

# Java Programming

## Graphical Objects

*Arash Habibi Lashkari*  
*Ph.D. Candidate of UTM University*  
*Kuala Lumpur, Malaysia*

***All Rights Reserved © 2010, [www.ahlashkari.com](http://www.ahlashkari.com)***

# Intro

- Graphic operations are supported by package *java.awt*.
- Facilities include: drawing different types of colored shapes, displaying images, animation

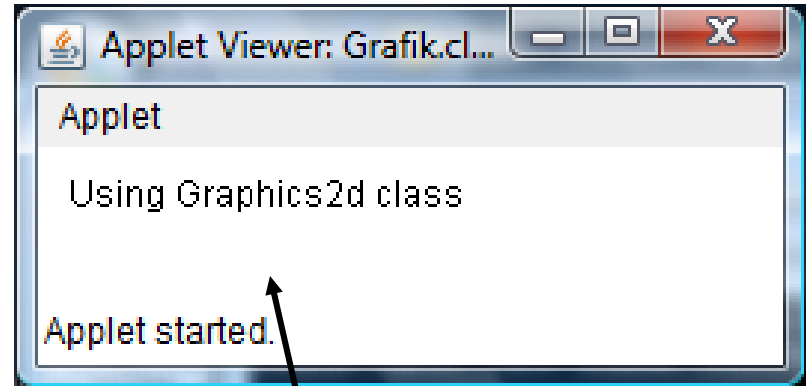
# Graphics and Graphics2D classes

- Package `java.awt` provides the class `Graphics`
- An instant of `Graphics` object contains information to perform graphic operations
- Graphic operations are executed by sending specific messages to graphic objects, including:
  - `drawString(String, int, int)`
  - `drawRect(int, int, int, int)`
  - `setColor(Color)`
  - `setFont(Font)`

# Graphics2D Class

- Java 1.2 provides the class `Graphics2D`
- It is a subclass of `Graphics`.
- Contains extended features which are more sophisticated.
- method `paint()` still receive `Graphics` object as parameter.

# Class Graphics2D



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

```
public class Graphic_01 extends JApplet {
```

```
    public void paint(Graphics g) {  
        Graphics2D g2d = (Graphics2D) g;  
        g2d.drawString("Using Graphics2d class", 10, 20);  
    }  
}
```

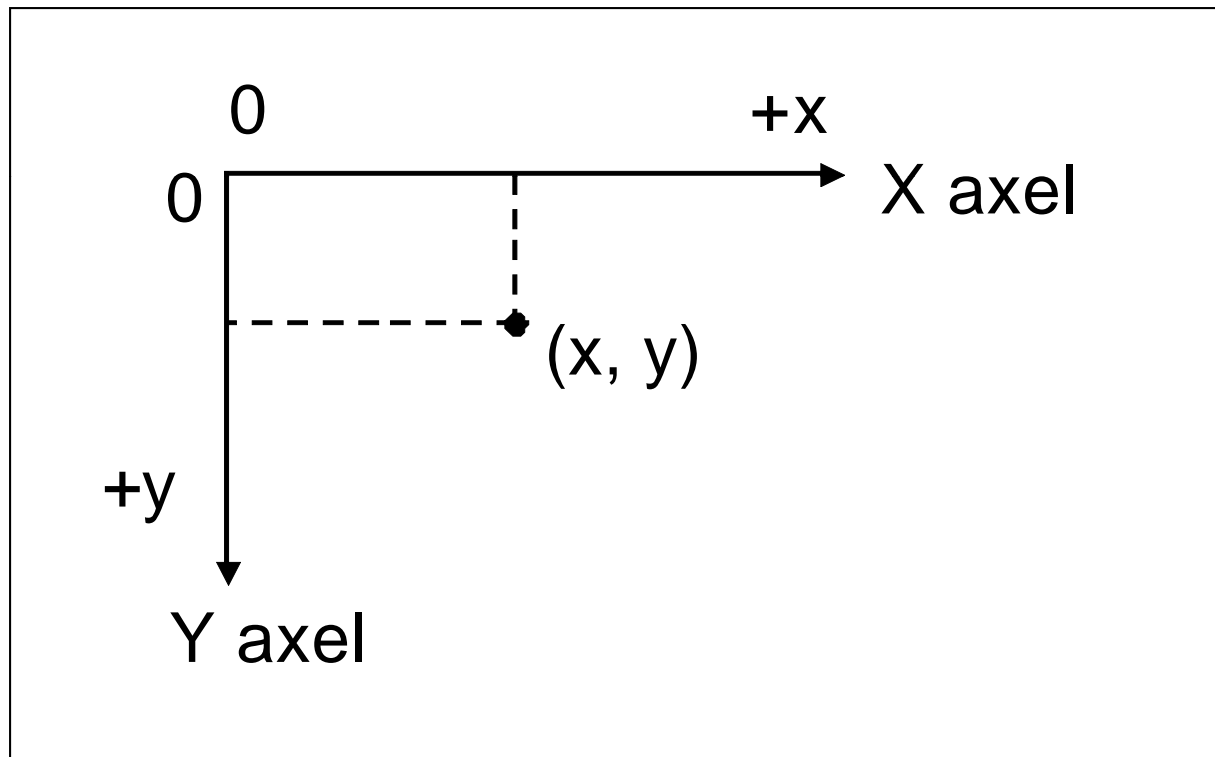
Staring at  
location (10,20)

Cast object g so that we may use  
Characteristics of *Graphics2d*

# Coordinate System

- Every *Container* object has its own coordinate system.
- Any point in a container may be represented as  $(x, y)$  coordinate.
- In AWT, value of  $x$  and  $y$  for  $(x, y)$  coordinate must be of type integer.

# Coordinate System

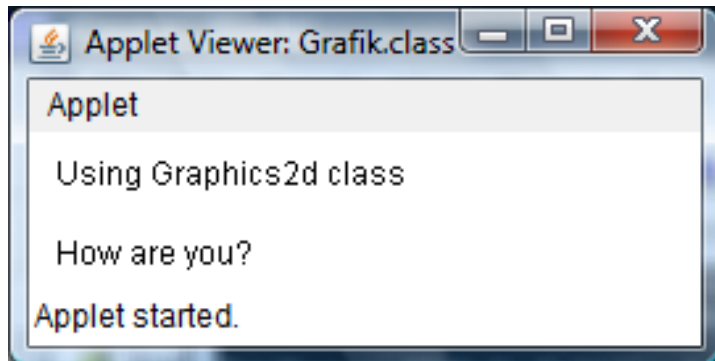


# Color and Text

- To display a text  $s$  at coordinate  $(x, y)$ , message :

`drawString(String s, int x, int y)` can be sent to object *Graphics*.

- `g2d.drawString("How are you?", 10, 50);`



"How are you"  
at location (10,50)

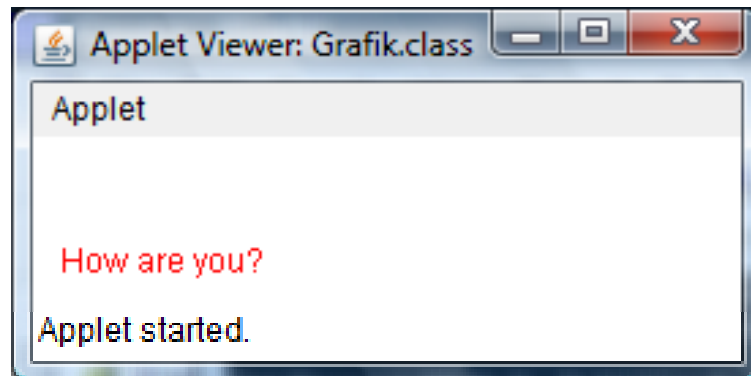


# Color and Text

- By default, object *Graphics* is set to draw text in black. For other colours, we need to send message `setColor(Color)` to the object first.

```
g2d.setColor(Color.red);
```

```
g2d.drawString("How are you?", 10, 50);
```



# Class Color

- There are several colours already defined in the library:

