

Packages and plotting with pt plot.

Java Package	Classes for
java.lang	Basic elements of Java language
java.util	Utilities; random number generators, date, time, <i>etc.</i>
java.awt	Abstract Windowing Toolkit; creates graphical interfaces
java.applet	Creates applets and interact with browsers
java.beans	Creates reusable software components
java.io	Data input and output

```
1 package <mypackage_name>;
2 public class <myclass1_name>
3     {
4     <normal class structure , multiple methods OK>
5     }
6     public class <myclass2_name>
7     {
8     <normal class structure , multiple methods OK>
9     }
```

Listing 11.1 EasyPtPlot.java

```

1 // EasyPtPlot.java: A simple plot application, plots f(x)
2 import tolemy.plot.*; //Plotting package
3 public class EasyPtPlot
4 { public static final double Xmin = -5., Xmax = 5.;
   //y range automatic
5 public static final int Npoint = 500;
6 public static void main(String[] args)
7 { Plot plotObj = new Plot(); // Create ptPlot object
8 plotObj.setTitle("f(x) vs x");
9 plotObj.setXLabel("x");
10 plotObj.setYLabel("f(x)");
11 // Format: addPoint(int dataSet, double x, double y, boolean
   connect)
12 double xStep = (Xmax - Xmin) / Npoint;
13 for (double x = Xmin; x <= Xmax; x += xStep)
14 { double y = Math.cos(x);
15 plotObj.addPoint(0, x, y, true); } //Add point
16 PlotApplication app = new PlotApplication(plotObj); //Display
17 }}

```

Calling PtPlot from Your Program

Plot myPlot = new Plot();	Name and create plot object <code>myPlot</code>
PlotApplication app = new PlotApplication(myPlot);	Display finished plot
myPlot.setTitle("f(x) vs x");	Add title to plot
myPlot.setXLabel("x");	Label x axis
myPlot.setYLabel("f(x)");	Label y axis
myPlot.addPoint(0, x, y, true);	Add (x, y) to set 0, connect points
myPlot.addPoint(1, x, y, false);	Add (x, y) to set 1, no connect points
myPlot.addLegend(0, "Set 0");	Label data set 0 in legend
myPlot.addPointWithErrorBars(0, x, y, yLo, yHi, true);	Plot, $(x, y - YLo)$, $(x, y + yHi)$ + error bars
myPlot.clear(0);	Remove all points from data set 0
myPlot.clear(false);	Remove data from all sets
myPlot.clear(true);	Remove all points, default options
myPlot.setSize(500, 400);	Set plot size in pixels (optional)
myPlot.setXRange(-10., 10.);	Set x range (default fit to data)
myPlot.setYRange(-8., 8.);	Set a y range (default fit to data).
myPlot.setXLog(true);	Use log scale for x axis
myPlot.setYLog(true);	Use log scale for y axis
myPlot.setGrid(false);	Turn off the grid
myPlot.setColor(false);	Color in black and white
myPlot.setButtons(true);	Display zoom-to-fit button on plot
myPlot.fillPlot();	Adjust x, y ranges to fit data
myPlot.setImpulses(true, 0);	Lines from points to x axis, set 0
myPlot.setMarksStyle("none", 0);	Draw <code>none</code> , <code>points</code> , <code>dots</code> , <code>various</code> , <code>pixels</code>
myPlot.setBars(true);	Display data as bar charts
String s = myPlot.getTitle();	Extract title (or other properties)

Writing your own program:

1. Pose the problem
2. Design the solution
 1. Identify constants, variables, methods
 2. Write algorithm (in pseudo-code)
3. Design your class [es]
 1. Identify global variables
 2. Identify functions
 3. Code your solution

