## Week 2, Lecture 2

## Iteration: for-loops




## This Week

***Java Genesis*:**

**–Ch3: Basic programming constructs**

**Lab Assessment 2 (deadline Week 5)**



## A Scenario

1. Open window with circle at its centre.
2. Wait one second.
3. Select at random a new position in the window.
4. Move the circle to the new position.
5. Repeat fifty times steps 2 to 4.

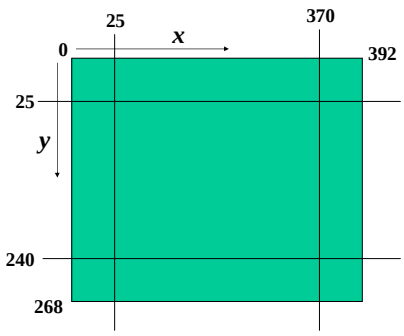


## CircleFigure class

**some messages:**

- **create( )**
- **moveTo(*x*, *y*)**
- **getXCentre( )**
- **getYCentre( )**








## Math class

**some messages:**

- **random( )**
- **sqrt(*x*)**
- **max(*x*, *y*)**
- **round(*x*)**

**(For x-coordinate of circle's centre)**

**Random integer in range 25 to 370**

**370-25+1 = 346**

**0 < Math.random() < 1**

**0 < 346*Math.random() < 346**

**(int)(346*Math.random()) int in range 0 to 345**

**(int)(346*Math.random())+25 in range 25 to 370**



**(For y-coordinate of circle's centre)**

**Random integer in range 25 to 240**

**240-25+1 = 216**

**0 < Math.random() < 1**

**0 < 216*Math.random() < 216**

**(int)(216*Math.random()) int in range 0 to 215**

**(int)(216*Math.random())+25 in range 25 to 240**



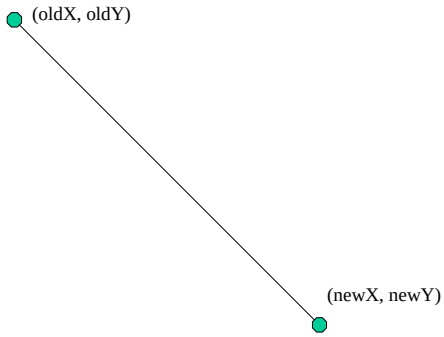
```
import genesis.*;

public class JumpingCircle {

   public static void main (String [ ] args) {
     CircleFigure.create();
     for (int i=1; i<=50; i++) {
        Delay.milliseconds(1000);
        int newX = (int)(346*Math.random())+25;
        int newY = (int)(216*Math.random())+25;
        CircleFigure.moveTo(newX, newY);
     }
   }
}
```



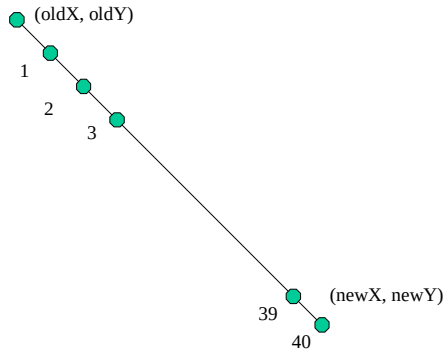
## New Scenario

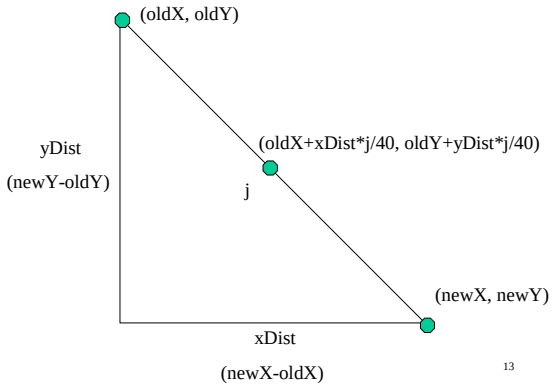
1. Open window with circle at its centre.
2. Select at random a new position in the window.
3. Move the circle to the new position by taking 40 equal steps waiting 25 milliseconds between steps.
4. Repeat fifty times steps 2 and 3.












```
import genesis.*;
public class LostCircle {

   public static void main (String [ ] args) {
       CircleFigure.create( );
       int newX,newY,oldX,oldY,xDist,yDist,currentX,currentY;
       for (int i=1; i<=50; i++) {
             newX = (int)(346*Math.random( ))+25;
             newY = (int)(216*Math.random( ))+25;
             oldX = CircleFigure.getXCentre( );
             oldY = CircleFigure.getYCentre( );
             xDist = newX - oldX;
             yDist = newY - oldY;
             for (int j=1; j<=40; j++) {
                Delay.milliseconds(25);
                currentX = (int)(oldX+xDist*j/40.0);
                currentY = (int)(oldY+yDist*j/40.0);
                CircleFigure.moveTo(currentX, currentY);
             }
       }
   }
}
```