# Department of Computing

# CS 212: Object Oriented Programming

# Class: BSCS-8AB

# Lab 11: Graphics

# Date: April 29, 2019

# Instructor: Hirra Anwar

**Learning Objectives**

The learning objective of this lab is to understand and practice the usage of Jtable and Graphics library.

**Description**

**Activity #1**

The following code uses the JTable component to create a table and adding an appropriate Listener. Complete the missing code if any and execute to see the results. Explain what the result is.

**JTable table;**

**String tableHeader[]= {"Col 1","Col 2","Col 3"};**

**String tableData[][]= new String[5][3];**

**for(int i=0; i<5; i++ )**

**{**

**for(int j=0; j< 3; j++)**

**{**

**tableData[i][j]=i+","+j;**

**}**

**}**

**table = new JTable(tableData, tableHeader);**

**table.addMouseListener( new MouseAdapter()**

**{**

**public void mouseClicked(MouseEvent me)**

**{**

**int row= table.getSelectedRow();**

**int col= table.getSelectedColumn();**

**JOptionPane.showMessageDialog(null, "Selected Row = "+row+"\nSelected Col:"+col);**

**}**

**});**

**JScrollPane scrollPane = new JScrollPane(table);**

**this.add(scrollPane, BorderLayout.CENTER);**

**Graphics Description:**

**To create graphics in your code you need to explore this library.**

* A Graphics is something you can paint on.
* Default Graphics object available in paint (…) method
* repaint () method to refresh Canvas
* Drawing Geometric Figures
  + Drawing Strings
  + Drawing Lines
  + Drawing Rectangles
  + Drawing Ovals
  + Drawing Arcs
  + Drawing Polygons

The important methods related to graphics include a very extensive list you can check online however main methods are;

* g.drawString(“Hello, World”, 20, 20);
* g.drawRect(x, y, width, height);
* g.fillRect(x, y, width, height);
* g.drawOval(x, y, width, height);
* g.fillOval(x, y, width, height);
* drawArc(**int** x, **int** y, **int** width, **int** height, **int** startAngle, **int** endAngle)
* g.setColor(Color.red);

To draw things on a component, you need to define a class that extends JPanel and overrides its paintComponent method to specify what to draw.

**Activity 2**

Execute the following code.

// Add imports

public class TestPaintComponent extends JFrame

{

public TestPaintComponent()

{

add(new NewPanel());

this.setBounds(100,100,250,250);

this.setVisible(true);

}

public static void main(String[] args)

{

TestPaintComponent frame = new TestPaintComponent(); frame.setTitle("Test Paint

Component");

}

class NewPanel extends JPanel

{

protected void paintComponent(Graphics g)

{

super.paintComponent(g);

g.drawLine(0, 0, 50, 50);

g.drawString("Banner", 40, 40);

}

}

}

**Activity 3**

**Execute the following code:**

**// add imports**

**public class ControlCircle2 extends JFrame**

**{**

**private JButton jbtEnlarge = new JButton("Enlarge");**

**private JButton jbtShrink = new JButton("Shrink");**

**private CirclePanel canvas = new CirclePanel();**

**public ControlCircle2()**

**{**

**JPanel panel = new JPanel(); // Use the panel to group buttons**

**panel.add(jbtEnlarge);**

**panel.add(jbtShrink);**

**this.add(canvas, BorderLayout.CENTER); // Add canvas to center**

**this.add(panel, BorderLayout.SOUTH); // Add buttons to the frame**

**jbtEnlarge.addActionListener(new Listener());**

**jbtShrink.addActionListener(new Listener());**

**}**

**/\*\* Main method \*/**

**public static void main(String[] args)**

**{**

**JFrame frame = new ControlCircle2();**

**frame.setTitle("ControlCircle2");**

**frame.setLocationRelativeTo(null); // Center the frame**

**frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**

**frame.setSize(200, 200); frame.setVisible(true);**

**}**

**class Listener implements ActionListener**

**{**

**public void actionPerformed(ActionEvent e)**

**{**

**System.out.println(new java.util.Date(e.getWhen()));**

**if (e.getSource() == jbtEnlarge)**

**canvas.enlarge();**

**else if (e.getSource() == jbtShrink)**

**canvas.shrink();**

**}**

**}**

**class CirclePanel extends JPanel**

**{**

**private int radius = 5; // Default circle radius**

**/\*\* Enlarge the circle \*/**

**public void enlarge()**

**{**

**radius++;**

**repaint();**

**}**

**/\*\* shrink the circle \*/**

**public void shrink()**

**{**

**radius--;**

**repaint();**

**}**

**/\*\* Repaint the circle \*/**

**protected void paintComponent(Graphics g)**

**{**

**super.paintComponent(g);**

**g.drawOval(getWidth() / 2 - radius, getHeight() / 2 - radius, 2 \***

**radius, 2 \* radius);**

**}**

**}**

**}**

* **Call repaint( ) when you have changed something and want your changes to show up on the screen**
* **When you call repaint( ), Java schedules a call to update(Graphics g).**

**Lab Task #1**

Create a GUI displaying data in tabular form. This data is personal information related to a student when he/she gets enrolled in the university. Various attributes that need to be shown include Name, Age, CNIC, Father’sName, Address, Contact No., Email.

**Lab Task #2**

For task 2, add a button to the above GUI which when clicked populates some data (atleast 2 rows) in the Student Table.

**Lab Task #3**

Create a GUI with graphics of a rectangle which gets enlarged when clicked. Add appropriate labels to display information related to the created rectangle on the GUI.

**Hand in**

Upload a single word file with code and screenshot of output on LMS