Ch 6 | Graphics and Images / Animation / Multimedia.

(6.1) Intro to graphics programming, creating closable frames
(6.2) Terminating graphics program, Frame layout displaying info in frame
(6.3) Graphics object, Text and font color
(6.4) drawing ~~shapes~~ shapes from line, drawing rectangle & ovals.
(6.5) filling shapes paint mode Images.

creating closable frames

There are various technique used to closed frames in java. Three of the used ~~methods~~ technique are described below as:

(a) using JFrame.EXIT_ON_CLOSE inside of ~~setDefaultFra~~ setDefaultCloseOperation(). A System.exit(0); will be called and entire application will exit.

(b) using JFrame.DISPOSE_ON_CLOSE inside of setDefaultCloseOperation(); Here the frame will be closed and dispose but application won't exit.

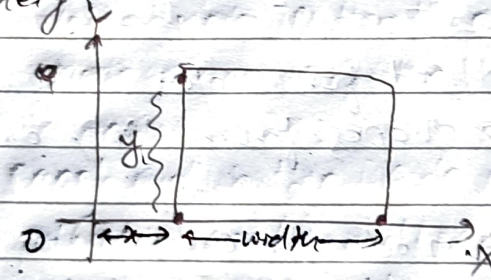
(c) using JFrame.DO_NOTHING_ON_CLOSE inside of setDefaultCloseOperation(); The frame will be closed but not disposed and won't exit.

(6.4) Drawing Lines, Rectangles & ovals

↳ Some of the methods and the parameters that those method need to draw lines, rectangles and ovals are displayed below:

(a) public void drawLine(int x1, int y1, int x2, int y2);
↳ Draws a line between coordinate (x_1, y_1) to (x_2, y_2)

(b) public void drawRect(int x, int y, int ~~height~~ width, int ~~width~~ height);



Here x, y is to represent the top left corner of rectangle and height is for rectangular height and width is for rectangular width.

(c) public void

ttfallfourb creation of balls at center that can be controlled towards left, right, top, buttom by buttons.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class TTfallfourB{
    int frameWidth = 400;
    int frameHeight = 400;
    int circleRad = 20;
    int circleX = (frameWidth - circleRad)/2;
    int circleY = (frameHeight - circleRad)/2;

    public static void main(String[] args)
    {
        TTfallfourB tfb = new TTfallfourB();
        tfb.creatGUI();
    }

    void createGUI()
    {
        JFrame frame = new JFrame("Button controlled ball");
        frame.setSize(frameWidth, frameHeight);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel panel = new JPanel()
        {@Override
```

```
Protected void paintComponent (Graphics g)
{
    super.paintComponent (g); //
    drawCircle (g); //draws the circle inbetween a marked
  }
};

//creating buttons
JButton up = new JButton ("UP");
JButton down = new JButton ("down");
JButton left = new JButton ("Left");
JButton Right = new JButton ("Right");

//Adding panel and button inside of frame
frame.getContentPane().add(panel);
frame.getContentPane().add(up, BorderLayout.NORTH);
frame.getContentPane().add(down, BorderLayout.SOUTH);
frame.getContentPane().add(left, BorderLayout.WEST);
frame.getContentPane().add(Right, BorderLayout.EAST);

//Adding ActionListener to button
up.addActionListener(new ActionListener()
    { public void actionPerformed (ActionEvent e)
      {
        circleY = circleY - 10;
        panel.repaint();
      }
    }

down.addActionListener(new ActionListener()
    { public void actionPerformed (ActionEvent e)
      {
        circleY = circleY + 10;
        panel.repaint();
```

```
}
}

left.addActionListener(new ActionListener()
{ public void actionPerformed(ActionEvent e)
  {
    circleX = circleX - 10;
    panel.repaint();
  }
}
right.addActionListener(new ActionListener()
{
  public void actionPerformed(ActionEvent e)
  {
    circleX += 10;
    panel.repaint();
  }
}
}
void drawCircle(Graphics g)
{
  g.setColor(Color.RED);
  g.fillOval(circleX, circleY, circleRad, circleRad);
}
}
```

// creating a ball which can be moved left, right, up, down on clicking the buttons.

(32)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class DynamicBall
{
    int frameHeight = 400;
    int frameWidth = 400;
    int circleX = frameWidth -
    int circleRad = 20;
    int circleX = (frameWidth - circleRad)/2;
    int circleY = (frameHeight - circleRad)/2;

    void createGUI(){
    {
        JFrame frame = new JFrame("Dynamic Ball with
                                   controller button");
        frame.setSize(frameWidth, frameHeight);
        frame.setVisible(true)
        frame.setLocationRelativeTo(null);

    // Button creation
        JButton up = new JButton("up");
        JButton down = new JButton("Down");
        JButton up = new JButton("up
```

(63)

