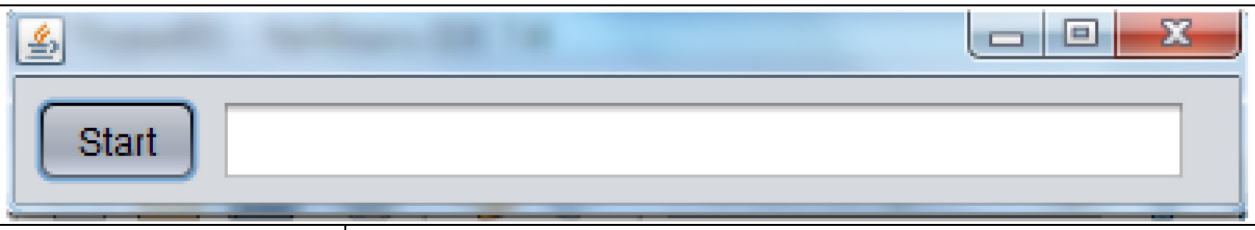
Fundamentals of Programming II Lab 14

Thread01



Thread

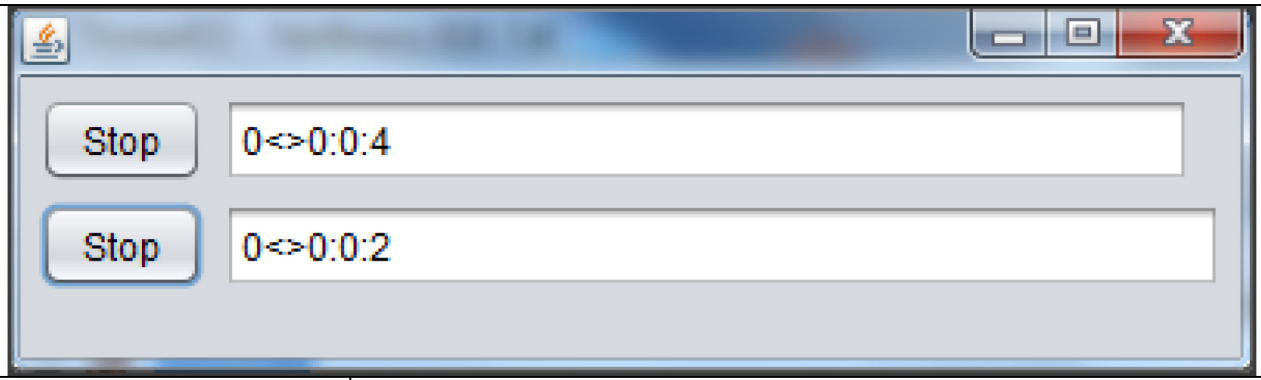
```
class StopWatch{

    int h, m, s, d;

    StopWatch() {
       h = 0;
       m = 0;
       s = 0;
    }

    public void run() {
       try {
          while (stopWatchStatus) {
             s = s + 1;
             m = m + (s / 60);
             s = s % 60;
             h = h + (m / 60);
             m = m % 60;
             d = d + (h / 24);
             h = h % 24;
             jTextField1.setText(d + "<>" + h + ":" + m +
":" + s);
             Thread.sleep(1000);
          }
       } catch (Exception e) {
          JOptionPane.showMessageDialog(null,
e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
       }

    }
```

		}
	Form decleration	boolean stopWatchStatus=false; StopWatch stopWatch=new StopWatch();

	start	if (jButton1.getText().equalsIgnoreCase("stop")) {
		stopWatchStatus=false;
		jButton1.setText("Resume");
		}else if
		(jButton1.getText().equalsIgnoreCase("Start")) {
		stopWatchStatus=true;
		stopWatch.run();
		jButton1.setText("Stop");
		}

