CS1110, 16 October 2008
Two topics: Turtles; loops

Start reading Sec. 2.3.8 and chapter 7 on loops.
The lectures on the ProgramLive CD can be a big help.

The next time someone rather casually use a number that includes the word "billion", think about it.

- A billion seconds ago was 1959.
- A billion minutes ago Jesus was alive.
- A billion hours ago our ancestors were living in the Stone Age.
- A billion days ago no creature walked the earth on two feet.
- A billion dollars lasts 8 hours and 20 minutes at the rate our government spends it.

1

Function HSVtoRGB

0 <= h < 360 (degrees)
Hi = floor(h/60) % 6
Wikipedia: f = h/60 – floor(h/60)
A4 handout: f = h/60 – Hi

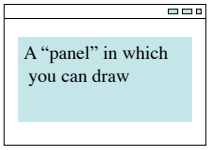
In fact, we and Wikipedia could write Hi as

Hi = floor(h/60) !!!!

0 <= h < 360
=> <arithmetic>
0 <= h/60 < 6
=> <arithmetic>
floor(h/60) is in 0..5
=> <arithmetic>
floor(h/60) = floor(h/60) % 6
=> <definition of Hi>
Wikipedia formula for f = A4 handout for f

2

Graphical User Interfaces (GUIs): graphics.



A "panel" in which you can draw

A JFrame, with a "panel" on which you can draw

You don't have to learn all this unless you want to. We will be telling you more and more about GUIs as the course progresses.

```
jframe= new JFrame("Turtle window");
jpanel= new JPanel();
jpanel.setPreferredSize(new Dimension(width, height));
jpanel.setBackground(Color.white);
jframe.getContentPane().add(panel, BorderLayout.CENTER);
jframe.pack();
jframe.setVisible(true);
graphics= jpanel.getGraphics();
```

3

(0,0) (0,1) (0, 2) …
(1,0) (1,1), (1,2) …
(2,0) (2,1), (2,2) …
…

Commands to draw

The panel: each pair (i,j) is a "pixel" or picture element.

d.graphics contains an object of class Graphics. It contains the methods to draw on the panel

```
// Draw line from (10, 10) to (50, 40).
d.graphics.drawLine(10,10,50, 40);

// Draw rectangle: top-left point (2, 5), width 40, height 60
d.graphics.drawRect(2, 5, 40, 60);

// Fill rectangle: top-left point (50, 70), width 40, height 60
d.graphics.fillRect(50, 70, 40, 60);
```

4

```
// Draw string s at (40, 30)
d.graphics.drawString(s, 40, 30);

// set the pen color to red
d.graphics.setColor(Color.red);

// Store the current color in c
Color c= d.graphics.getColor();

// Draw oval: top-left point (2, 5), width 40, height 60
d.graphics.drawRect(2, 5, 40, 60);

// Fill an oval: top-left point (50, 70), width 40, height 60
d.graphics.fillRect(50, 70, 40, 60);
```

(0,0) (0,1) (0, 2) …
(1,0) (1,1), (1,2) …
(2,0) (2,1), (2,2) …
…

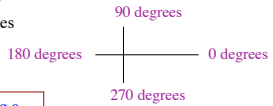
For more on graphics, see class Graphics in the Java API and page 1-5 in the CD ProgramLive. For more on GUIs, read chapter 17 --corresponding part of the CD is much easier!

5

Assignment A5: drawing with a Turtle

We have written a class Turtle, an instance of which maintains:

- point (x, y): where the "Turtle" is
- angle: the direction the Turtle faces
- a pen color
- whether pen is up or down



Class Turtle has methods for moving a Turtle around, drawing as it goes.

Draw equilateral triangle with side lengths 30; turtle ends up at starting point and facing the same direction:

```
forward(30);  addAngle(120);
forward(30);  addAngle(120);
forward(30);  addAngle(120);
```

In A5, write methods to draw shapes, draw spirals, make balls that move and bounce off the sides of the window, and draw things using recursive procedures.

