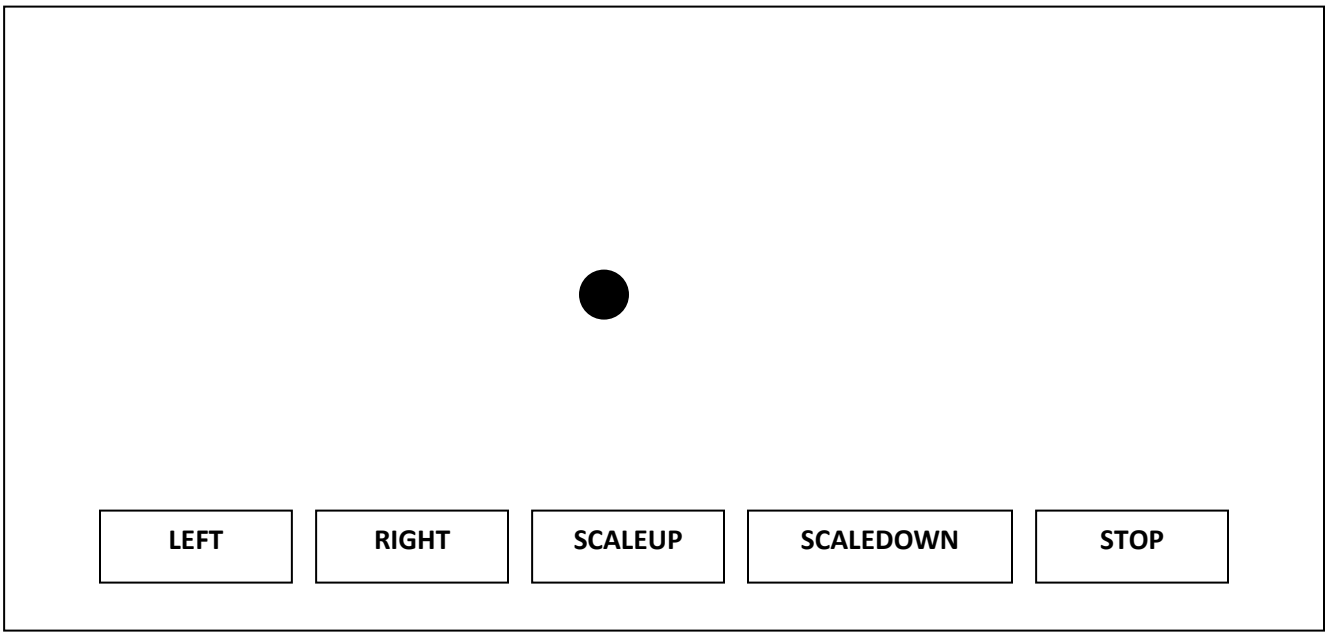
BIRLA INSTITIUTE OF TECHNOLOGY AND SCIENCE, PILANI
First Semester, 2016-17
O Object-Oriented Programming (CS F213)
LAB-10 (GUI and Multithreading)

AGENDA

- **GUI and Multithreading**

Consider the frame window shown in the following figure.



This frame window shows five buttons labeled as 'LEFT','RIGHT','SCALEUP','SCALEDOWN' & 'STOP'. There is circular icon filled with red color (of initial radius 10) also shown in the center location of the frame window. The location of the circular icon on the frame window can be moved continuously either to left one pixel position after every 100 milliseconds (via leftThread started via pressing of LEFT button) or to right one pixel position(via rightThread started via pressing of RIGHT button). Similarly the size of the icon can be increased by one after every 100 milliseconds (via scaleUpThread started via pressing of SCALEUP button) or decreased by one (via scaleDownThread started via pressing of SCALEDOWN button).

The actions that are to be carried out on behalf of pressing of these buttons is/are explain below.

- **Pressing of LEFT Button:**

When this button is pressed for the first time it starts a thread named leftThread (instance of class LeftThread). But before starting the leftThread, any other previously running thread should be suspended and then only leftThread should be started. The job of the leftThread is to move the position of circular icon to the left by 1 pixel position and then redisplay the icon on the frame window after every 100 milliseconds time interval. The leftThread will be suspended if (i) any other button has been pressed by user or

(ii) the left end of the frame window has been reached. **[For more details observe the demo video]**. If leftThread is already running at the time of pressing of LEFT Button, then this pressing will be ignored. If leftThread is in suspended state at the time of pressing of this button then following actions should be carried out

(i) Any other running thread should be suspended and
(ii) leftThread's working should be resumed.