Problem: calculating the projectile motion

$$V_{0x} = V_0 \cos \theta, \quad V_{0y} = V_0 \sin \theta.$$

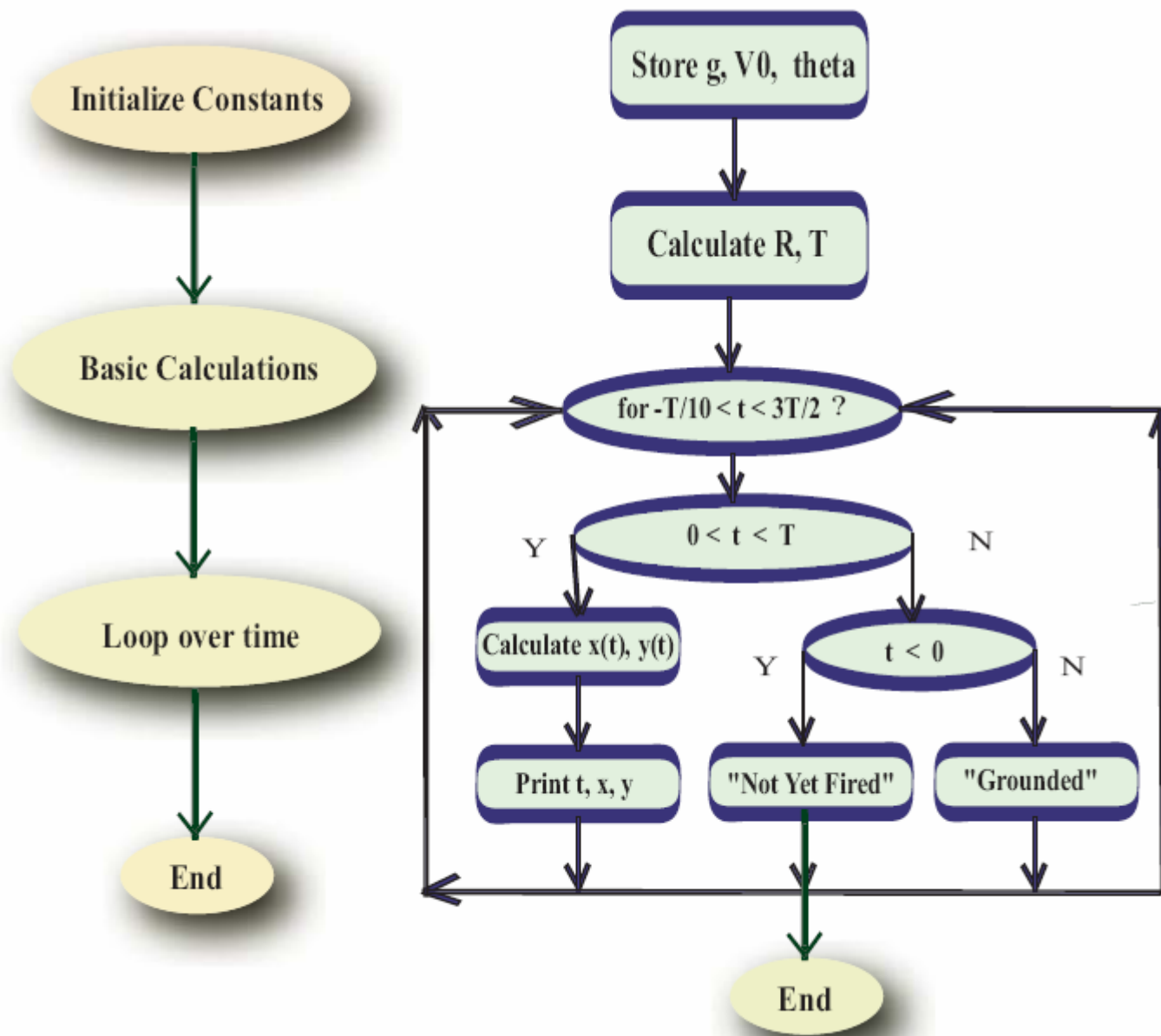
$$v_x(t) = V_{0x}, \quad v_y(t) = V_{0y} - gt$$

$$x(t) = V_{0x}t, \quad y(t) = V_{0y}t - \frac{1}{2}gt^2$$

$$T = \frac{2V_{0y}}{g},$$

$$R = x(T) = V_{0x} \frac{2V_{0y}}{g} = \frac{2V_0^2 \sin \theta \cos \theta}{g}$$

Sample algorithm: projectile motion



Java flow control

Name	Example	Comment
if-then	<code>if ((x < 3) && (y == 12)) z = y*x;</code>	One-way if.
if-then-else	<code>if (x <= 0.) {y = y * y;} else y = 2 * y;</code>	Two-way if. the catchall
if-then-else-if	<code>if (score >= 90) {grade='A';} else if (score >= 80) {grade='B';} else if (score >= 70) {grade= 'C';}</code>	Multi-way if Any else-if's OK Inaccessible if other else true
for	<code>for (count=0; count<100; count=count+1) {<statements>}</code>	Initial; for; increment Multi-line code block
do	<code>do {<statements>} while (<boolean>; <labelname>: x = x*y; ... break <labelname>; if (i==99) continue;</code>	Goes through at least once Control sent to here go to labelname Jump to loop end
do while	<code>while (Math.sin(x) < 100.) {y = y*y; x = x + y;}</code>	Evaluate as long as true; line to execute
switch	<code>switch (month) {case 1: s="Jan"; break; case 12: s = "Dec"; break;}</code>	Case 1, break after code block Can have many cases