Thread02		

	start	if (jButton1.getText().equalsIgnoreCase("stop")) {
		stopWatch.suspend();
		jButton1.setText("Resume");
		} else if
		(jButton1.getText().equalsIgnoreCase("Resume")) {
		stopWatch.resume();
		jButton1.setText("Stop");
		} else if
		(jButton1.getText().equalsIgnoreCase("Start")) {
		stopWatch.start();
		jButton1.setText("Stop");
		}

Thread03		
	Thread	class StopWatchThread implements Runnable {

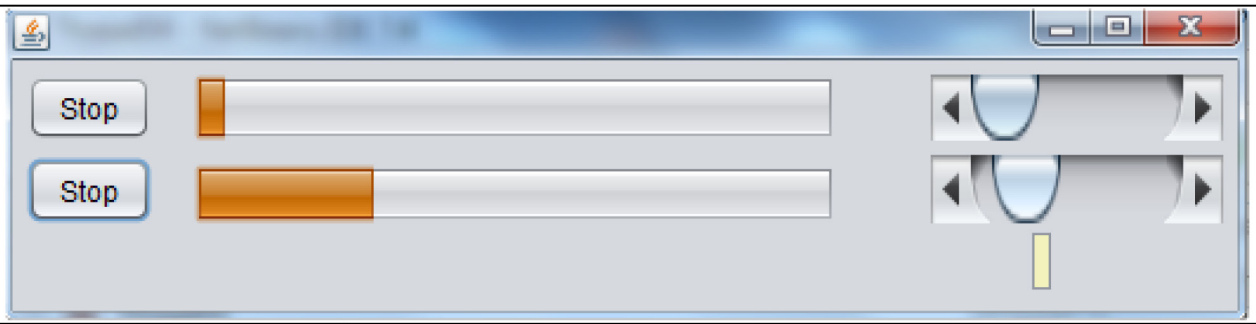
```
    int h, m, s, d;
    String name="";
    StopWatchThread(String name) {
       h = 0;
       m = 0;
       s = 0;
       this.name=name;
    }

    public void run() {
       try {
          while (true) {
             s = s + 1;
             m = m + (s / 60);
             s = s % 60;
             h = h + (m / 60);
             m = m % 60;
             d = d + (h / 24);
             h = h % 24;
             if(this.name.equalsIgnoreCase("Thread1"))
                jTextField1.setText(d + "<>" + h + ":" + m
+ ":" + s);
             else
if(this.name.equalsIgnoreCase("Thread2"))
                jTextField2.setText(d + "<>" + h + ":" + m
+ ":" + s);
             Thread.sleep(1000);
          }
       } catch (InterruptedException e) {
          JOptionPane.showMessageDialog(null,
e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
       }

    }
```

	}
Form decleration	Runnable stopWatchT1 = new StopWatchThread("Thread1"); Thread stopWatch1 = new Thread(stopWatchT1); Runnable stopWatchT2 = new StopWatchThread("Thread2"); Thread stopWatch2 = new Thread(stopWatchT2);

Button1	if (jButton1.getText().equalsIgnoreCase("stop")) {
	stopWatch1.suspend();
	jButton1.setText("Resume");
	} else if (jButton1.getText().equalsIgnoreCase("Resume")) {
	stopWatch1.resume();
	jButton1.setText("Stop");
	}else if (jButton1.getText().equalsIgnoreCase("Start")) {
	stopWatch1.start();
	jButton1.setText("Stop");
	}

Button2	if (jButton2.getText().equalsIgnoreCase("stop")) {
	stopWatch2.suspend();
	jButton2.setText("Resume");
	} else if (jButton2.getText().equalsIgnoreCase("Resume")) {
	stopWatch2.resume();
	jButton2.setText("Stop");
	}else if (jButton2.getText().equalsIgnoreCase("Start")) {
	stopWatch2.start();
	jButton2.setText("Stop");
	}