6

The for loop, for processing a range of integers

```
x= 0;
// add the squares of ints
// in range 2..200 to x
x= x + 2*2;
x= x + 3*3;
…
x= x + 200;
```

for each number i in the range 2..200, add i*i to x.

loop counter: i
initialization: **int** i= 2;
loop condition: i <= 200;
increment: i= i + 1
repetend or body: { x= x + i*i; }

The for-loop:

```
for (int i= 2; i <= 200; i= i +1) {
   x= x + i*i;
}
```

repetend: the thing to be repeated. The block:

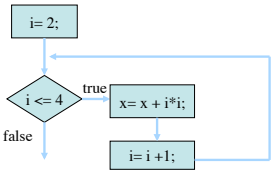
{ x= x + i*i; }

Execution of the for-loop

The for-loop:

```
for (int i= 2; i <= 4; i= i +1) {
   x= x + i*i;
}
```

loop counter: i
initialization: **int** i= 2;
loop condition: i <= 4;
increment: i= i + 1
repetend or body: { x= x + i; }



To execute the for-loop.
1. Execute initialization.
2. If loop condition false, terminate execution.
3. Execute repetend.
4. Execute increment, repeat from step 2.

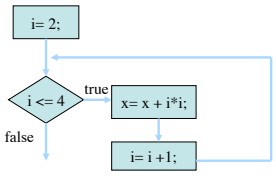
Called a "flow chart"

Execution of the for-loop

At the end:

```
x= 1;
for (int i= 1; i <= 4; i= i +1) {
   x= 2*x;
}
```

A. x = 16 i = 5
B. x = 16 i = 4
C. x = 8 i = 5
D. x = 32 i = 32
E. none of the above



To execute the for-loop.
1. Execute initialization.
2. If loop condition false, terminate execution.
3. Execute repetend.
4. Execute increment, repeat from step 2.

Called a "flow chart"

Note on ranges.

2..5 contains 2, 3, 4, 5. It contains 5+1 – 2 = 4 values
2..4 contains 2, 3, 4. It contains 4+1 – 2 = 4 values
2..3 contains 2, 3. It contains 3+1 – 2 = 2 values
2..2 contains 2. It contains 2+1 – 2 = 1 values
2..1 contains . It contains 1+1 – 2 = 0 values

The number of values in **m..n** is **n+1 – m**.

In the notation m..n, we require always, without saying it, that

m <= n + 1 .

If m = n + 1, the range has 0 values.

Pattern for processing range of integers:

range a..b-1

```
for (int k= a; k < b; k= k + 1) {
      Process integer k;
}
```

```
// Print the integers in 10..n-1
// inv: All ints in 10..k-1 been printed
for (int k= 10; k < n; k= k +1) {
   System.out.println(k);
}
// All ints in 10..n-1 been printed
```

range c..d

```
for (int i= c; i <= d; i= i + 1) {
      Process integer i;
}
```

```
// Print the integers in 1..10
// inv: All ints in 10..i-1 printed
for (int i= 1; i <= 10; i= i +1) {
   System.out.println(i);
}
// All ints in 10..i-1 printed
```


The pattern for processing range of integers:

range a..b-1

```
for (int i= a; i < b; i= i + 1) {
      Process integer i;
}
```

```
// Print indices of all 'e's in String s
// inv: Indices of 'e's in s[0..s.i-1]
for (int i= 0; i < s.length(); i= i +1) {
   if (s.charAt(i) == 'e')
      System.out.println(i);
}
// Indices of 'e's in s[0..s.length()-1]
// printed
```

range c..d

```
for (int i= c; i <= d; i= i + 1) {
      Process integer i;
}
```

```
// Store in double var. v the sum
//      1/1   + 1/2 + …+ 1/n
v= 0;
// inv: 1/1   + 1/2 + …+ 1/(i-1)
for (int i= 1; i <= n; i= i +1) {
      v= v + 1.0 / i;
}
// v= 1/1   + 1/2 + …+ 1/n
```

Loops are often not easy to develop or understand.

