "+ am_pm;
            out.println("Current Time is: " + CT + "\n");
         %>
      </center>
   </body>
</html>
```

26.Display day using switch case

day.jsp

```
<html>
<head>
<title>Display the day</title>
</head>
<body>
<h1>
<%
int day=Integer.parseInt(request.getParameter("input"));
switch(day)
{
case 0:
out.println("It\'s Sunday.");
break;
case 1:
out.println("It\'s Monday.");
break;
case 2:
out.println("It\'s Tuesday.");
break;
case 3:
out.println("It\'s Wednesday.");
break;
case 4:
out.println("It\'s Thursday.");
break;
case 5:
out.println("It\'s Friday.");
break;
case 6:
out.println("It\'s Saturday.");
break;
default:
out.println("Invalid day");
}
%>
</h1>
</body>
</html>
```

day.html

```
<html>
<head>
<title>SWITCHCASE</title>
</head>
<body>
```

```
<h1>SWITCHCASE</h1>
<h3>
<form action="day.jsp" method="POST">
Enter the day:<input type="text" name="input"><br>
<input type="submit" value="submit">
</form>
</h3>
</body>
</html>
```

27.Fibonacci

```
<html>
<head><title>FIBONACCI SERIES IN JSP</title></head>
<body>
<form method="get">
<h3> Enter the number of terms you want:
<input type="text" name="limit">
</h3>
</form>
<h3>
<%
 String s = request.getParameter("limit");
 if (s != null) {
%>
<%@ page import = "java.io.*" %>
<%@ page import = "java.lang.*" %>
<%
    int n=0;
    n=Integer.parseInt(s);
   out.println("No of terms to be printed is  "+n);
%>
<br>
<br>
<br>
The series generated are listed below :<br><br>
<%
    int a=1;
    int b=1;
   out.println(""+a+",\t"+b+",\t");
  for(int i=3;i<= n;i++)
{
    int c=a+b;
   out.print(""+c+",\t");
  a=b;
 b=c;
}
}
%>
</h3>
</body>
</html>
```

28.Cookie

```
<html>
<head>
<title>Cookies</title>
</head>
<body>
<h1>Cookies</h1>
<h3>
<form method="post">
Username:<input type="text" name="uname"><br>
Password:<input type="text" name="password"><br>
<input type="submit" value="Login">
</form>
<%
String s=request.getParameter("uname");
if(s != null)
{
Cookie unames = new Cookie("Username",request.getParameter("uname"));
Cookie passwords = new Cookie("Password",request.getParameter("password"));
unames.setMaxAge(60*60*24);
passwords.setMaxAge(60*60*24);
response.addCookie(unames);
response.addCookie(passwords);
out.print("Login successfully");
}
%>
</h3>
</body>
</html>
```

29.Page count

```
<%!
   int pageCount=0;
   void addCount()
   {
    pageCount++;
   }
%>
<% addCount();%>
<html>
<head>
<title>
no of time visiting
</title>
</head>
<body>
<center>
<h2>No Of Times Visiting</h2>
<p>The site has been visited<%= pageCount %>times</p>
</center>
<br/><br/>
</body>
</html>
```

30.Font

```
<%! int fontsize;%>
<html>
<head>
<title>for loop</title>
</head>
<body>
<%for(fontsize=1;fontsize<=3;fontsize++){%>
<font color="red"size="<%=fontsize%>">
JAVA SERVER PAGE
</font>
<br/>
<%}%>
</body>
</html>
```