

Applet

```
/*  
  
 * To change this license header, choose License Headers in Project Properties.  
  
 * To change this template file, choose Tools | Templates  
  
 * and open the template in the editor.  
  
 */
```

```
package javaapplication1;
```

```
/**  
  
 *  
  
 * @author GAURAV  
  
 */
```

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

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

```
import java.util.*;
```

```
import java.text.*;
```

```
public class DigitalClock extends Applet implements Runnable {
```

```
    Thread t = null;
```

```
    int hours=0, minutes=0, seconds=0;
```

```
    String timeString = "";
```

```
    @Override
```

```
    public void init() {
```

```
setBackground( Color.green);  
}
```

@Override

```
public void start() {  
    t = new Thread( this );  
    t.start();  
}
```

@Override

```
public void run() {  
    try {  
        while (true) {  
  
            Calendar cal = Calendar.getInstance();  
            hours = cal.get( Calendar.HOUR_OF_DAY );  
            if ( hours > 12 ) hours -= 12;  
            minutes = cal.get( Calendar.MINUTE );  
            seconds = cal.get( Calendar.SECOND );  
  
            SimpleDateFormat formatter = new SimpleDateFormat("hh:mm:ss");  
            Date date = cal.getTime();  
            timeString = formatter.format( date );
```

```
        repaint();

        Thread.sleep( 1000 ); // interval given in milliseconds
    }
}

catch (InterruptedException e) { }
}
```

```
@Override

public void paint( Graphics g ) {

    g.setColor( Color.blue );

    g.drawString( timeString, 50, 50 );

}

}
```

```
/*

* To change this license header, choose License Headers in Project Properties.

* To change this template file, choose Tools | Templates

* and open the template in the editor.

*/

package javaapplication1;
```

```
/**
 *
 * @author GAURAV
 */
import java.applet.*;
import java.awt.*;
import java.util.*;
import java.text.*;

public class AnalogClock extends Applet implements Runnable {

    int width, height;

    Thread t = null;

    boolean threadSuspended;

    int hours=0, minutes=0, seconds=0;

    String timeString = "";

    public void init() {

        width = getSize().width;

        height = getSize().height;

        setBackground( Color.black );

    }
```

```
public void start() {  
    if ( t == null ) {  
        t = new Thread( this );  
        t.setPriority( Thread.MIN_PRIORITY );  
        threadSuspended = false;  
        t.start();  
    }  
    else {  
        if ( threadSuspended ) {  
            threadSuspended = false;  
            synchronized( this ) {  
                notify();  
            }  
        }  
    }  
}
```

```
public void stop() {  
    threadSuspended = true;  
}
```

```
public void run() {  
    try {  
        while (true) {
```

```

Calendar cal = Calendar.getInstance();

hours = cal.get( Calendar.HOUR_OF_DAY );

if ( hours > 12 ) hours -= 12;

minutes = cal.get( Calendar.MINUTE );

seconds = cal.get( Calendar.SECOND );


SimpleDateFormat formatter

    = new SimpleDateFormat( "hh:mm:ss", Locale.getDefault() );

Date date = cal.getTime();

timeString = formatter.format( date );


// Now the thread checks to see if it should suspend itself

if ( threadSuspended ) {

    synchronized( this ) {

        while ( threadSuspended ) {

            wait();

        }

    }

    repaint();

    t.sleep( 1000 ); // interval specified in milliseconds

}

}

catch (Exception e) { }

}

```

```

void drawHand( double angle, int radius, Graphics g ) {

    angle -= 0.5 * Math.PI;

    int x = (int)( radius*Math.cos(angle) );

    int y = (int)( radius*Math.sin(angle) );

    g.drawLine( width/2, height/2, width/2 + x, height/2 + y );

}

```

```

void drawWedge( double angle, int radius, Graphics g ) {

    angle -= 0.5 * Math.PI;

    int x = (int)( radius*Math.cos(angle) );

    int y = (int)( radius*Math.sin(angle) );

    angle += 2*Math.PI/3;

    int x2 = (int)( 5*Math.cos(angle) );

    int y2 = (int)( 5*Math.sin(angle) );

    angle += 2*Math.PI/3;

    int x3 = (int)( 5*Math.cos(angle) );

    int y3 = (int)( 5*Math.sin(angle) );

    g.drawLine( width/2+x2, height/2+y2, width/2 + x, height/2 + y );

    g.drawLine( width/2+x3, height/2+y3, width/2 + x, height/2 + y );

    g.drawLine( width/2+x2, height/2+y2, width/2 + x3, height/2 + y3 );

}

```

```

public void paint( Graphics g )

{

```

```

        width = getSize().width;

        height = getSize().height;


        g.setColor( Color.gray );

        drawWedge( 2*Math.PI * hours / 12, width/6, g );

        drawWedge( 2*Math.PI * minutes / 60, width/5, g );

        drawHand( 2*Math.PI * seconds / 60, width/4, g );

        g.setColor( Color.white );

        g.drawOval(0,0,width,height);

        // g.drawString( timeString, 10, height-10 );
    }
}


/*

* To change this license header, choose License Headers in Project Properties.
* To change this template file, choose Tools | Templates
* and open the template in the editor.
*/

package javaapplication1;


/**
 *
 * @author GAURAV
 */

```



```
import java.awt.*;

import java.applet.*;

public class krti extends Applet

{

    @Override

    public void paint(Graphics g)

    {

        g.drawOval(100,100,400,350);

        g.drawOval(200,215,50,30);

        g.drawOval(350,215,50,30);

        g.drawOval(70,250,30,70);

        g.drawOval(500,250,30,70);

        g.setColor(Color.pink);

        g.fillOval(260,350,80,30);

        g.setColor(Color.red);

        g.fillOval(287,185,20,20);

        g.setColor(Color.black);

        g.fillOval(219,224,10,10);

        g.setColor(Color.black);

        g.fillOval(369,224,10,10);

        g.drawLine(270,330,330,330);

        g.drawLine(270,330,300,250);

        g.drawLine(300,250,330,330);

        g.setFont(new Font("default", Font.BOLD, 25));
```

```
g.drawString("Designed by: Krti", 180,480);  
}  
}  
/*  
<applet code="Face.class" height="400" width="600"></applet>  
*/
```