

CS1110, 20 October 2009 Intro to loops and for-loops

Reading: Sec. 2.3.8 and chapter 7 on loops.

The lectures on the ProgramLive CD can be a big help.

Please have clickers out.

Assignment A5 out today: graphics, loops, recursion. Due Thursday October 29.

Public service announcement: Whether you are a Freshman or Senior, Career Services can help you find exciting opportunities for next summer. Your college career office is the primary resource for jobs, internships, graduate school information, and assistance with career exploration. You are encouraged to schedule an appointment with an individual counselor or advisor to create a personalized career plan and maximize your effectiveness on the job search.

Those of you in Arts & Sciences should know that the career staff in Goldwin Smith specialize in assisting liberal arts students. Additionally, their new web site includes information on connecting with alumni, identifying summer funding options, and field specific employment & graduate school resources. <http://as.cornell.edu/careers>

1

From recursion to loops: doing things repeatedly

We write programs to make computers do things.

We often want to make them do things *multiple times*.

1. Perform n trials or get n samples.
 - A5: draw a triangle six times to make a hexagon
 - Run a protein-folding simulation for 10^6 time steps
2. Process each item in a given String, Vector, or other "list"
 - Compute aggregate statistics for a dataset, such as the mean, median, standard deviation, etc.
 - Send everyone in a certain (Facebook) group an individual appointment time
3. Do something an unknown number of times
 - ALVINN, the van that learned to drive itself, continuously watched human driving behavior and adjusted its model accordingly

2

From recursion to loops: doing things repeatedly

We've talked about *recursion*.

Alternatives:

while-loops

for-loops (special syntax for common special cases)

```
<set things up>;
while (stuff still to do) {
    <process current item>;
    <prepare for next item>;
}
```

3

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*200;
```

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

loop counter: i
initialization: $\text{int } i = 2$;
loop condition: $i \leq 200$;
increment: $i = i + 1$
repetend or body: $\{ x = x + i^2; \}$

The for-loop:
for ($\text{int } i = 2; i \leq 200; i = i + 1$) {
 $x = x + i^2$;
}

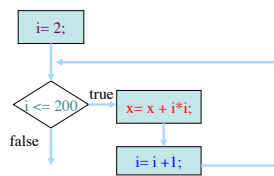
4

Execution of the for-loop

The for-loop:

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

loop counter: i
initialization: $\text{int } i = 2$;
loop condition: $i \leq 200$;
increment: $i = i + 1$
repetend or body: $\{ x = x + i; \}$



Called a "flow chart"

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.

5

Note on ranges

(later, will make reasoning about loops easier)

2..5 contains 2, 3, 4, 5. It contains $5 + 1 - 2 = 4$ values

2..4 contains 2, 3, 4. It contains $4 + 1 - 2 = 3$ 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

$m \leq n + 1$ (so, "5..4" is OK but not "5..3")

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

6

Application: URL analysis for search engines

Problem: how does a search engine (e.g., Google) decide which webpages are the most important to present?

(Small) part of the answer: use URL cues

- “Deep” URLs are usually less important, e.g.,
www.fake.com/this/that/other/minor/tiny/detail.htm

This requires counting the number of slashes in a URL (given as a String).

7

The pattern for processing range of integers: range a..b-1 range c..d

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

```
// store in count # of '/'s in String s  
// inv: count is # of '/'s in s[0..s.i-1]  
count=0;  
for (int i= 0; i < s.length(); i= i + 1) {  
    if (s.charAt(i) == '/')  
        count= count+1;  
}  
// count is # of '/'s in s[0..s.length()-1]
```

```
// Store in double var. v the sum  
// 1/1 + 1/2 + ...+ 1/n  
v= 0; // call this 1/0 for today  
// inv: v is 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
```

8

Application: Some Personalized Email (SPEM)

Problem: how can we get people to read our mass email messages?

One answer: make it personal.

- Only one recipient
- Customized message (“Hi Lisa, great seeing you at the talk yesterday. Don’t forget the meeting tomorrow”; “Hail Batman. This course needs a better class of Criminal. Don’t forget the meeting tomorrow”)
- We don’t want to add duplicate recipients to the list (people notice and hate getting redundant emails).

This requires storing individualized information, iterating over the items we stored, and figuring out msg/mail output.

9

Some Personalized Email (SPEM): design decisions

How shall we represent a group of recipients (e.g., TAs vs. students)?

- (A) String containing all email addresses catenated together
- (B) Vector of items of new class MailRecip
- (C) New class that contains items of new class MailRecip

10

Loops are often not easy to develop or understand.

Our goal: Provide you with a methodology 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.

11

Try these problems, first by hand, and then checking with DrJava.

1. Set c to the number of chars in 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.

12