*Color.*

*Color.gray*

*Color.black*

*Color.pink*

*Color.orange*

*Color.green*

*Color.cyan*

*Color.lightGray*

*Color.darkGray*

*Color.red*

*Color.blue*

*Color.yellow*

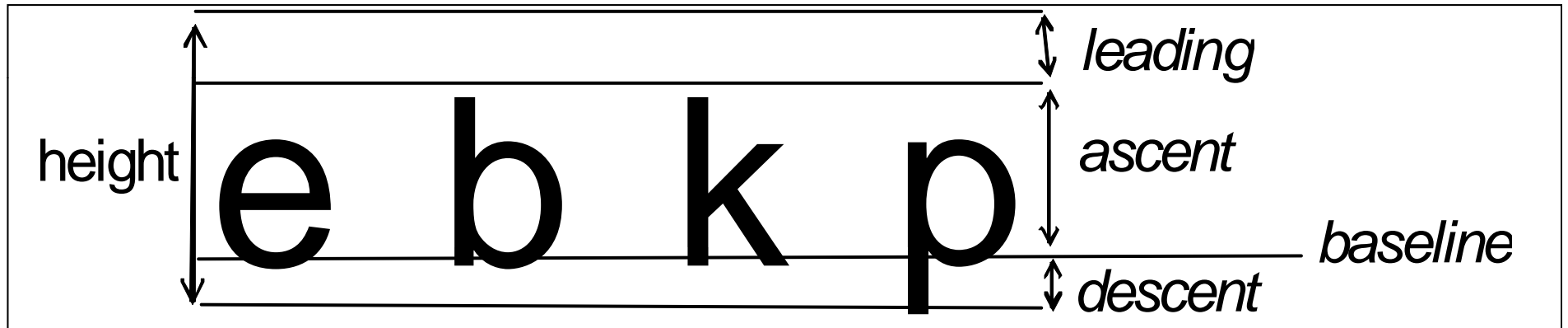
*Color.magenta*

# Class FontMetrics

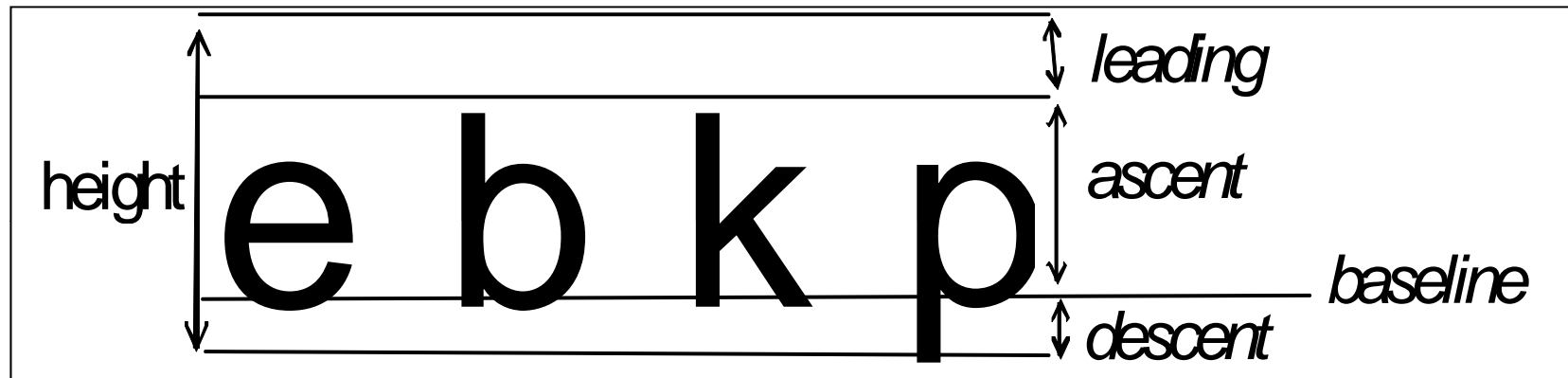
- The Component class has a method `getFontMetrics(Font)` which returns object `FontMetrics` for a specific font.

```
Font font = new Font("TimesRoman", Font.PLAIN, 14);  
FontMetrics fm = getFontMetrics(font);
```

# Class FontMetrics

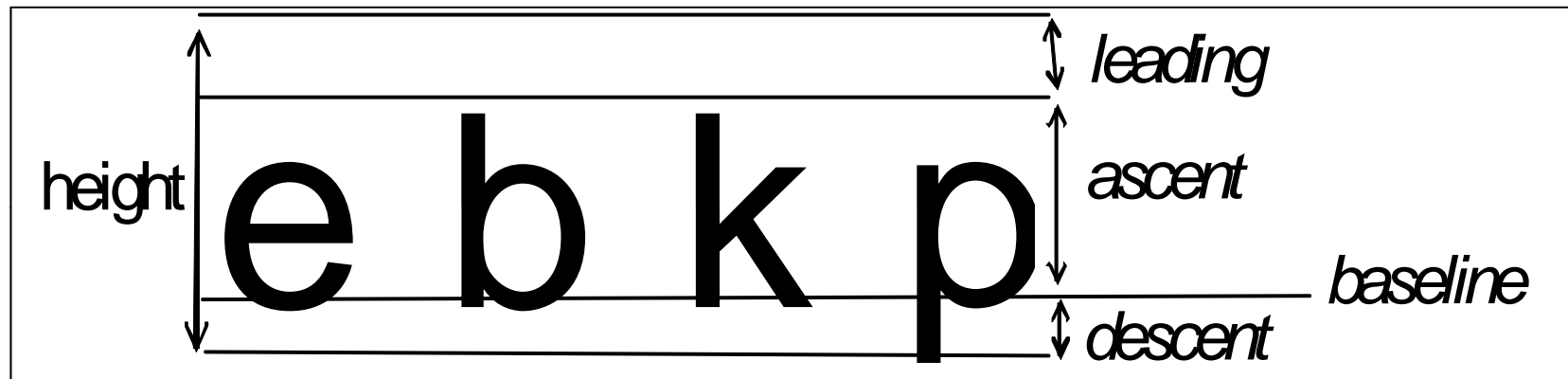


# Class FontMetrics



- Amongst the methods/messages that can be sent to object *FontMetrics*:
  - **int stringWidth(String str)**  
Get the width of string *str* for a specific font
  - **getAscent()**  
Get the *ascent* value for a specific font
  - **getDescent()**  
Get the *descent* value for a specific font
  - **int getHeight()**  
Get the height value for a specific font

# Class FontMetrics



- `int getLeading()`  
Get the *leading* value for a specific font
- `int getMaxAscent()`  
Get the maximum *ascent* value (i.e. *ascent* + *leading*) for a specific font
- `int getMaxDescent()`  
Get the maximum *descent* value (i.e. *descent* + *leading*) for a specific font

# Example..

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

public class FontMetric extends JApplet {
    public void paint(Graphics g) {
        Graphics g2d = (Graphics2D) g;

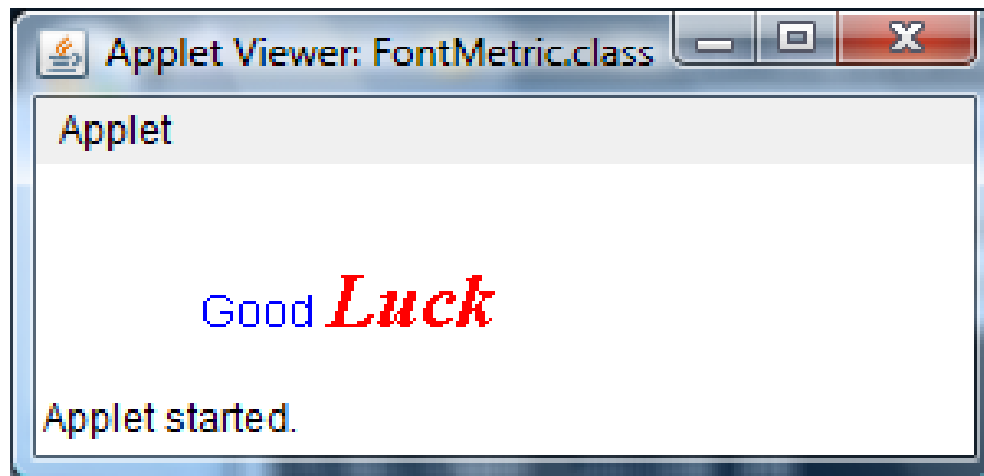
        Font font = new Font("SanSerif", Font.PLAIN, 14);
        FontMetrics metric = getFontMetrics(font);