Thread04



Thread

```
class RacerThread implements Runnable {

      int progress=0;
      int delay=1000;
      String name="";
      RacerThread(String name) {
         this.name=name;
      }

      public void run() {
         try {
            while (progress<100) {
               progress++;
               if(this.name.equalsIgnoreCase("Thread1"))
               {
                  jProgressBar1.setValue(progress);
                  delay=(1000)*((100-
jScrollBar1.getValue())/100)+100;
               }
               else
if(this.name.equalsIgnoreCase("Thread2"))
               {
                  jProgressBar2.setValue(progress);
                  delay=(1000)*((100-
jScrollBar2.getValue())/100)+100;
               }
               Thread.sleep(delay);
            }
```

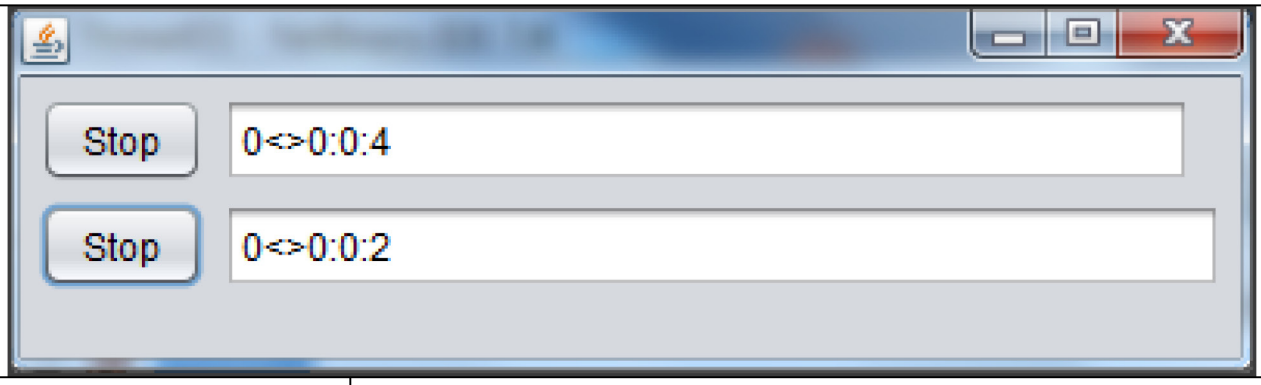
		if(this.name.equalsIgnoreCase("Thread1"))
		{
		jProgressBar1.setValue(0);
		jButton1.setText("Finished");
		jButton1.setEnabled(false);
		}
		else
		if(this.name.equalsIgnoreCase("Thread2"))
		{
		jProgressBar2.setValue(0);
		jButton2.setText("Finished");
		jButton2.setEnabled(false);
		}
		} catch (InterruptedException e) {
		JOptionPane.showMessageDialog(null,
		e.getMessage(), "Error",
		JOptionPane.ERROR_MESSAGE);
		}
		
		}
		}

	decleration	Runnable RRacerThread1 = new RacerThread("Thread1"); Thread RacerThread1 = new Thread(RRacerThread1); Runnable RRacerThread2 = new RacerThread("Thread2"); Thread RacerThread2 = new Thread(RRacerThread2);

	Button 1	if (jButton1.getText().equalsIgnoreCase("stop")) {
		RacerThread1.suspend();
		jButton1.setText("Resume");
		} else if
		(jButton1.getText().equalsIgnoreCase("Resume")) {
		RacerThread1.resume();
		jButton1.setText("Stop");

		}else if (jButton1.getText().equalsIgnoreCase("Start")) {
		RacerThread1.start();
		jButton1.setText("Stop");
		}

	Button 2	if (jButton2.getText().equalsIgnoreCase("stop")) {
		RacerThread2.suspend();
		jButton2.setText("Resume");
		} else if (jButton2.getText().equalsIgnoreCase("Resume")) {
		RacerThread2.resume();
		jButton2.setText("Stop");
		}else if (jButton2.getText().equalsIgnoreCase("Start")) {
		RacerThread2.start();
		jButton2.setText("Stop");
		}