Date
Page

TwentyOneSpringFourB.java

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

public class TwentyOneSpringFourB extends JPanel
{
    int javaStudent = 20;
    int phpStudent = 35;
    int pythonStudent = 30
    int golangStudent = (100 - (javaStudent + phpStudent + python
                                Student));

public static void main(String[] args
{
    TwentyOneSpringFourB tfb = new TwentyOneSpringFour
    B();
    tfb.creatAndShowGUI();

}
void creatAndShowGUI()
{
    JFrame frame = new JFrame("Band diagram");
    frame.setSize(400,400);
    frame.setVisible(true);
    frame.setLocationRelativeTo(null);
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.getContentPane.add(new TwentyOneSpringFourB());

@override
public void paintComponent(Graphics g)
    { super.paintComponent(g);
```

```
int totalStudent = 200;
int barWidth = getWidth / 4;
int barMaxHeight = getHeight() - 50;

int x = 50;
int y = getHeight() - 20;

drawBar(g, x, y, barWidth, (javaStudent * barMaxHeight) / totalStudent
        Color.BLUE, "Java");
drawBar(g, x + barWidth, y, barWidth, (phpStudent * barMax
        Height) / totalStudent,
        Color.GREEN, "PHP");

drawBar(g, x + (2 * barWidth), y, barWidth, (pythonStudent * barMaxHeight)
        / totalStudent,
        Color.RED, "Python");

drawBar(g, x + 3 * barWidth, y, barWidth, (golangStudent * barMaxHeight)
        / totalStudent,
        Color.BROWN, "Golang");

}

void drawBar(Graphics g, int x, int y, int barWidth,
        int barHeight, Color color, String label)
{
    g.setColor(color);
    g.fillRect(x, y - height, width, height);
    g.setColor(Color.BLACK);
    g.drawString(label, x + barWidth / 2 - 10, y + 15);
}
}
}
```

```
Pie chart
pd. create
}

void createG
{
    JFrame f
    f. setBa
    f. setVi
    f. setD
    f. setL
    f. getC
}
```

Date
Page

2nd fall 15a

17. 248 student are total, where 25 supports virtual classroom, 15 supports physical, & 8 are neutral. Here we gotta create a pie chart to illustrate it:

Soln:

```
import javax.swing.*;
import java.awt.*;

public class PieChartDemo extends JPanel
{
    int supportiveStd = 25;
    int againstStd = 15;
    int neutralStd = 8;
    int totalStd = supportiveStd + againstStd +
                   neutralStd;
    public static void main (String[] args)
    {
        PieChartDemo pd = new PieChartDemo();
        pd.createGUI();
    }
void createGUI()
{
    JFrame f = new JFrame ("Pie chart GUI build");
    f.setSize(400, 400);
    f.setVisible(true);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setLocationRelativeTo(null);
    f.getContentPane().add(new PieChartDemo());
}
```

totalStudent,

* barMaxHeight)
totalStudent

barMaxHeight)
totalStudent

```
@Override
public void paintComponent(Graphics g)
{
    super.paintComponent(g);

    int y = getHeight()/2;
    int x = getWidth()/2;
    int rad = Math.min(x, y);

    drawPie(g, x, y, rad, 0, (360 * supportiveStd / totalStd), Color.GREEN);
    drawPie(g, x, y, rad, (360 * supportiveStd / totalStd), 360 * (supportiveStd + againstStd) / totalStd), Color.BLUE);
    drawPie(g, x, y, rad, (360 * supportive+againstStd / totalStd), 360, Color.BLACK);
}

void drawPie(Graphics g, int x, int y, int radius,
    int startAngle, int endAngle, Color color)
{
    g.setColor(color);
    g.fillArc(x-radius, y-radius, radius*3, radius*3, startAngle, endAngle-startAngle);
}
}
```

```
1gsp4b | Nepals
1) import java
import java.
public clas
{
    public st
    {
        JFram
        fra
        fra
        .fra
    }
}
```

graph.h | Nepali flag

```
import javax.swing.*;
import java.awt.*;
public class NepalFlag extends JFrame
{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame("Nepali Flag");
        frame.setSize(500, 500);
        frame.setVisible(true);
        frame.setLocationRelativeTo(null);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_
Color.BLUE);   CLOSE);
color.BLACK);   NepalFlag nf = new NepalFlag();
        frame.add(nf);
    }
    public void paint(Graphics g)
    { // for left bar of flag (sammaine lathi)
        g.setColor(Black);
        g.fillRect(10, 20, 10, 200);

        // for the outer blue polygon
        g.setColor(Color.Blue);
        int[] x = {20, 200, 700, 240, 20, 20};
        int[] y = {20, 150, 150, 300, 300, 20};
        g.fillPolygon(x, y, 6);

        // for inner red polygon
        int[] xx = {30, 170, 75, 275, 30, 30};
        int[] yy = {40, 140, 140, 290, 290, 3};
        g.setColor(Color.RED);
        g.fillPolygon(xx, yy, 6);
```

```
// for sunshape
g.setColor(Color.WHITE);
g.fillArc(50, 80, 40, 40, 0, -180);

// for moon
g.fillOval(50, 200, 50, 50);

}

}
```