        String str1 = "Good ";
        int str1width = metric.stringWidth(str1);
        g2d.setFont(font);
        g2d.setColor(Color.blue);
        g2d.drawString(str1, 50, 50);
        g2d.setFont(new Font("Serif", Font.BOLD+Font.ITALIC, 24));
        g2d.setColor(Color.red);
        g2d.drawString("Luck", 50+str1width, 50);
    }
}
```

Get str1width

Draw luck at  
Location after str1

# Example





# Drawing Basic Shapes

- Java2D provides a *Shape* class hierarchy which contains several classes for different shapes. Some of the subclasses of class *Shape*:
  - *Line2D*,
  - *Rectangle2D*, *RoundRectangle2D*,
  - *Ellipse2D*, *Arc2D*, *QuadCurve2D*,
  - *CubicCurve2D*, *GeneralPath*.

# Drawing Basic Shapes

- Following messages can be sent to object Graphics2D to draw and colour the object Shape:

- `void draw(Shape s)`

Draw the frame of shape s

- `void fill(Shape s)`

Colour the inner area of shape s

# Lines

- A line can be represented by object `Line2D.Float` or `Line2D.Double`.
- Both classes `Line2D.Float` and `Line2D.Double` are inner classes of class `Line2D`.
- For `Line2D.Float`, the end point is of type float
- Where as for `Line2D.Double`, the end point is of type double.

# Lines

- *To create an object Line2D with end point (5, 10) and (25, 40):*

`new Line2D.Float(5, 10, 25, 40)`    OR  
`new Line2D.Double(5, 10, 25, 40)`

- *Object Line2D can be created without any parameter. In this case, the end point are (0,0) and (0, 0)*

# Lines

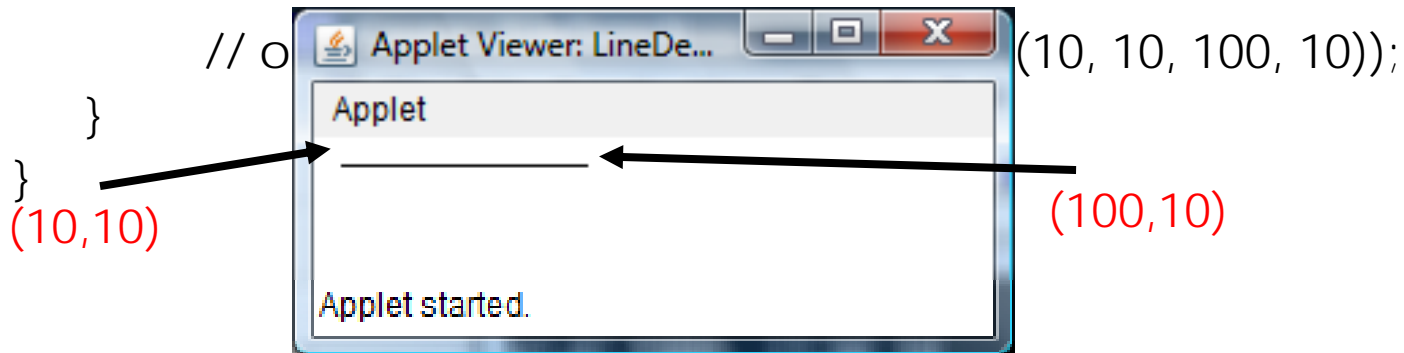
- *The following messages are understood by object Line2D:*
  - `setLine(float x1, float y1, float x2, float y2)`
  - `setLine(double x1, double y1, double x2, double y2)`
- *Will set the end point coordinate to (x1, y1) and (x2, y2)*

# Example

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

```
public class LineDemo1 extends JApplet {  
    public void paint(Graphics g) {
```

```
        Graphics2D g2d = (Graphics2D) g;  
        Line2D.Double line = new Line2D.Double(10, 10, 100, 10);  
        g2d.draw(line);
```



# Example..

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

public class LineDemo2 extends JApplet {
    private static final int distance = 10;

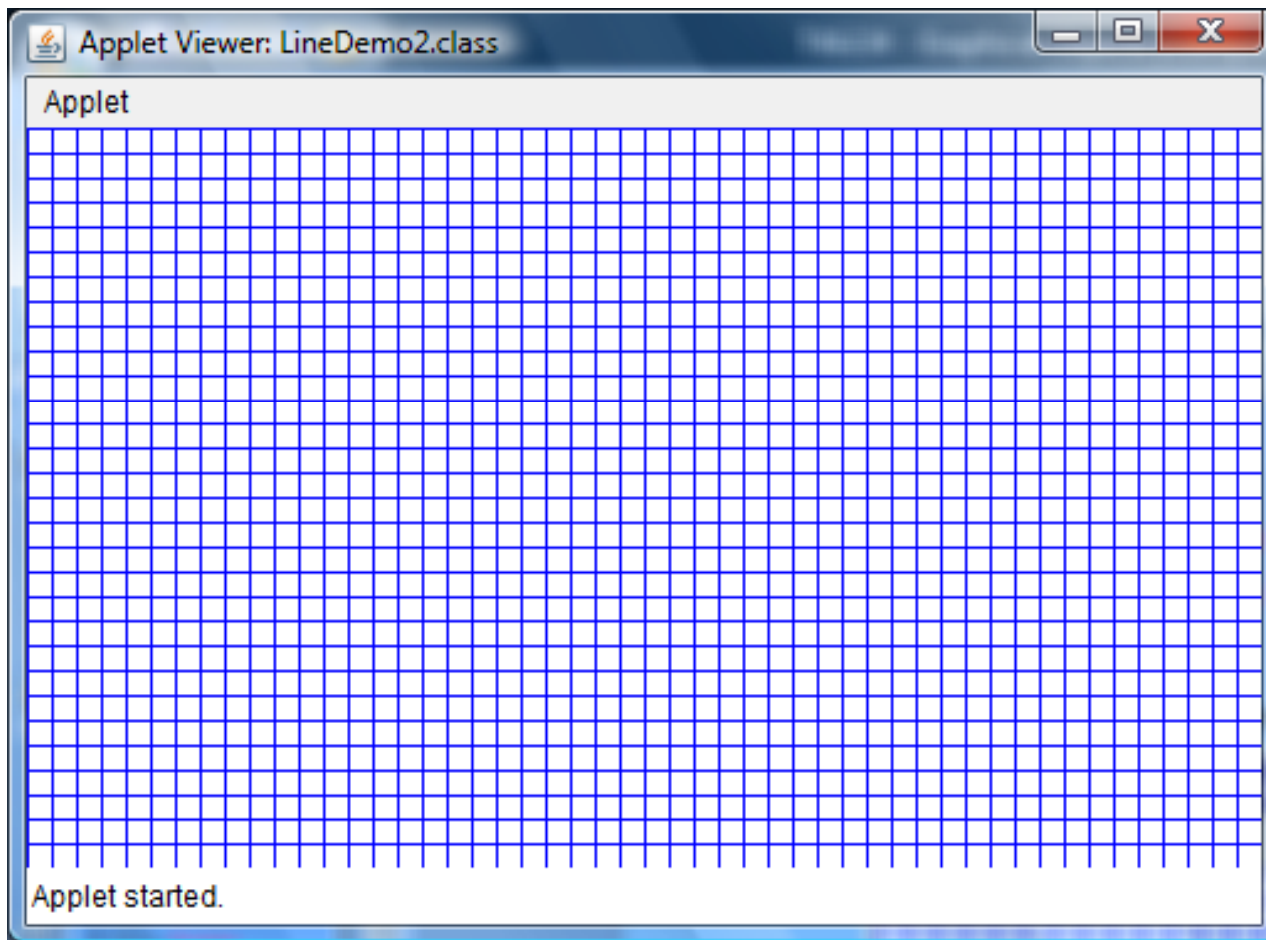
    public void paint(Graphics g) {
        Line2D.Double line = new Line2D.Double();
        Graphics2D g2d = (Graphics2D) g;
        g2d.setColor(Color.blue);
        int appletWidth = getSize().width;
        int appletHeight = getSize().height;

        for (int i=0; i < appletHeight; i += distance)
            line.setLine(0, i, appletWidth, i);
            g2d.draw(line);
        for (int i=0; i < appletWidth; i += distance) {
            line.setLine(i, 0, i, appletHeight);
            g2d.draw(line);
        }
    }
}
```

Draw vertical line

Draw horizontal line

# Example..





# Line Thickness

- The thickness of a drawn line can be set by replacing the default object **Stroke** for object Graphics2D with a new object Stroke

```
Graphics2D g2d = (Graphics2D) g;  
Line2D.Double line = new Line2D.Double(10, 10, 100, 10);  
g2d.draw(line);  
g2d.setStroke(new BasicStroke(10));  
line = new Line2D.Double(10, 30, 100, 30);  
g2d.draw(line);
```