Thread 05		

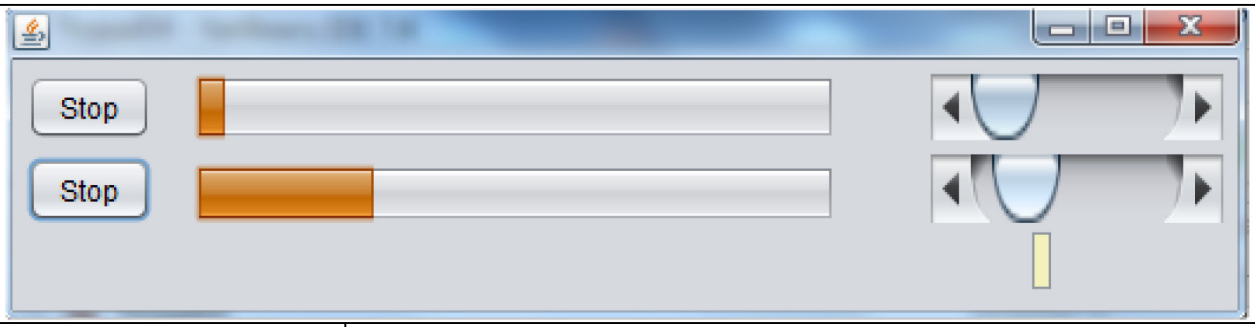
	thread	class StopWatchThread extends Thread {
		
		int h, m, s, d;
		String name="";
		StopWatchThread(String name) {
		h = 0;
		m = 0;
		s = 0;
		this.name=name;
		}

		public void run() {
		try {
		while (true) {
		s = s + 1;
		m = m + (s / 60);
		s = s % 60;
		h = h + (m / 60);
		m = m % 60;
		d = d + (h / 24);
		h = h % 24;
		if(this.name.equalsIgnoreCase("Thread1"))
		jTextField1.setText(d + "<>" + h + ":" + m + ":" + s);
		else if(this.name.equalsIgnoreCase("Thread2"))
		jTextField2.setText(d + "<>" + h + ":" + m + ":" + s);
		Thread.sleep(1000);
		}
		} catch (InterruptedException e) {
		JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
		}
		
		}
		}

	Decleration	Thread stopWatch1 = new StopWatchThread("Thread1"); Thread stopWatch2 = new StopWatchThread("Thread2");

	Button1	if (jButton1.getText().equalsIgnoreCase("stop")) {
		stopWatch1.suspend();
		jButton1.setText("Resume");
		} else if

		(jButton1.getText().equalsIgnoreCase("Resume")) {
		stopWatch1.resume();
		jButton1.setText("Stop");
		}else if
		(jButton1.getText().equalsIgnoreCase("Start")) {
		stopWatch1.start();
		jButton1.setText("Stop");
		}

Thread06		

	Thread	class RacerThread extends Thread {
		
		int progress=0;
		int delay=1000;
		String name="";
		RacerThread(String name) {
		this.name=name;
		}
		
		public void run() {
		try {
		while (progress<100) {
		progress++;
		if(this.name.equalsIgnoreCase("Thread1"))
		{
		jProgressBar1.setValue(progress);
		delay=(1000)*((100-
		jScrollBar1.getValue())/100)+100;
		}
		else

