Readings and References

Looping

CSE 142, Summer 2002 Computer Programming 1

http://www.cs.washington.edu/education/courses/142/02su/

cse142-08-Looping © 2002 University of Washington

• Reading

- » Chapter 12, An Introduction to Programming and Object Oriented Design using Java, by Niño and Hosch
- » Chapter 15, *Introduction to Programming in Java*, Dugan
- Other References

10-July-2002 cse14

cse142-08-Looping © 2002 University of Washington

2

What is a loop?

- Loop some definitions from dictionary.com
 - » Something having a shape, order, or path of motion that is circular or curved over on itself.
 - » A segment of film or magnetic tape whose ends are joined, making a strip that can be continuously replayed.
 - » Computer Science. A sequence of instructions that repeats either a specified number of times or until a particular condition is met.

Why do we want loops in our code?

- Do something for a given number of times or for every object in a collection of objects
 - » for every animal in the list, provide it with a meal
 - » for every shape in the blob, move the shape
 - » find the classroom with the most seats
 - » calculate the average meal size for all the pets
 - » make a list of all movies that Kevin Bacon has appeared in with Harrison Ford
- Termination of some loops is based on a count

10-July-2002

3

Why do we want loops in our code?

- Keep doing something until we arrive at a termination condition
 - » read until the end of an input command file
 - » search the disk until we find a requested file
 - » read packets from the network until all information for a web page has been read in
 - » remove items from a request queue and process them until the queue is empty
- Termination of some loops is based on a condition

,	cse142-08-L	ooping © 2002 University of Washingt.	on
	fo	or example	
• a countin	ng loop i	mplemented with	th for
can declare variabl	e here	check for termination i runs from 0 to 19	update loop control shorthand for i=i+1
can declare variabl or use existing varia	e here able	check for termination i runs from 0 to 19	update loop control shorthand for i=i+1
can declare variabl or use existing vari	e here able for (in	<pre>check for termination i runs from 0 to 19 t i=0; i<20; i++)</pre>	update loop control shorthand for i=i+1
can declare variabl or use existing varia	e here able for (in tes	<pre>check for termination i runs from 0 to 19 t i=0; i<20; i++) tB.grow();</pre>	update loop control shorthand for i=i+1
can declare variabl or use existing varia	e here able for (in tes }	<pre>check for termination i runs from 0 to 19 t i=0; i<20; i++) tB.grow();</pre>	update loop contro shorthand for i=i+
can declare variabl or use existing varia	e here able for (in tes }	<pre>check for termination i runs from 0 to 19 t i=0; i<20; i++) tB.grow();</pre>	update loop control shorthand for i=i+1

cse142-08-Looping © 2002 University of Washington

10-July-2002

The for loop

- A counting loop is usually implemented with for
 - » The **for** statement is defined in section 14.13 of the Java Language Specification



limited life of a loop control variable

- The scope of a local variable declared in the ForInit part of a for statement includes all of the following:
 - » Its own initializer
 - » Any further declarators to the right in the ForInit part of the for statement
 - » The Expression and ForUpdate parts of the for statement
 - » The contained Statement

7

some shortcuts

- i++
 - » theAnimal = pets.get(i++);
 - » get the value of i for use in the call to get(int), then increment i and store the incremented value
 - » This is known as post-increment
- ++i
 - » theAnimal = pets.get(++i);
 - » get the value of i, increment it, set a copy aside for the call to get(int) and store incremented value in i
 - » This is known as pre-increment

```
10-July-2002
```

cse142-08-Looping © 2002 University of Washington

compound assignment operators

- can shorten statements like this
 - » from this: a = a + b; » to this: a += b;

area = area * factor \Leftrightarrow area *= factor;

```
cse142-08-Looping © 2002 University of Washington
```

10

Why do we want loops in our code?

- Keep doing something until we arrive at a termination condition
 - » read until the end of an input command file
 - » search the disk until we find a requested file
 - » read packets from the network until all information for a web page has been read in
 - » remove items from a request queue and process them until the queue is empty
- Termination of some loops is based on a condition

The while loop

- condition loop is usually implemented with while
 - » The **while** statement is defined in section 14.11 of the Java Language Specification



Note: reaching a limit by counting is satisfying a condition. for loops can be rewritten as while loops, and vice versa

10-July-2002

11

9

while example body of loop may not execute at all • a condition loop implemented with while • Notice that depending on the values of the control variables, it is quite possible that the body of the loop will not execute at all in both for and while check for termination update loop control indeterminate any variable can be part operation of the loop of the controlling condition causes changes that check for termination will eventually cause toDate is already greater than goal, goal = 75;loop to terminate and so the entire loop is skipped boolean atEndOfFile = false; periods = 0;while (!atEndOfFile) { toDate = 100;read another line and set atEndOfFile if appropriate while (toDate < goal) {</pre> process the new line if needed toDate += toDate*rate; } periods++; } Looper.java 10-July-2002 cse142-08-Looping © 2002 University of Washington 13 10-July-2002 cse142-08-Looping © 2002 University of Washington 14