# Example

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

public class LineDemo3 extends JApplet {
    public void paint(Graphics g) {

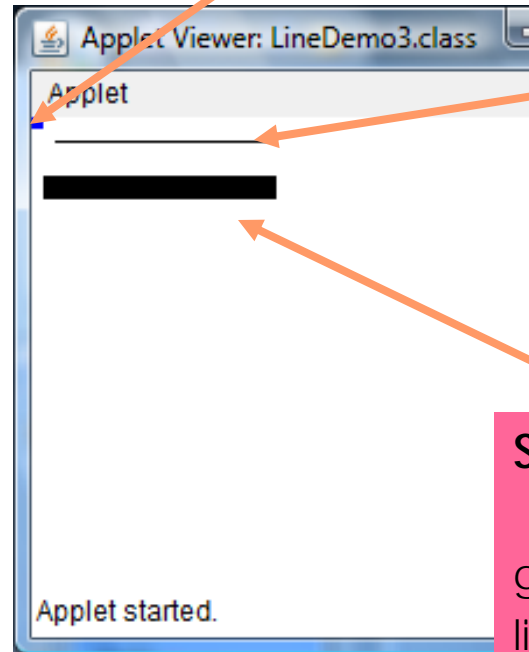
        Graphics2D g2d = (Graphics2D) g;
        Line2D.Double line = new Line2D.Double(10, 10, 100, 10);
        g2d.draw(line);
        g2d.setStroke(new BasicStroke(10));
        line = new Line2D.Double(10, 30, 100, 30);
        g2d.draw(line);
        line = new Line2D.Double(0, 0, 0, 0);
        g2d.setColor(Color.blue);
        g2d.draw(line);
    }
}
```

# Line Width Thickness

Third line is a point at (0,0) with width 10, and coloured blue  
`line = new Line2D.Double(0, 0,0,0);`  
`g2d.setColor(Color.blue);`  
`g2d.draw(line);`

## First line

`Line2D.Double line = new`  
`Line2D.Double(10, 10, 100, 10);`  
`g2d.draw(line);`



## Second line drawn using object *BasicStroke* with width 10

`g2d.setStroke(new BasicStroke(10));`  
`line = new Line2D.Double(10, 30, 100, 30);`  
`g2d.draw(line);`

# Rectangle

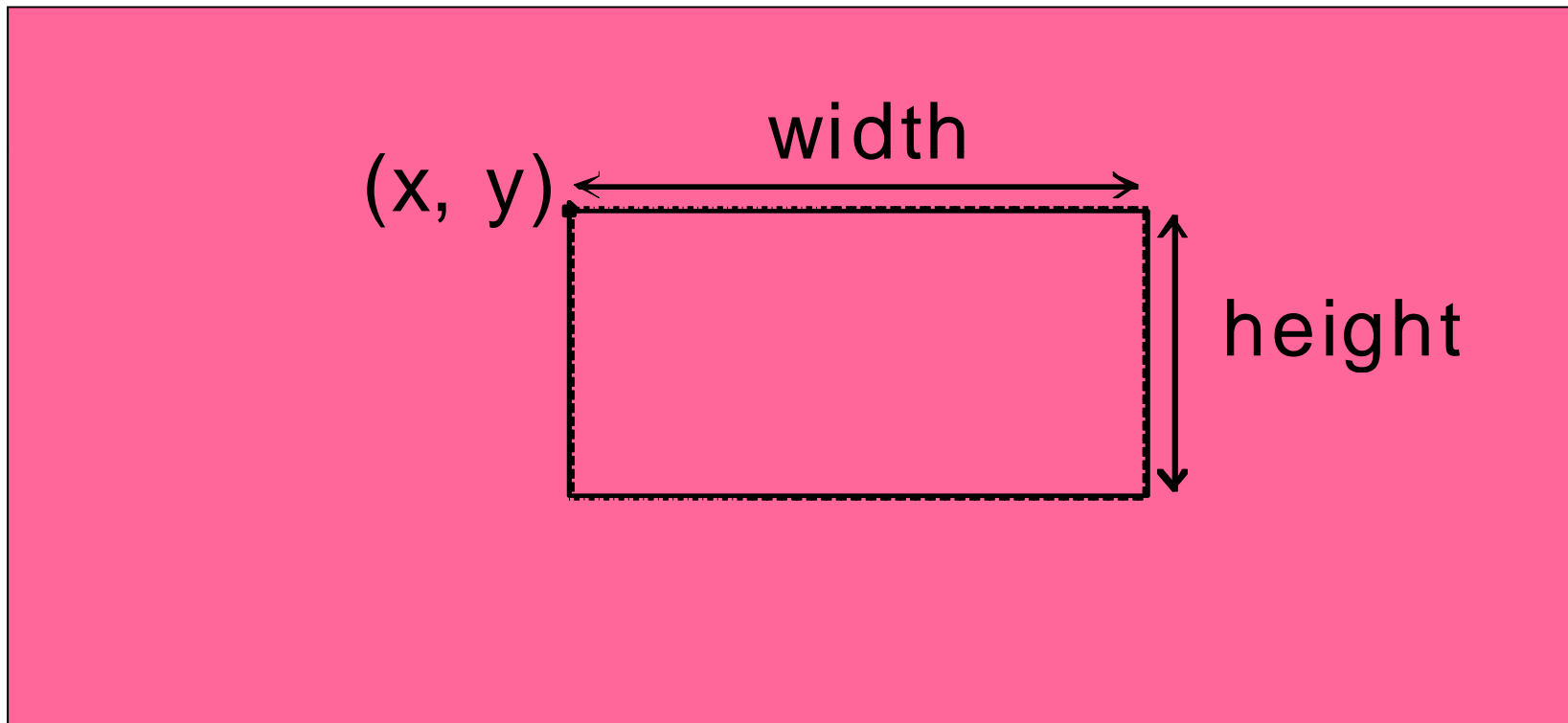
- *Class Rectangle2D is used for drawing a rectangle. It has 2 inner classes : Rectangle2D.Float and Rectangle2D.Double.*
- *To create object Rectangle2D, use :*
  - `new Rectangle2D.Float(x, y, width, height)`
  - OR**
  - `new Rectangle2D.Double(x, y, width, height)`

# Rectangle

- The following messages are understood by object Rectangle2D:
  - `setRect(float x, float y, float w, float h)`
  - `setRect(double x, double y, double w, double h)`

To set the top left corner (x, y) and width and height of the rectangle to w and h.

# Rectangle



# Example..

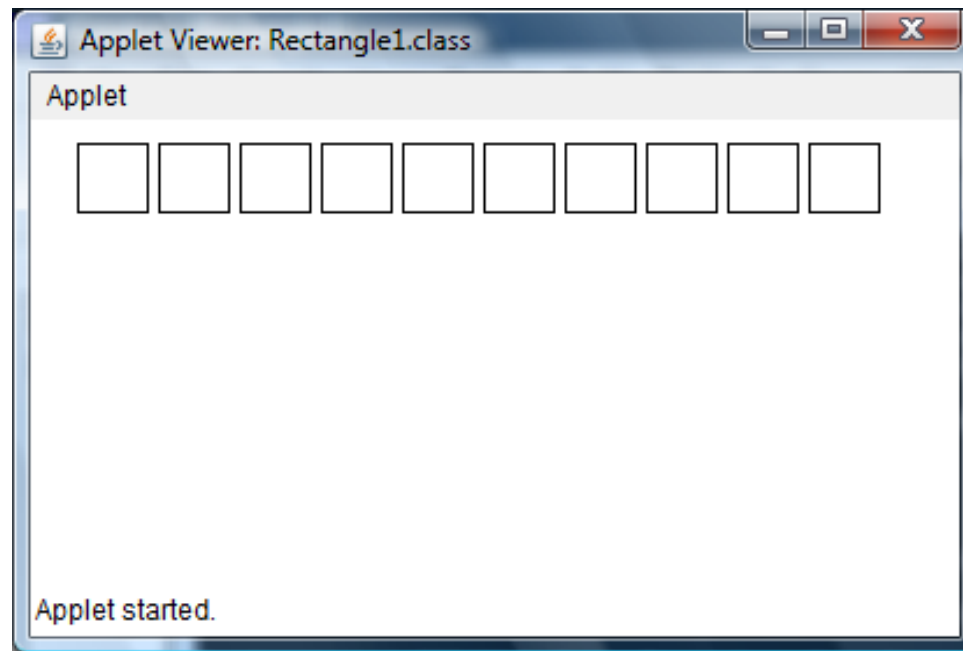
```
import java.awt.*;
import javax.swing.*;
import java.awt.geom.*;
public class Rectangle1 extends JApplet {
private static int SIZE = 30;
private static int DISTANCE = 5;
private static int X_INIT = 20;
private static int Y_ROW = 10;
private static int NUMRECT = 10;