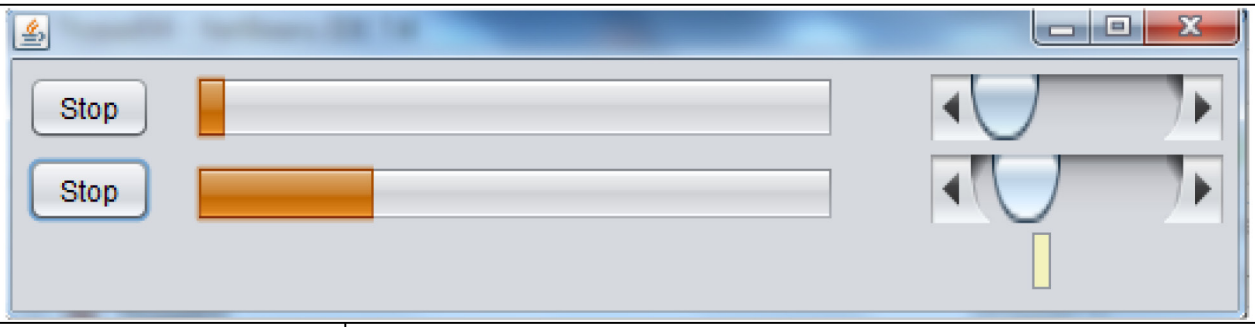
		if(this.name.equalsIgnoreCase("Thread2"))
		{
		jProgressBar2.setValue(progress);
		delay=(1000)*((100-
		jScrollBar2.getValue())/100)+100;
		}
		Thread.sleep(delay);
		}
		if(this.name.equalsIgnoreCase("Thread1"))
		{
		jProgressBar1.setValue(0);
		jButton1.setText("Finished");
		jButton1.setEnabled(false);
		}
		else
		if(this.name.equalsIgnoreCase("Thread2"))
		{
		jProgressBar2.setValue(0);
		jButton2.setText("Finished");
		jButton2.setEnabled(false);
		}
		} catch (InterruptedException e) {
		JOptionPane.showMessageDialog(null,
		e.getMessage(), "Error",
		JOptionPane.ERROR_MESSAGE);
		}
		
		}
		}

	Decleration	Thread RacerThread1 = new RacerThread("Thread1"); Thread RacerThread2 = new RacerThread("Thread2");

	button	if (jButton1.getText().equalsIgnoreCase("stop")) {
		RacerThread1.suspend();
		jButton1.setText("Resume");

		} else if (jButton1.getText().equalsIgnoreCase("Resume")) {
		RacerThread1.resume();
		jButton1.setText("Stop");
		}else if (jButton1.getText().equalsIgnoreCase("Start")) {
		RacerThread1.start();
		jButton1.setText("Stop");
		}

Thread07



Thread

```
class RacerThread extends  Thread {
      int progress=0;
      int delay=1000;
      String name="";
      RacerThread(String name) {
         this.name=name;
      }
      public void run() {
         try {
            while (true)
            {
               progress++;
               if (progress==100)
               {
                  progress=0;
                  JOptionPane.showMessageDialog(null,
name+" : I am still
alive","hi",JOptionPane.INFORMATION_MESSAGE);
```

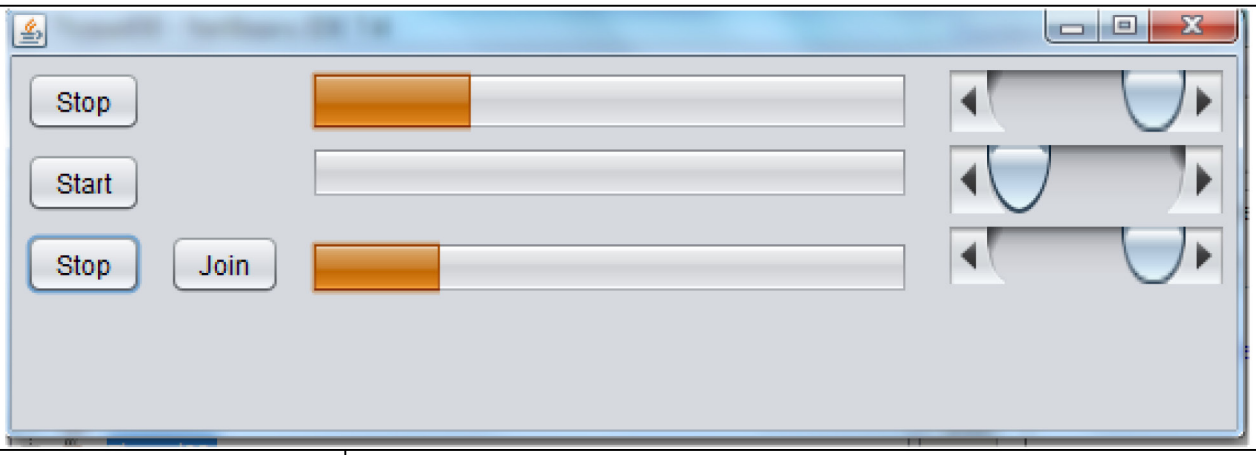
		}
		if(this.name.equalsIgnoreCase("Thread1"))
		{
		jProgressBar1.setValue(progress);
		delay=(1000)*((100-
		jScrollBar1.getValue())/100)+100;
		}
		else
		if(this.name.equalsIgnoreCase("Thread2"))
		{
		jProgressBar2.setValue(progress);
		delay=(1000)*((100-
		jScrollBar2.getValue())/100)+100;
		}
		Thread.sleep(delay);
		}
		