- **Pressing of RIGHT Button:**

When this button is pressed for the first time it starts a thread named rightThread (instance of class RightThread). But before starting the rightThread, any other previously running thread should be suspended and then only rightThread should be started. The job of the rightThread is to move the position of circular icon to the right by 1 pixel position and then redisplay the icon on the frame window after every 100 milliseconds time interval. The rightThread will be suspended if (i) any other button has been pressed or (ii) the right end of the frame window has been reached. **[For more details observe the demo video]**. If rightThread is already running at the time of pressing of RIGHT Button, then this pressing will be ignored. If rightThread is in suspended state at the time of pressing of this button then following actions should be carried out

(i) Any other running thread should be suspended and
(ii) rightThread's working should be resumed

- **Pressing of SCALEUP Button:**

When this button is pressed for the first time it starts a thread named scaleUpThread (instance of class ScaleUpThread). But before starting the scaleUpThread, any other previously running thread should be suspended and then only scaleUpThread should be started. The job of the scaleUpThread is to increment the size of circular icon by 1 and then redisplay the icon on the frame window after every 100 milliseconds time interval. The scaleUpThread will be suspended if (i) any other button has been pressed or (ii) the radius of the circular icon been reached upto 50. **[For more details observe the demo video]**. If scaleUpThread is already running at the time of pressing of SCALEUP Button, then this pressing will be ignored. If scaleUpThread is in suspended state at the time of pressing of this button then following actions should be carried out

(i) Any other running thread should be suspended and
(ii) scaleUpThread's working should be resumed

- **Pressing of SCALEDOWN Button:**

When this button is pressed for the first time it starts a thread named scaleDownThread (instance of class ScaleDownThread). But before starting the scaleDownThread, any other previously running thread should be suspended and then only scaleDownThread should be started. The job of the scaleDownThread is to decrement the sizeof circular icon by 1 and then redisplay the icon on the frame window after every 100 milliseconds time interval. The scaleDownThread will be suspended if (i) any other button has been pressed or (ii) the radius of the circular icon been reached to lower limit 5. **[For more details observe the**

demo video]. If scaleDownThread is already running at the time of pressing of SCALEDOWN Button, then this pressing will be ignored. If scaleDownThread is in suspended state at the time of pressing of this button then following actions should be carried out

(iii) Any other running thread should be suspended and
(iv) scaleUpThread's working should be resumed

- **Pressing of STOP Button:**

  Pressing of STOP button will suspend any previously running thread.

Initially the state of all threads are in not started state and hence in not running state.

The following description will explain the simulated working of this application for two cases.

Case 1:

Suppose after loading the application, user presses the buttons LEFT, LEFT, RIGHT, SCALEUP, SCALEUP, LEFT, RIGHT and STOP in sequence. The following table describes the actions carried out in response to the pressing of buttons. Initially no thread is started.

Sr.No	Button Pressed	Action to be carried out
1	LEFT	Since no thread is running at this point so this action will start the leftThread
2	LEFT	Since leftThread is already running , so this pressing will be ignored
3	RIGHT	Suspends leftThread (already running) and starts rightThread (it is the first press)
4	SCALEUP	Suspends rightThread (already running) and starts scaleUpThread (it is the first press)
5	SCALEUP	Since scaleUpThread is already running so this press will be ignored
6	LEFT	Note that leftThread has been started already and only suspended temporarily earlier. So now scaleUP thread will be suspended and leftThread will have to be resumed.
7	RIGHT	Suspend leftThread, resume rightThread (already started)
8	STOP	Suspends rightThread. Note that at this step all threads are in suspended state

Case 2:

Suppose after loading the application, user presses only LEFT button only once. So in response to the pressing of LEFT button, leftThread will be started. So icon's position will start moving to the left. When icon's location reached to the left end of the window and in between no other button has been pressed, then the leftThread will suspend itself as the location of circle icon has reached up to the left end of frame window. Similarly other threads also work in same way. For example rightThread will suspend itself automatically if the location of circle icon has reached up to the right end of frame window (and no other button is pressed in between). The scaleUpThread will be suspend itself automatically when radius of the icon reaches upper limit of 50 (and no other button is pressed in between). The scaleDownThread will be suspend itself automatically when radius of the icon reaches lower limit 5 (and no other button is pressed in between).

What is Given to You?

You are given a compliable and executable source java file named ***Online.java***. The code in the file is sufficiently commented to express the desired functionality.

The ***Online.java*** file has the code for the following classes

(i) ***CircleIcon –*** This class implements Icon interface for displaying circle icon. You can add any functionality in the form of method(s) that you think is required as per question specification
(ii) ***OnlineMain –*** This is driver class. This class supports displaying of frame with circle icon with radius 10 in the center.
(iii) Four thread class(s) named ***LeftThread, RighThread, ScaleUpThread*** and ***ScaleDownThread***. All these are incomplete class(s).

Tasks you have to do in the file Online.java.

1. Add any method(s) in the CircleIcon class that you think is/are required as per the specification and required behavior.

2. Write the java code for the four thread classes named LeftThread, RightThread, ScaleUpThread and ScaleDownThread. [Note: You are not allowed to use any of the deprecated methods such as suspend(), resume(), stop() etc. methods]

3. Complete the driver class (OnlineMain) code by creating five threads named leftThread (of class LeftThread), rightThread(of class RightThread), scaleUpThread(of class ScaleUPThread) and scaleDownThread (of ScaleDownThread). You have to write the action listeners for various buttons also. Use anonymous inner classes for writing the action listeners.

**********GOOD LUCK**********