public void paint(Graphics g) {
Graphics2D g2d = (Graphics2D) g;
int x = X_INIT;
Rectangle2D.Double rect = new Rectangle2D.Double();
for (int i=1; i <= NUMRECT; i++) {
rect.setRect(x, Y_ROW, SIZE, SIZE);
g2d.draw(rect);
x += (SIZE+DISTANCE);
}}}
```

Draw rectangle

Increase distance  
in x direction

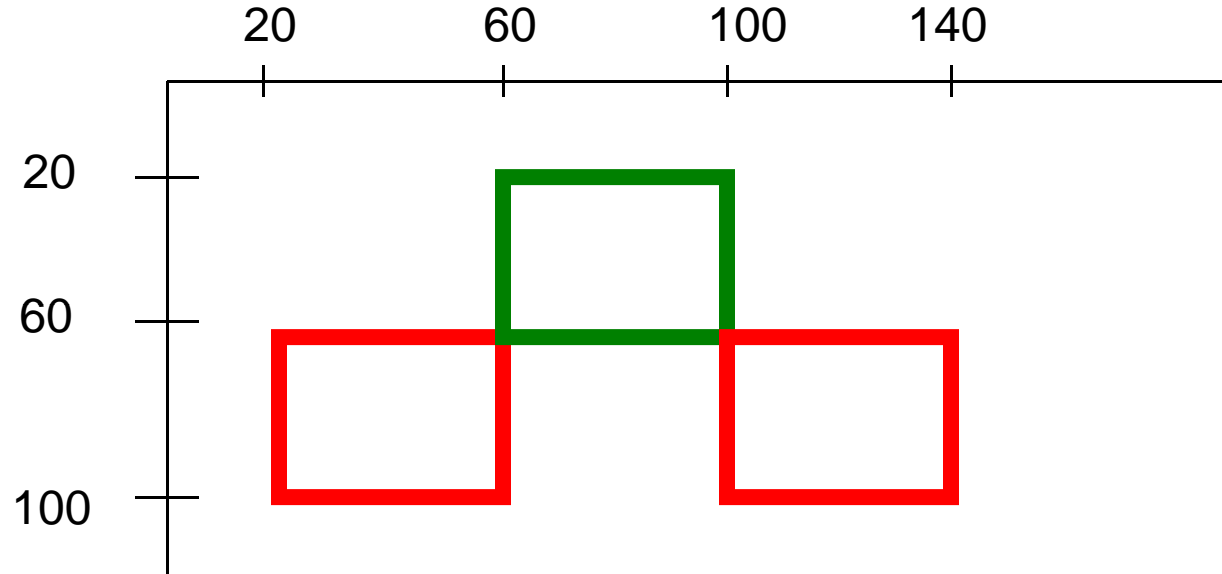
# Output:





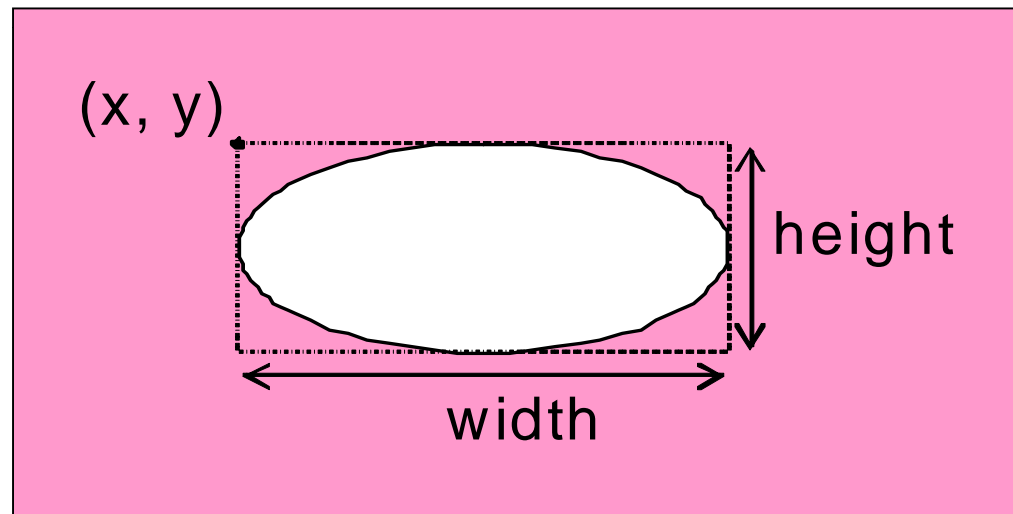
# Exercise

- Given the following shapes:



- Write a java program to draw these shapes
- Find out how to use `RoundRect` to draw rectangle with rounded corners

# Ellipse



# Ellipse

- Used to draw **circles** with **height = width**
- To create object ellipse:
  - `new Ellipse2D.Float(x, y, width, height)`
  - OR
  - `new Ellipse2D.Double(x, y, width, height)`

# Ellipse

- The following messages are understood by object `Ellipse2D`:
  - `setFrame(float x, float y, float w, float h)`
  - `setFrame(double x, double y, double w, double h)`
- To set the top left corner, width and height as  $(x, y)$ ,  $w$  and  $h$ .

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

```
public class Ellipse extends JApplet {
    private static int SIZE = 50;
    static int DISTANCE = 10;
    static int X_INIT = 20;
    static int Y_ROW = 10;
    static int NUMELLIPSE = 5;
```

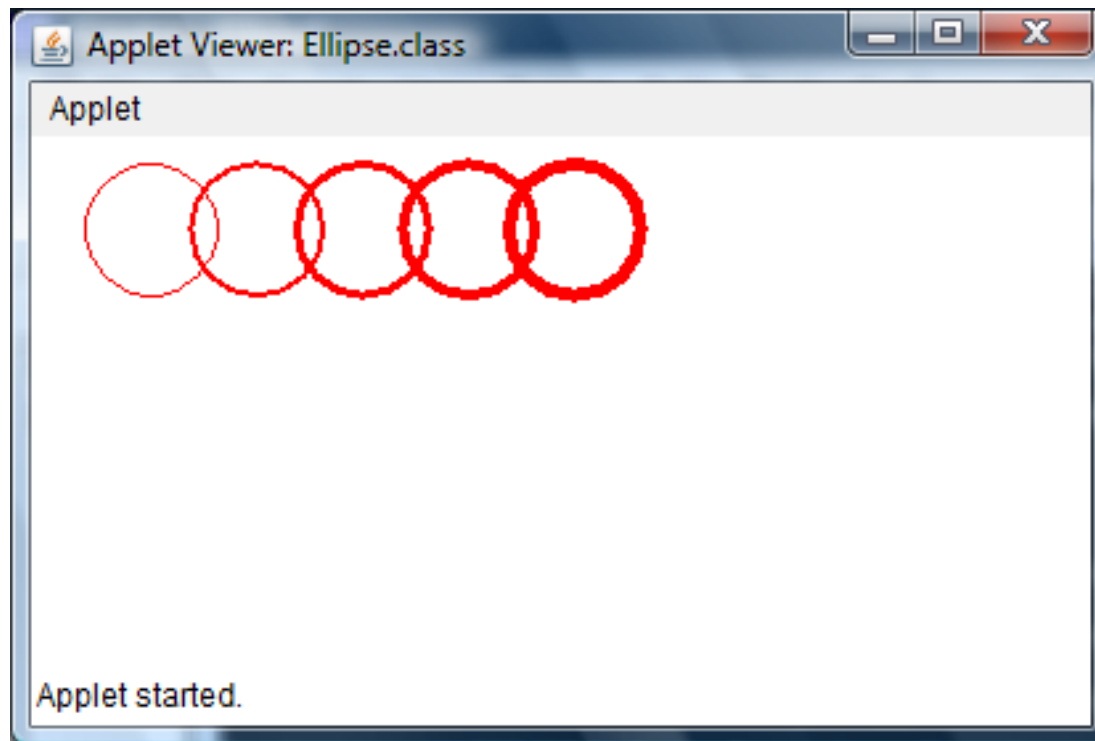
Increase thickness

Draw ellipse

