CS 121 – Intro to Programming:Java - Lecture 3
Announcements

Course home page:

http://twiki-edlab.cs.umass.edu/bin/view/Moll121/WebHome

First programming assignment due in class on Thursday. Screen shot instructions are on line, under course resources. Book material is ungraded. Yes: more work coming!

Honors section - Friday, 11:15 (time: almost certain). See me after class if you’re still on board or want to join.

Captive Lab?
More Administration

The grading formula:

- Programming assignments: 20% (6-8 of these..)
- OWL assignments: 15%
- Midterm exam: 25%
- Final Exam: 40%

Also: to get a C in the course, you must get at least a C on the final!
Midterm - 10/17
The layout of the course: Tuesday: a big-picture lecture day

Thursday - the so-called discussion class - will be just that: discussion, + a fast-paced interactive problem-solving experience. Here are the rooms…

DISCUSSION 1: LGRT 321

DISCUSSION 2: HASA 113

DISCUSSION 3: MRST 211

DISCUSSION 4: ECSCII 119 (changed; not sure: go to this one)

You are expected to attend class - but roll isn’t taken. Miss classes and lectures at your own peril.

Bring your brains.
A typical Java program
A very, very simple application

Class G

main
public class Howto{

// a baby intro example

public static void main(String args[]) {
    System.out.println("Welcome to 121");
    System.out.println("3 + 5");
    System.out.println(3 + 5);
}
}

Notice: this Java program (application) consists of a single class
A two class model

This is more like 1000 class application than it is like 1 class application!
public class InfantTester{

    public static void main (String[] args){
        Infant myKid = new Infant("Lizzie",4);
        int lizAge = myKid.getAge();
        System.out.println("my kid's name is " + myKid.getName());
        myKid.anotherMonth();
        System.out.println("my kid is now " + myKid.getAge() + " months");
    }
}

Important - the Infant class is “off-stage” -- but it’s been compiled in the same directory, and so we can use it. And we have.
public class Infant{

    private String name;
    private int age;  // in months

    public Infant(String who, int months){
        name = who;
        age = months;
    }

    public String getName(){return name;}

    public int getAge(){return age;}

    public void anotherMonth(){age = age + 1;}
}

Infant myKid = new Infant("jordie", 9);
Infant theirKid = new Infant("maya", 7);
myKid.getAge();
theirKid.getAge();
The Chairs class:

We’re interested in creating Chairs objects (for a furniture store..)

Each chair has: a **name** attribute, a **count** attribute, a **price** attribute

Types / classes for each of the attributes?

And how would you make one

Chairs seats = ??????????
Chairs seats = new Chairs(“desk”, 21, 79.95);
How do you retrieve the price of one?
How do you retrieve their count?

What should this statement do;
   double value = seats.totalValue();
How would you calculate this?
public class ChairTester{

    public static void main(String[] args){

        Chairs seats = new Chairs("easyboy", 12, 239.95);
        System.out.println("total value $" +
              seats.totalValue());
        int count = seats.getCount();
        seats.setCount(count + 1);
        System.out.println("total value $" +
              seats.totalValue());
    }
}

public class Chairs{

    private String name;
    private int count;
    private double price;

    public Chairs(String chairName, int ct, double cost){
        name = chairName;
        count = ct;
        price = cost;
    }
    public String getName(){return name;}
    public int getCount(){return count;}
    public double getPrice(){return price;}

    public void setCount(int ct){count = ct;}
    public void setPrice(double cost){price = cost;}
    public double totalValue(){return (price * count);}
}
Assignment statements

The cell model

```java
int n = 5; int k;

double dough = 15.95;

Infant screamer = new Infant("jack", 11);

n = n + 1;

n = n * n;
```
int dog, lamb, pig;
lamb = 2; pig = 7;
dog = lamb + 3;
lamb = dog * lamb;
lamb = dog * lamb;
lamb = pig;
pig = lamb;
int dog, lamb, pig;
lamb = 2; pig = 7;
dog = lamb + 3;
lamb = dog * lamb;
lamb = dog * lamb;  // now lamb = 50, pig = 7
dog = pig;  // lamb = 50, pig = 7
pig = lamb;  // lamb = 50, pig = 50
lamb = dog;  // lamb = 7, pig = 50

// dog is a temporary repository