Computer Science 121 - Lecture 8

Announcements

Ch 4 homework (4.11) due Friday

Ch 5 embedded problems due next Wednesday

Programming Assignment 3 due Tuesday

First midterm Tuesday Oct 8: Ch 1-5

Remember to use the learning resource center, see details, schedule at course website, and also at umass.edu/lrc

CS major-applying: cs.umass.edu/admissions/ugrad-admission

BY OCTOBER 1!
Conditional, Looping Statements in Java

Conditional and looping statements are flow of control constructions.

At a primitive level, Java programs are made up of statements, and it often makes sense to have

1) statements repeat in a systematic way; and
2) statements execute conditionally.
Java.util - includes Scanner Random, etc.

Infant class
Car class, etc.

myCar.unusedCap();

Looping; Conditional statements;

Packages

Classes

Methods

Statements
Conditionals first - Consider:

```java
if (n % 2 == 0) System.out.println(n + " is even");
```

Lots going on here: statement says: “if the remainder after dividing n by 2 is equal to (==) 0, then report that n is an even number

```java
if (n % 2 != 0) System.out.println(n + " is odd");
else System.out.println("n is even");
```

An important point: (n % 2 == 0) is a boolean expression (returns a boolean value) -- a boolean must go into the test slot of an if stmt! Nothing else will do!!
The for loop - a control line, a body

```java
for(int j = 1; j <= 5; j++){
    System.out.println(j + " " + j%3);
}
```

Ans:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

5%3 = 2
for(int j = 3; j < 12; j = j + 4){
 System.out.println(j);}

3-7-11

for(int j = 30; j > 20; j = j - 3){
 System.out.println(j);}

30 – 27 – 24 – 21
General form:

```
for(initialize; test; increment) {
    bunch of statements to be executed
}
```
String s = “blah-blah”;
for(int j = 0; j < s.length(); j++){
    System.out.print(j + “ “);
    System.out.println(s.charAt(j));
}

Here we say that the for loop “walks” the string (s)
0  b
1  l
2  a
3  h
4  -
A fragment that checks if s contains a cap A:

String s = "..."; // so - assume s is any string (elephant, snake, zebra,..)

boolean answer = false;
for(int k = 0; k < s.length(); k++){
    if (s.charAt(k) == 'A') answer = true;
}
System.out.println(answer);
Boolean connectives

&& (and)           || (or)           ! (not)

if ((n < 5) || (n > 10)) System.out.println("hi");
if((n == 7) && (k > 12)) System.out.println("bye");

See Example Generator element for boolean expressions toward end of 4.3
boolean b = true;
for(int j = 0; j < 5; j = j + 1){
    b = !b;
    System.out.println(b);
}
boolean b = true;
for(int j = 0; j < 5; j = j + 1){
    b = !b;
    System.out.println(b);
}
false
true
false
true
false
The loop as scoreboard - a more complex generate & test.

How many numbers between 1 and 100 are divisible by 3 or 7?

```java
int counter = 0;  //initialized outside of loop
for(int i = 1; i <= 100; i = i+1){
    if ((i % 7) == 0 || (i % 3) == 0)
        counter = counter + 1;  // incremented inside loop
}
System.out.println(counter);  // report outside of loop
```
Add up the numbers from 1 to 100 – an accumulator pattern
Set up counter outside loop...

```java
int counter = 0;

for(....) {blah blah}

System.out.println(counter);
```

So: counter enters the loop at 0, and then each value from 0 to 100, in succession, is added to counter
int counter = 0;
for(int i = 1; i <= 100; i = i+1){
    counter = counter + i;
}
System.out.println(counter);

Note: loop body could be:

    counter += i;
Some syntax:

```plaintext
for(init; test; increment)
  stmt;
```

Or

```plaintext
for(....; ....;....){
  stmt1;
  stmt1;
  stmt2;
  ...
  stmtn;
}
```

A Block
Some syntax:

if (boolean)
    stmt;

Or

if(boolean){
    stmt1;
    stmt2;
    ...
    stmtn;
}

A Block
Some syntax:

```plaintext
if (boolean)
    stmt;
else stmt;..... or else {smt1...stmn}
or
if(boolean){
    stmt1;
    stmt2;
    stmntn;
}
else stmt;..... or else {smt1...stmn}
```
Write a complete program that reads in a String, then reports if there are strictly more a’ s (or A’ s) than b’ s (or B’ s) in that string.