```
public void paint(Graphics g) {
    Graphics2D g2d = (Graphics2D) g;
    int x = X_INIT;
    Ellipse2D.Double elips = new Ellipse2D.Double();
    for (int i=1; i <= NUMELLIPSE; i++) {
        elips setFrame(x, Y_ROW, SIZE, SIZE);
        g2d.setColor(Color.red);
        g2d.setStroke(new BasicStroke(i));
        g2d.draw(elips);
        x += (SIZE-DISTANCE);
    }
}
```

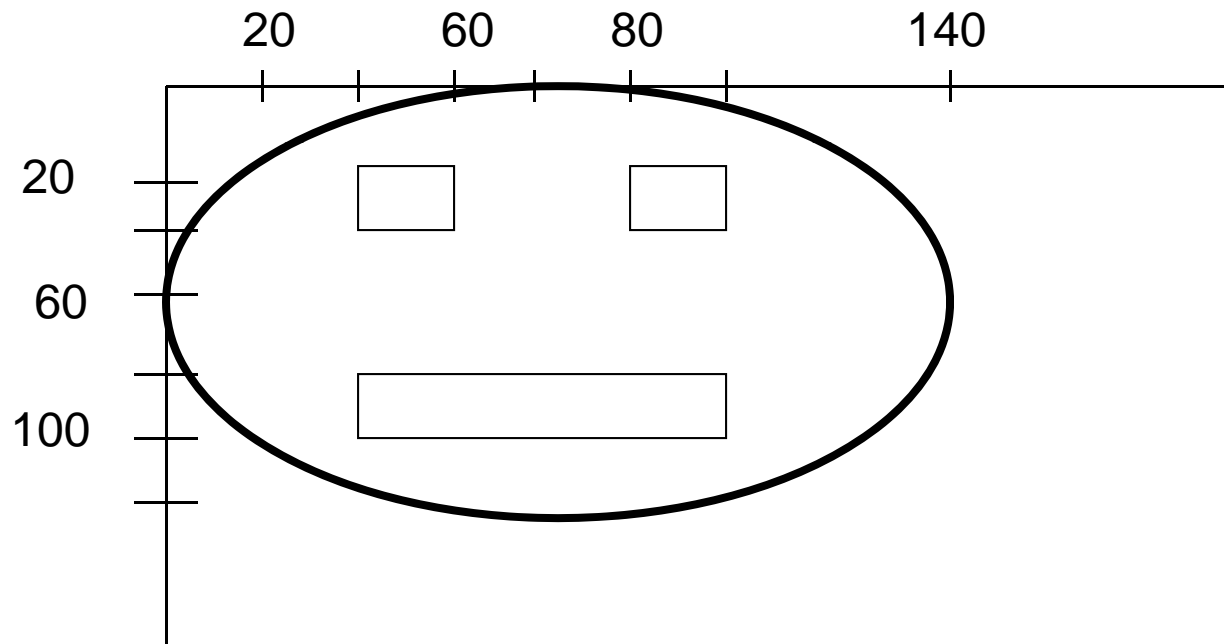
Increase distance

# Output:



# Exercise

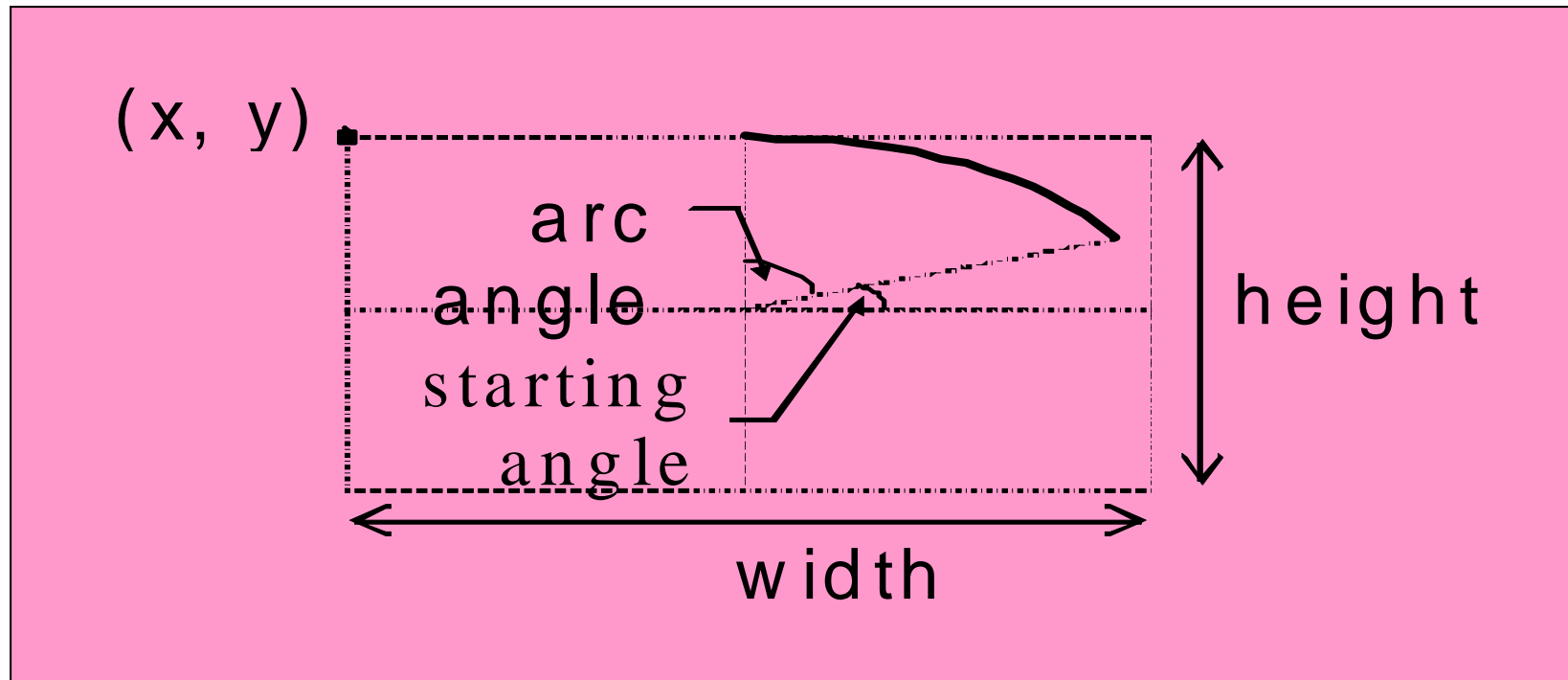
- Given the following shapes:



- Write a java program to draw these shapes

# Arc

- Arcs can be represented by *Arc2D* object





# Arc

- To create an arc object:

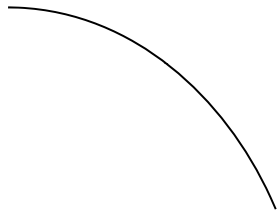
- `new Arc2D.Float(x, y, width, height, startingAngle, arcAngle, type)`

**OR**

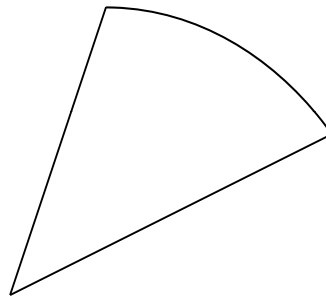
- `new Arc2D.Double(x, y, width, height, startingAngle, arcAngle, type)`

- Type can be specified as `Arc2D.OPEN`, `Arc2D.PIE` or `Arc2D.CHORD`

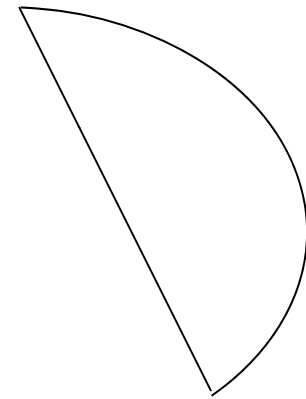
# Arc type



Arc2D.OPEN



Arc2D.PIE



Arc2D.CHORD

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

```
public class ArcDemoP_01 extends JApplet {  
private static int X_TT_HEAD = 100;  
private static int Y_TT_HEAD= 100;  
private static int X_TT_EYE = 100;  
private static int Y_TT_EYE = 80;  
private static int RADIUS_HEAD = 50;  
private static int RADIUS_EYE = 5;  
private static int STARTANGLE= 30;
```

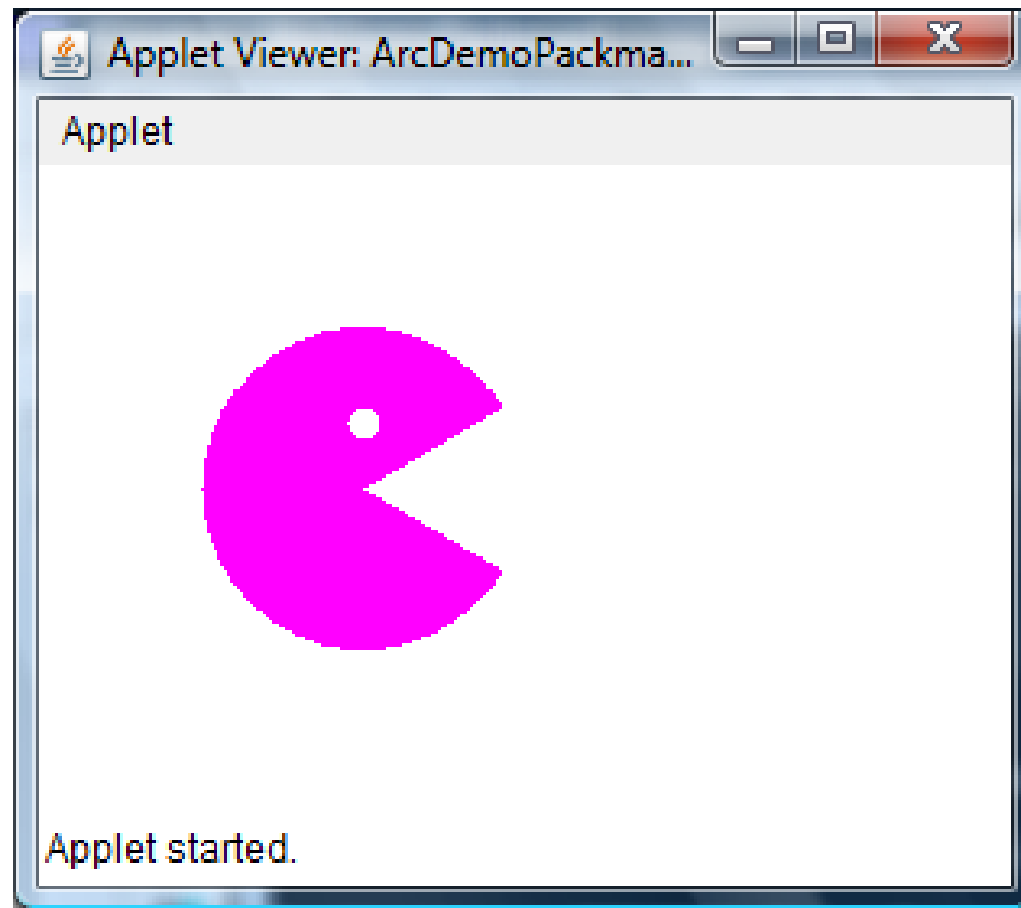
```

