1. **CREATE A BANNER USING APPLET**

**SOLUTION 1**

Following example demonstrates how to play a sound using an applet image using Thread class. It also uses drawRect(), fillRect(), drawString() methods of Graphics class.

**CODE**

import java.awt.\*;

import java.applet.\*;

public class SampleBanner extends Applet implements Runnable {

String str = "This is a simple Banner ";

Thread t ;

boolean b;

public void init() {

setBackground(Color.gray);

setForeground(Color.yellow);

}

public void start() {

t = new Thread(this);

b = false;

t.start();

}

public void run () {

char ch;

for( ; ; ) {

try {

repaint();

Thread.sleep(250);

ch = str.charAt(0);

str = str.substring(1, str.length());

str = str + ch;

}

catch(InterruptedException e) {}

}

}

public void paint(Graphics g) {

g.drawRect(1,1,300,150);

g.setColor(Color.yellow);

g.fillRect(1,1,300,150);

g.setColor(Color.red);

g.drawString(str, 1, 150);

}

}

**SOLUTION 2**

The following is another sample example to create a banner using Applet

**CODE**

import java.awt.\*;

import java.applet.\*;

public class NewApplet extends Applet implements Runnable {

String msg = " It is a moving Banner. ";

char cha;

boolean stopFlag = true;

Thread t = null;

public void start() {

t = new Thread(this);

stopFlag = false;

t.start();

}

public void run() {

for(;;) {

try {

repaint();

Thread.sleep(250);

cha = msg.charAt(0);

msg = msg.substring(1,msg.length());

msg = msg + cha;

if(stopFlag) break;

}

catch(InterruptedException e) {}

}

}

public void stop(){

stopFlag = true;

t = null;

}

public void paint(Graphics g) {

g.drawString(msg,60,30);

}

}

1. **DISPLAY CLOCK USING APPLET**

**SOLUTION 1**

Following example demonstrates how to display a clock using valueOf() methods of String Class. & using Calender class to get the second, minutes & hours.

**CODE**

import java.awt.\*;

import java.applet.\*;

import java.applet.\*;

import java.awt.\*;

import java.util.\*;

public class ClockApplet extends Applet implements Runnable {

Thread t,t1;

public void start() {

t = new Thread(this);

t.start();

}

public void run() {

t1 = Thread.currentThread();

while(t1 == t) {

repaint();

try {

t1.sleep(1000);

}

catch(InterruptedException e){}

}

}

public void paint(Graphics g) {

Calendar cal = new GregorianCalendar();

String hour = String.valueOf(cal.get(Calendar.HOUR));

String minute = String.valueOf(cal.get(Calendar.MINUTE));

String second = String.valueOf(cal.get(Calendar.SECOND));

g.drawString(hour + ":" + minute + ":" + second, 20, 30);

}

}

**SOLUTION 2**

The following is an another sample example to display clock using Applet.

**CODE 2**

import java.applet.\*;

import java.awt.\*;

import java.util.\*;

import java.text.\*;

public class javaApplication6 extends Applet implements Runnable {

Thread t1 = null;

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

String time = "";

public void init() {

setBackground( Color.green);

}

public void start() {

t1 = new Thread( this );

t1.start();

}

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 d = cal.getTime();

time = formatter.format( d );

repaint();

t1.sleep( 1000 );

}

}

catch (Exception e) { }

}

public void paint( Graphics g ) {

g.setColor( Color.blue );

g.drawString( time, 50, 50 );

}

}