		} catch (InterruptedException e) {
		JOptionPane.showMessageDialog(null,
		e.getMessage(), "Error",
		JOptionPane.ERROR_MESSAGE);
		}
		
		}
		}

	Decleration	Thread RacerThread1 = new RacerThread("Thread1"); Thread RacerThread2 = new RacerThread("Thread2");

	Start button	if (jButton2.getText().equalsIgnoreCase("stop")) {
		RacerThread2.suspend();
		jButton2.setText("Resume");
		} else if
		(jButton2.getText().equalsIgnoreCase("Resume")) {
		RacerThread2.resume();
		jButton2.setText("Stop");

```
        }else if
(jButton2.getText().equalsIgnoreCase("Start")) {
            RacerThread2.start();
            jButton2.setText("Stop");
        }
```

Thread08



Thread

```
class RacerThread extends Thread {

        int progress = 0;
        int delay = 1000;
        String name = "";

        RacerThread(String name) {
            this.name = name;
        }

        public void run() {
            try {
                while (true) {
                    progress++;
                    if (progress == 100) {
                        progress = 0;
                        JOptionPane.showMessageDialog(null,
name+" : I am still alive", "hi",
JOptionPane.INFORMATION_MESSAGE);

                    }
```

		if (this.name.equalsIgnoreCase("Thread1")) {
		jProgressBar1.setValue(progress);
		delay = (1000) * ((100 - jScrollBar1.getValue()) / 100) + 100;
		} else if (this.name.equalsIgnoreCase("Thread2")) {
		jProgressBar2.setValue(progress);
		delay = (1000) * ((100 - jScrollBar2.getValue()) / 100) + 100;
		}else if (this.name.equalsIgnoreCase("Thread3")) {
		jProgressBar3.setValue(progress);
		delay = (1000) * ((100 - jScrollBar3.getValue()) / 100) + 100;
		}
		Thread.sleep(delay);
		}
		
		} catch (InterruptedException e) {
		JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
		}
		
		}
		}

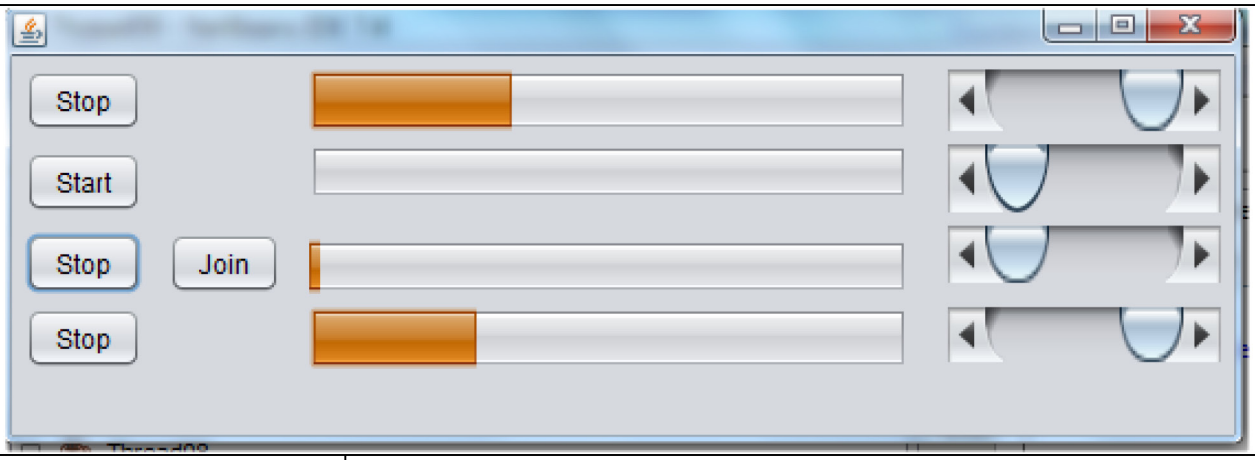
	Decleration	Thread RacerThread1 = new RacerThread("Thread1"); Thread RacerThread2 = new RacerThread("Thread2"); Thread RacerThread3 = new RacerThread("Thread3");