public void paint(Graphics g) {
    Graphics2D g2d = (Graphics2D) g;
    Arc2D head = new Arc2D.Double(
        X_TT_HEAD - RADIUS_HEAD,
        Y_TT_HEAD - RADIUS_HEAD,
        2*RADIUS_HEAD, 2*RADIUS_HEAD, STARTANGLE,
        360-2*STARTANGLE, Arc2D.PIE);
    g2d.setPaint(Color.magenta);
    g2d.fill(head);

    Ellipse2D eye = new Ellipse2D.Double(
        X_TT_EYE - RADIUS_EYE, Y_TT_EYE - RADIUS_EYE,
        2*RADIUS_EYE, 2*RADIUS_EYE);
    g2d.setPaint(Color.white);
    g2d.fill(eye); }
}

```

# Output



# Polygon and PolyLine

- **Polygon** : shape which is constructed by several connected line segments to form a closed space
- **Polyline** : shape which is constructed by several connected line segments, however the last vertex is not connected to the first vertex
- **GeneralPath** object is used to represent a polygon or polyline.
- To create object **GeneralPath**
  - `new GeneralPath()`

# Polygon and PolyLine

- Methods of object `GeneralPath`:
  - `moveTo(x, y)` :add vertex (x, y)
  - `lineTo(x, y)`: add vertex (x, y) which is connected to current vertex
  - `closePath()`: close the space created
- Method `closePath()` is called if object `GeneralPath` is used to present a polygon

Array x[] and y[]  
store x and y values  
for polygon

```
public class Polygon extends JApplet {
```

Return to  
Starting position  
(90,90)

```
    void paint(Graphics g) {  
        g.moveTo(90, 85, 140, 185, 200, 200, 185, 140, 85);  
        g.moveTo(90, 85, 30, 75, 60, 120, 105, 150, 95);
```

```
        Graphics2D g2d = (Graphics2D) g;
```

Go to position  
(90,90)

```
        GeneralPath poly = new GeneralPath();
```

```
        poly.moveTo(x[0], y[0]);
```

```
        for (int i=1; i < x.length; i++)
```

```
            poly.lineTo(x[i], y[i]);
```

```
        poly.closePath();
```

```
        g2d.draw(poly);
```

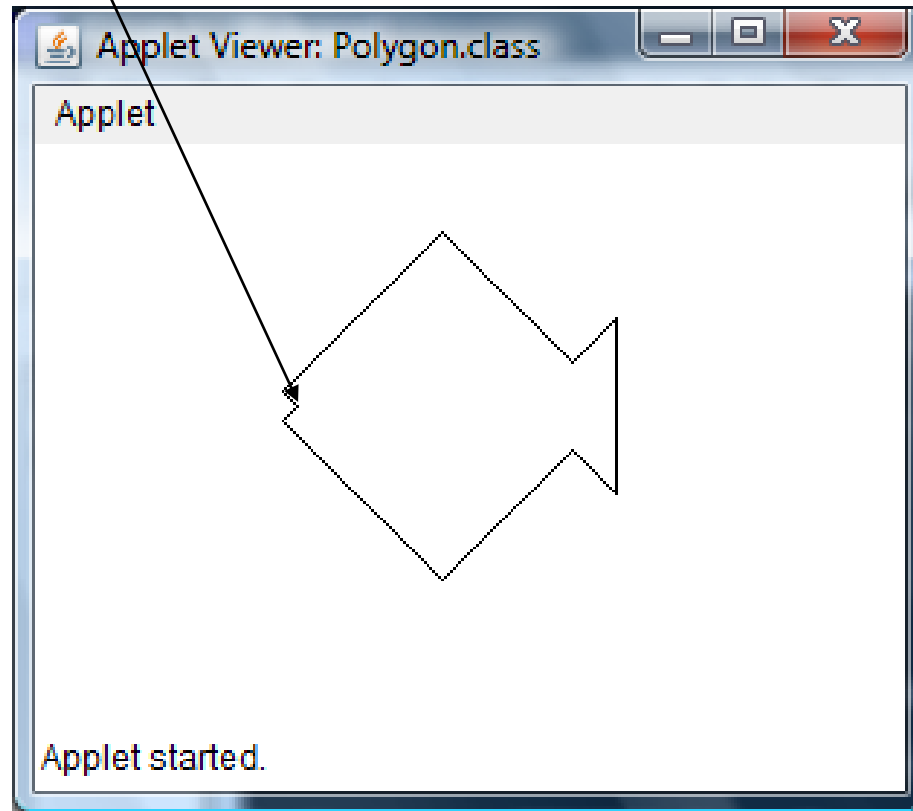
Draw polygon using  
x[] and y[] coordinates