“abbaAAaaAbbbB” -> true (7/6)
“monkeybusiness” -> false (0/1)
“junkie” -> false (0/0)
A simple plan for organizing programs

Three phases:

• Gather data / input

• Process the data

• Report answers / results
Input: Scanner + nextLine(); make lower case

Process: Walk the string, counting a’s and b’s

Report: more a’s than b’s?
import java.util.*;

public class AB{
    public static void main(String[] args){
        Scanner sca = new Scanner(System.in);
        System.out.println("enter a phrase");
        String s = sca.nextLine();
        s = s.toLowerCase();
        int ctrA = 0; int ctrB = 0; char c;
        for(int j = 0; j < s.length(); j++){  // walk String s
            c = s.charAt(j);
            if (c == 'a') ctrA++;
            if (c == 'b') ctrB++;
        }
        System.out.println("more a's?? " + (ctrA > ctrB));
    }
}
import java.util.*;

public class AB{
    public static void main(String[] args){
        Scanner sca = new Scanner(System.in);
        System.out.println("enter a phrase");
        String s = sca.nextLine();
        s = s.toLowerCase();
        int ctrA = 0; int ctrB = 0; char c;
        for(int j = 0; j < s.length(); j++) {
            c = s.charAt(j);
            if (c == 'a') ctrA++;
            if (c == 'b') ctrB++;
        } // walk s
        System.out.println("more a's?? " + (ctrA > ctrB));
    }
}
An alternative solution..
import java.util.*;

public class AB{
    public static void main(String[] args){
        Scanner sca = new Scanner(System.in);
        System.out.println("enter a phrase");
        String s = sca.nextLine();
        s = s.toLowerCase();
        int ctr = 0; char c;
        for(int j = 0; j < s.length(); j++){
            c = s.charAt(j);
            if (c == 'a') ctr++;
            if (c == 'b') ctr--;
        }
        System.out.println("more a's?? " + (ctr > 0));
    }
}
The English character set (letters, digits, etc) are laid out in a row, every letter has a fixed position. Rule: arithmetic involving char data is always converted automatically to numerical data:

‘A’ + 30 \rightarrow 95

‘A’*’ A’ = 4225

You can cast to get back a character:

(char)95 \rightarrow - (a dash, or hyphen)

(char)4235 \rightarrow ‘\text{\£}’ (Arabic? Hebrew?)
for(char ch = 'a'; ch < 'f'; ch = (char)(ch + 1)) {
    System.out.print(ch);
}

Comparing chars: chars come in fixed order, each char has a position in the order

What is (‘a’ + 1) (ans: 98) / (‘a’ < 50) is false

What is (char)98 (ans: ‘b’)

What is (‘B’ - ‘b’) (ans: 32) - huh??

If arithmetic/boolean operators appear in exprs involving chars, chars treated as ints!
for(char ch = 'A'; ch <= 'z'; ch = (char)(ch + 1)){
    System.out.print(ch);
}

Output:
ABCDEFGHIJKLMNOPQRSTUVWXYZ\[^__`a
bcdefghijklmnopqrstuvwxyz
for(int n =(int)'A'; n<=(int)'z';n++){
    System.out.print((char)n);}

Same result!
What does this loop do??

String s = “horse”;
String answer = “”;
char c;

for(int j = 0; j < s.length();j++){
    c = s.charAt(j);
    answer = c + answer;
}
System.out.println(answer);
Caesar Cipher

Used by Caesar to communicate with his generals.

A simple linear shift cipher - nowadays - useless.