	Start button	if (jButton1.getText().equalsIgnoreCase("stop")) {
		RacerThread1.suspend();
		jButton1.setText("Resume");

		} else if (jButton1.getText().equalsIgnoreCase("Resume")) {
		RacerThread1.resume();
		jButton1.setText("Stop");
		} else if (jButton1.getText().equalsIgnoreCase("Start")) {
		RacerThread1.setDaemon(true);
		RacerThread1.start();
		jButton1.setText("Stop");
		}

	Join button	try {
		RacerThread3.join(5000);
		} catch (Exception e) {
		
		}

Thread09



	Thread	class RacerThread extends Thread {
		
		int progress = 0;
		int delay = 1000;
		String name = "";
		
		RacerThread(String name) {
		this.name = name;
		}
		
		public void run() {

```
        try {
            while (true) {
                progress++;
                if (progress == 100) {
                    progress = 0;
                    JOptionPane.showMessageDialog(null,
name+" : I am still alive", "hi",
JOptionPane.INFORMATION_MESSAGE);

                }
                if (this.name.equalsIgnoreCase("Thread1"))
{
                    jProgressBar1.setValue(progress);
                    delay = (1000) * ((100 -
jScrollBar1.getValue()) / 100) + 100;
                } else if
(this.name.equalsIgnoreCase("Thread2")) {
                    jProgressBar2.setValue(progress);
                    delay = (1000) * ((100 -
jScrollBar2.getValue()) / 100) + 100;
                }else if
(this.name.equalsIgnoreCase("Thread3")) {
                    jProgressBar3.setValue(progress);
                    delay = (1000) * ((100 -
jScrollBar3.getValue()) / 100) + 100;
                }
                else if
(this.name.equalsIgnoreCase("Thread4")) {
                    jProgressBar4.setValue(progress);
                    delay = (1000) * ((100 -
jScrollBar4.getValue()) / 100) + 100;
                }
                Thread.sleep(delay);
            }

        } catch (InterruptedException e) {
```

	JOptionPane.showMessageDialog(null, e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE); } } }
Decleration	Thread RacerThread1 = new RacerThread("Thread1"); Thread RacerThread2 = new RacerThread("Thread2"); Thread RacerThread3 = new RacerThread("Thread3"); Thread RacerThread4 = new RacerThread("Thread4");

Start button	if (jButton1.getText().equalsIgnoreCase("stop")) {
	RacerThread1.suspend();
	jButton1.setText("Resume");
	} else if (jButton1.getText().equalsIgnoreCase("Resume")) {
	RacerThread1.resume();
	jButton1.setText("Stop");
	} else if (jButton1.getText().equalsIgnoreCase("Start")) {
	RacerThread1.setDaemon(true);
	
	RacerThread4.setPriority(Thread.MAX_PRIORITY);
	RacerThread1.start();
	jButton1.setText("Stop");
	}

Join button	try {
	RacerThread3.join(5000);
	} catch (Exception e) {
	
	}