```
    }
```

```
}
```

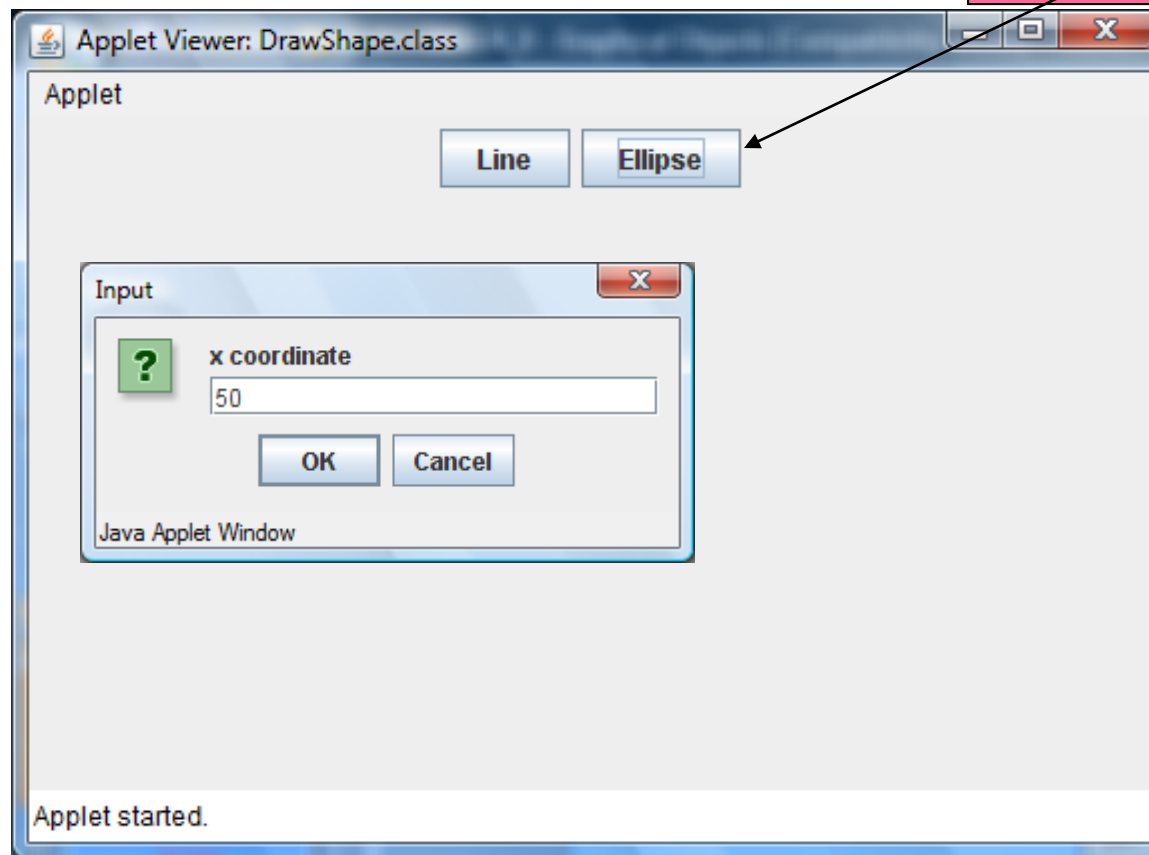


Starting coordinate  
(90,90)

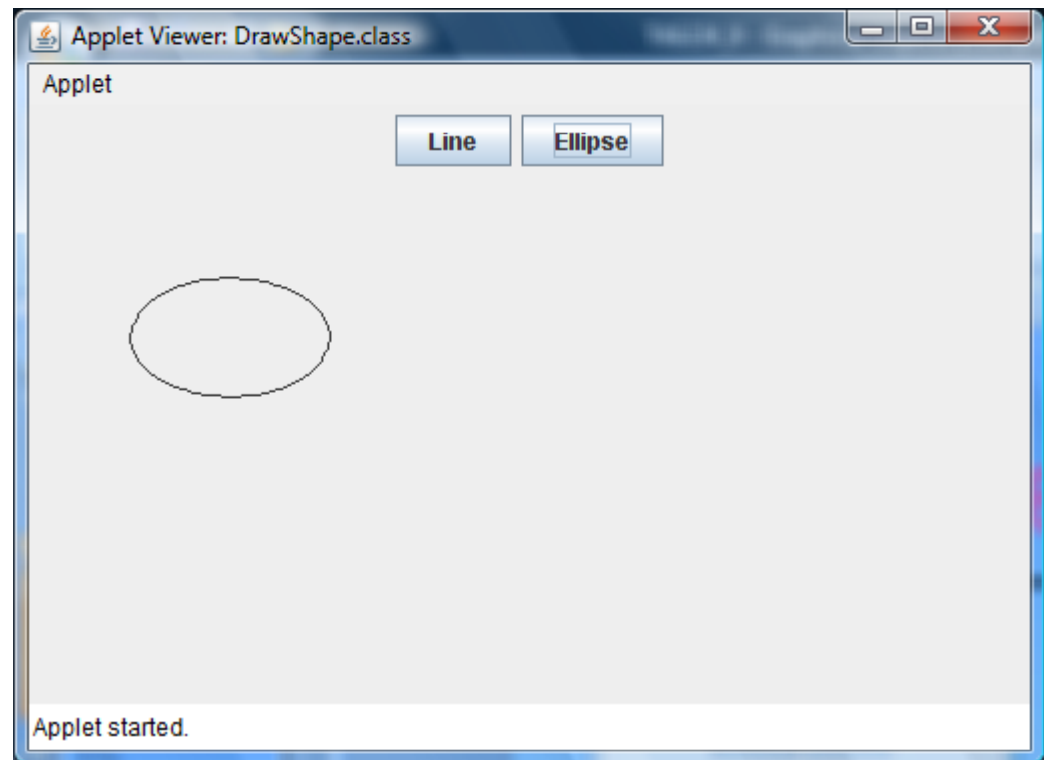
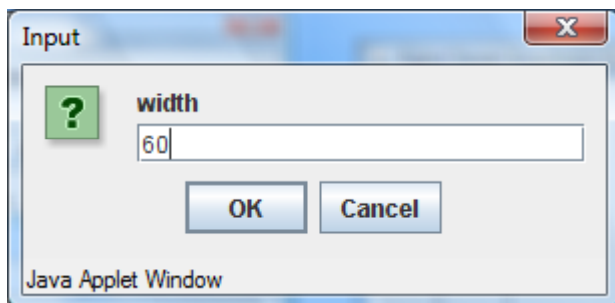
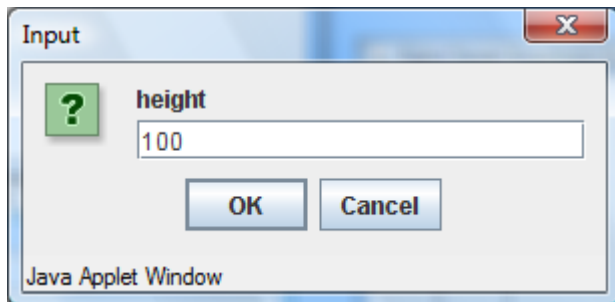
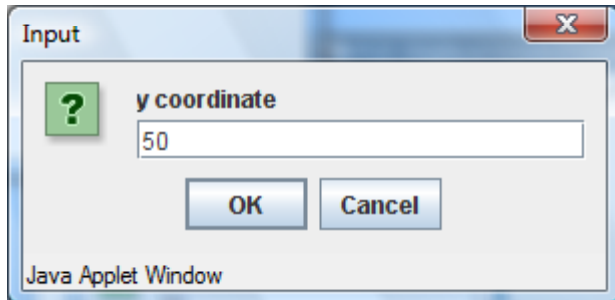


# Example

Click Ellipse button



# Example



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

```
public class DrawShape extends JApplet implements ActionListener{
    private int x,y;
    private String s="";
    private int height,width,radius;
    GraphicDisplay graphic = new GraphicDisplay();
```

```
public void init(){
    Container pane = getContentPane();
    pane.setLayout(new BorderLayout());
    JPanel panel = new JPanel();

    JButton line = new JButton("Line");
    line.addActionListener(this);
    panel.add(line);
    JButton circle = new JButton("Ellipse");
    circle.addActionListener(this);
    panel.add(circle);
    pane.add(panel,"North");
    pane.add(graphic, "Center");
}
```

Preparing container

Register components

GraphicDisplay class

```

public void actionPerformed(ActionEvent e){
    s = e.getActionCommand();
    if(s.equals("Line")) {
        String inputDatax =
        JOptionPane.showInputDialog(null,"x coordinate");
        x = Integer.parseInt(inputDatax);

        String inputDatay = JOptionPane.showInputDialog(null,"y

        y = Integer.parseInt(inputDatay);

        String inputx1 =
        .showInputDialog(null,"height ");
        int x1 = Integer.parseInt(inputx1);

        String inputy1 =
        JOptionPane.showInputDialog(null,"width ");
        int y1 = Integer.parseInt(inputy1);
        graphic.setShape(x, y, x1, y1, s);
    }
}

```

If button "Line" is clicked, get the starting and end point for line

If button "Ellipse" is clicked, get (x,y,height and width) for ellipse

```
else ←
{
    String inputDataX = JOptionPane.showInputDialog("x coordinate");
    x = Integer.parseInt(inputDataX);

    String inputDataY = JOptionPane.showInputDialog("y coordinate");
    y = Integer.parseInt(inputDataY);

    String inputHeight = JOptionPane.showInputDialog("height ");
    height = Integer.parseInt(inputHeight);

    String inputWidth = JOptionPane.showInputDialog("width ");
    width = Integer.parseInt(inputWidth);
    graphic.setShape(x, y, height, width, s);
}
}
}
```

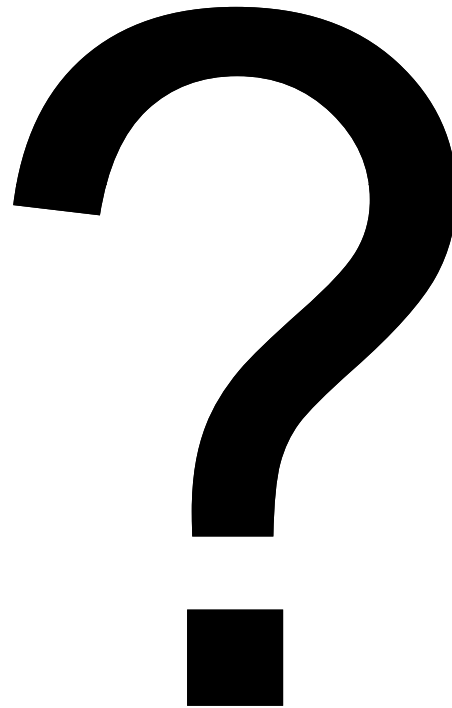
```
class GraphicDisplay extends JPanel {  
    private int height,width,radius;  
    private int x,y;  
    private String s = "";
```

This class will display the  
Shape chosen through  
button

```
    public void setShape(int xx, int yy, int l, int t, String shape) {  
        x = xx;  
        y = yy;  
        width = l;  
        height = t;  
        s = shape;  
        repaint();  
    }  
    public void paint(Graphics g){  
        super.paint(g);  
        Graphics2D g2d = (Graphics2D) g;  
        Ellipse2D.Double elips = new Ellipse2D.Double();  
        if (s.equals("Line")) {  
            g2d.draw(new Line2D.Double(x, y, x+width, y+height));  
        }  
        else{  
            g2d.draw(new Ellipse2D.Double(x, y, width, height));  
        }  
    }  
}
```

return

# Questions





**THANK YOU**

**Arash Habibi Lashkari**

PHD. Candidate of UTM

Kuala Lumpur, Malaysia

Feb, 2010

**THE END**