So if the shift amount is 2

ZOO  ->  BQQ

Shift of 5? 26? 31?
Key idea:
If letters fall at positions 0 - 25, then z is at 25, a is at 0, b at 1, e at 4, m at 12, o at 14.

If z is shifted 2, it goes to z -> a -> b
If z is shifted 5, z -> a -> b -> c -> d -> e

Use mod (%) to do the wrap:
(‘z’+2) % 26  -->  27 % 26 --> 1 = ‘b’
(‘z’+5) % 26  -->  30 % 26 --> 4 = ‘e’
(‘z’+31) % 26  -->  56 % 26 --> 4 = ‘e’
import java.util.*;

public class CipherTest{
    public static void main(String[] args){
        Scanner sca = new Scanner(System.in);
        System.out.println("enter a phrase");
        String s = sca.nextLine();
        System.out.println("enter a shift amount");
        int k = sca.nextInt();
        Cipher c = new Cipher(k);
        System.out.println(s);
        System.out.println(c.encrypt(s));
    }
}

public class Cipher{
    private int shift;

    public Cipher(int s){
        shift = s;
    }

    public String encrypt(String plain){
        plain = plain.toUpperCase();
        String result = "";
        for(int j = 0; j < plain.length(); j++){
            result = result + letterShift(plain.charAt(j));
        }
        return result;
    }
}
private char letterShift(char c){
    // expects c to be either a non-letter or upper case
    if (Character.isLetter(c)){
        int cPos = (c - 'A'); // a position from 0 to 25
        cPos = (cPos + shift) % 26; // deal with overflow
        return ((char)('A' + cPos)); // remake as caps
    }
    else return c;
}

Character.isLetter(c) is like
( 'a' <= c) && (c <= 'z')
The Math class and static methods

• Not all methods are invoked by objects.
• The Math class, for example
• Doesn’t make sense to attach a function like sqrt or max to an object
• All Math fns are static, Math.sqrt() etc.
STOP!

Major big deal:

myKid.getAge();

vs

Math.sqrt(5);
STOP!

Major big deal:

$\text{getAge()}$ serves myKid

myKid.getAge();

vs

$\text{sqrt}$ serves (?) Math

Math.sqrt(5);
double r = Math.max(3.5, 7.1);

double r = Math.sqrt(2.0);

double r = Math.sin(.7);

double r = Math.min(3.5, 7.1);

double r = Math.pow(2, 5);  // 2^5 returned as double
An unusual function in the Math class:

Math.random()

 Gives a random value $r$, $0 \leq r < 1$

What does random mean? Basically: generating many in a row will reveal no pattern...
A preliminary problem:

Out of 100 random numbers, how many are less than 0.5?
More ambitious:

Create an application that considers some large number of random numbers and reports the average of those numbers.
A random number between 0 and 2:

\[ 2 \times \text{Math.random()} \]

A random number between 10 and 13

\[ 10 + 3 \times \text{Math.random()} \]
Random numbers to estimate pi

(aside: Math.PI)

If a circle has radius 1.0, what is its area?

Suppose the surface of the earth is 77% water, 23% land, and a meteor hitting earth is equally likely to hit anywhere.

If 100 hit earth in the last decade, how many would you expect to land on water?

If 41 / 1000 in last century landed in fresh water, what can you conclude?
Area of quarter circle = \( \pi / 4 \)

Length = 1.0 - the unit square

Area of circle = \( \pi \times 1.0 \times 1.0 = \pi = 3.14159 \)
public class RandomPi{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        int trials = 0; int inside = 0;
        System.out.println("Enter number of trials");
        trials = s.nextInt();
        System.out.println("Trials:" + trials);
        double x,y;
        for(int j = 0; j < trials; j++){
            x = Math.random(); y = Math.random();
            if (Math.sqrt((x*x + y*y)) < 1) inside++;
        }
        System.out.println("pi guess: "+
            (double)(4*inside)/trials);
    }
}
Enter number of trials
Trials: 100000
pi guess: 3.14116

Enter number of trials
Trials: 10000000
pi guess: 3.1415424