Logical operators

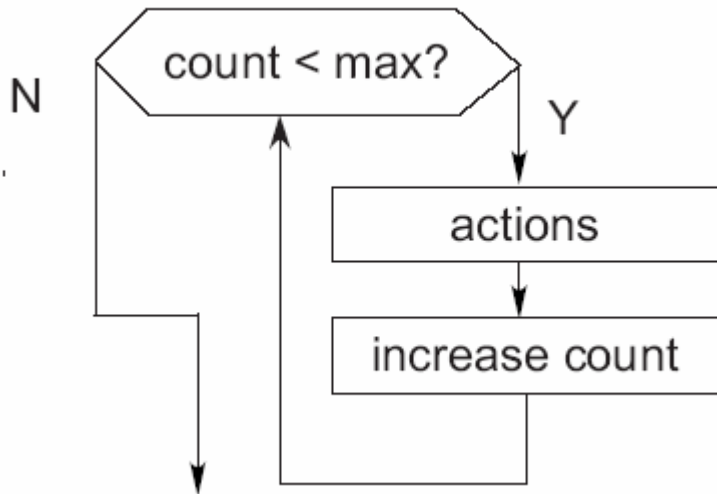
Operator	Example	Return <i>true</i> if
>	x > y	X is greater than y
>=	x >= y	X is greater than or equal to y
<	x < y	X is less than y
<=	x <= y	X is less than or equal to y
==	x == y	X and y are equal
!=	x != y	X and y are not equal

Combining logical operators

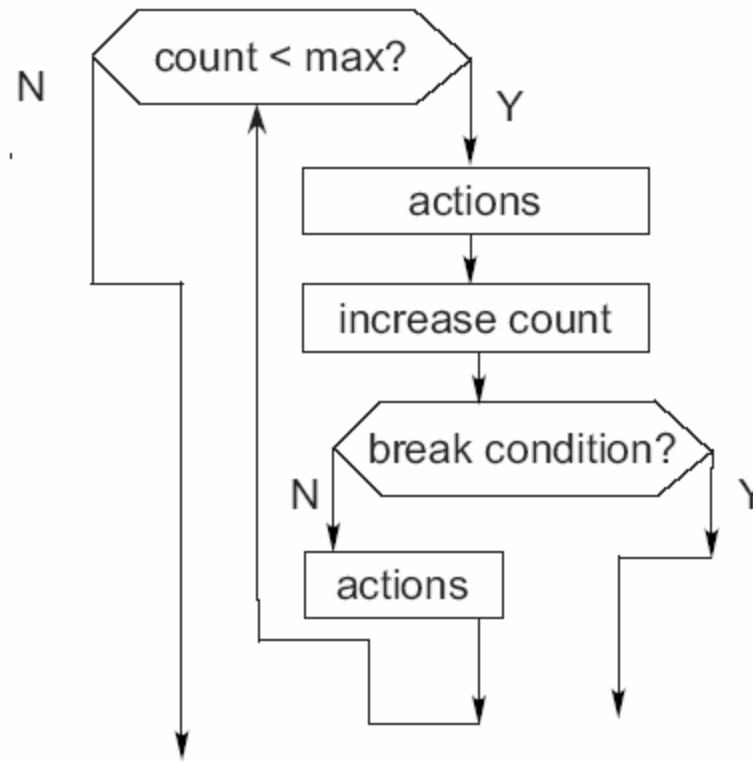
Operator	Example	Name	Return <i>true</i> if
&&	x && y	Logical and	x and y both true, conditionally evaluates y
	x y	Logical or	either x or y true, conditionally evaluates y
!	!x	Not	x is false
&	x & y	And	x and y both true, always evaluates y
	x y	Or	either x or y true, always evaluates y

Pseudocode for Java flow control

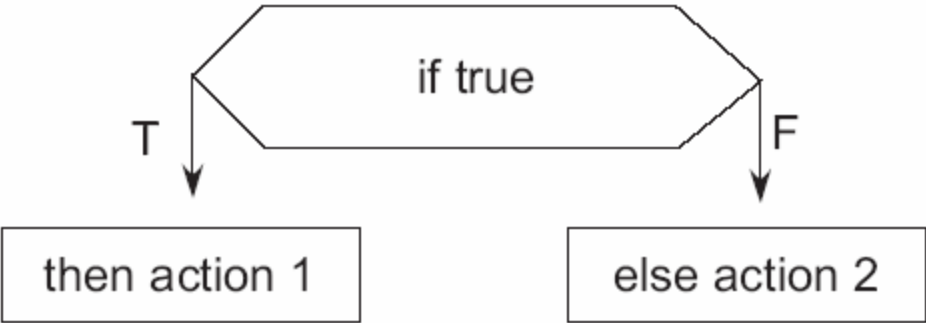
For loop



For loop with break statement



If statement



Do loop

