Computer Science 121 - Lecture 8

Announcements

Ch 4 OWL assignment due next Friday

Programming Assignment 3 due Next Wednesday --
This assignment available at the course website.

Resources - office hours (TAs, me); discussions;
learning resource center Tutoring: Su-Wed 5-10,(M 4-10), 10th floor main library room 1067;

SI group work: M: 8:45-10; W 7:15-8:30 More information at www.umass.edu/lrc/

Next Tuesday is Monday schedule
**integer division**

\[ \frac{5}{4} = 1 \]
\[ \frac{5}{9} = 0 \]
\[ \frac{9}{5} = 1 \text{ (but note: } \frac{9}{5.0} = 1.8) \]

**remainder - mod - %**

\[ 5 \mod 3 = 2 \text{ (remainder, after dividing 5 by 3)} \]
\[ 5 \mod 9 = 5 \text{ (remainder, after dividing 5 by 9)} \]
\[ 9 \mod 5 = 4 \text{ (remainder, after dividing 9 by 5)} \]
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
**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

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

Ans:

```
4/3 = 1  5%3 = 2
1 0 1
2 0 2
3 1 0
4 1 1
5 1 2
```
for(int j = 3; j < 12; j = j + 4){
    System.out.println(j);
}

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

```plaintext
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));
}
for(int j = s.length() - 1; j >= 0; j--){
    System.out.print(j + " ");
    System.out.println(s.charAt(j));
}
Boolean connectives

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

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

boolean b = true;
for(int j = 0; j < 5; j = j + 1){
    b = !b;
    System.out.println(b);
}
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 - accumulater pattern

```java
int counter = 0;
for(int i = 1; i <= 100; i = i+1){
    counter = counter + i;
}
System.out.println(counter);
```

How would you add numbers from variable first to last, where first, last, could be anything?
import java.util.Scanner;

public class AddUp{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter start,stop nums");
        int first = s.nextInt();
        int last = s.nextInt();
        int sum = 0;
        for (int n = first; n <= last; n=n+1){
            sum = sum + n;
        }
        System.out.println("sum from "+first+" to "+last);
        System.out.println(": " + sum);
    }
}
Some syntax:

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

Or

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

A Block
Some syntax:

```java
if (boolean) stmt;
```

Or

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

A Block
Some syntax:

```plaintext
if (boolean)
    stmt;
else stmt;..... or else {smt1...stmn}
or
if(boolean){
    stmt1;
    stmt2;
    stmtn;
}
else stmt;..... or else {smt1...stmn}
```

A Block
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).

“abbaAAaaAabbbB” -> true (7/6)
“monkeybusiness” -> false (0/1)
“junkie” -> false (0/0)
Casting

What is the value of

\[(\text{int})'E'\]

??

Casting: reinterpreting data in a different form

Also: \((\text{int})3.14 \rightarrow 3\)
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 → 95
'A'*'A' = 4225

You can cast back to get a character:
(char)95 → - (a dash, or hyphen)
(char)4235 → 'א' (Hebrew aleph?)
for(char c = 'a'; c < 'f'; c = (char)(c + 1)) {
    System.out.print(c);
}

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 operators appear in expressions involving chars, the chars are treats as ints!
for(char ch = 'A'; ch < 'z'; ch = (char)(ch + 1)){
    System.out.print(ch);
}

Output:

ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz

Output:
Binary conversion - b a binary string

10011

16s  8s  4s  2s  1s  - these are powers of 2

Decimal value:  $16 + 2 + 1 = 19$
int pow = 1;
int total = 0;
for(int j = b.length()-1; j >= 0; j--){
    if (b.charAt(j) == '1') total = total + pow;
    pow = 2*pow;
}

At the end, total holds decimal integer representation of b
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 funcion like sqrt or max to an object
• All Math fns are static, Math.sqrt() etc.
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 checks to see if the average of those numbers is $1/2$ (0.5).
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