Our goal: Provide you with a methodolgy for the development of loops that process a range of integers.

1. Separate your concerns —focus on one thing at a time.
2. Make small steps toward completing the loop.
3. Don't introduce a new variable without a good reason.
4. Keep program simple.

Development of a loop to process a range a..b

Follow this methodology for ease in writing loops!!!

```
// Store in m the sum of even
//  numbers in 10..46
 m= 0;
// m = sum of even ints in 10..(k-1)
for (int k= 10 ; k <= 46; k= k+1  ) {
    // Process k

    if (k % 2 == 0) {
       m= m + k;
    }
}
// m = sum of even ints in 10..46
```

```
for (int i= a;  i <= b;  i= i + 1) {
        Process integer i;
}
```

Step 1. Recognize that a range of integers has to be processed.

Step 2. Write a postcondition, based on the spec, which says what is true at the end.

Step 3. Write the skeleton of the loop.

Step 4. Fill in the loop control.

Step 5. Write down, before the loop, what the variables mean and initialize other variables.

Step 6. Write the method body (to process k).

Development of a loop to process a range a..b-1

```
// Set c to the number of chars in String s that are digits 0..9
for (int i=   ;    ;       ) {
        Process integer i;
}
```

What is the range of integers to process?

A. 1 .. s.length()

B. 1 .. s.length() – 1

C. 0 .. s.length()

D. 0 .. s.length() – 1

E. I don't know.

Development of a loop to process a range a..b-1

```
// Set c to number of chars in String s that are digits '0'..'9'
for (int i=   ;    ;       ) {
        Process integer i;
}
```

What is the the postcondition?

A. c = no. of chars in s that are in '0'..'9'

B. c = no. of chars in s[0..s.length()-1] that are in '0'..'9'

C. c = no. of chars in s[0..s.length()] that are in '0'..'9'

D. A or B

E. I don't know

Development of a loop to process a range a..b-1

```
// Set c to number of chars in String s that are digits '0'..'9'
for (int i=   ;    ;       ) {
        Process integer i;
}
// c = no. of chars in s[0..s.length()-1] that are in '0'..'9'
```

Write the initialization, loop condition, and increment

A. `for (int i= 1; i <= 9 ; i= i + 1 )`
B. `for (int i= 1; i <= s.length() ; i= i + 1 )`
C. `for (int i= 1; i < s.length() ; i= i + 1 )`
D. `for (int i= 0; i < s.length() ; i= i + 1 )`
E. `for (int i= 0; i <= s.length() - 1 ; i= i + 1 )`

Development of a loop to process a range a..b-1

```
// Set c to number of chars in String s that are digits '0'..'9'
// What should be true here about c and i?
for (int i=  0;   i < s.length();   i= i + 1      ) {
        Process integer i;
}
// c = no. of chars in s[0..s.length()-1] that are in '0'..'9'
```

A. // c= no. of chars in s[0..i-1] that are in '0' .. '9'
B. // c= no. of chars in s[0..i] that are in '0' .. '9'
C. // c= no. of chars in s[1..i] that are in '0' .. '9'
D. I don't know.

Development of a loop to process a range a..b-1

```
// Set c to number of chars is String s that are digits '0'..'9'
 // inv: c = no. of chars of s[0..i-1] that are in '0'..'9'
for (int i=  0;   i < s.length();   i= i + 1       ) {
        Process integer i;
}
// c = no. of chars of s[0..s.length()-1] that are in '0'..'9'
```

How should c be initialized c?

A. c= 1; C. c= 5;

B. c= 0; D. c= -1;

Try these problems. Develop them using the methodology given on slide 9. Then type them into DrJava and test them!

1. Set c to the number of chars is String s that are digits (in 0..9).
2. Store in res a copy of String s but with no blanks.
3. Store in res a copy of String s but with adjacent duplicates removed.
4. Set boolean v to the value of "no integer in 2..n–1 divides x".
5. Set boolean v to the value of "every element in Vector v is an object of class JFrame".
6. Add up the squares of the odd integers in the range m..n.