CS 121 - Intro to Programming:Java - Lecture 4

Announcements

Power issues - book/OWL down 4 PM fri - sun eve (so prog 1 due date pushed off until Mon morning). Prog 2 now up on website.

Introductory Survey - Please do it!

Ch 1,2 OWL hwk due tonight at 11

Ch3 Embedded problems due Wed at noon (ignore regular expression trapdoor problems).

Office hours - mine posted, TAs later today

http://twiki-edlab.cs.umass.edu/bin/view/CS121Fall2010/WebHome

CourseWork link at top of website has due dates
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");
    }
}

Variables are “live” inside main: myKid,lizAge
* Methods that return something create/adjust contents directly (Lines 4,5,6,8);
* void methods don’t return anything, may write to screen,or adjust internals of objects(L6,7,8)
public class Infant{

    private String name; // name, age - attributes
    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;}
}

Question:

Line 5 associates a value of 4 with myKid:

```java
int lizAge = myKid.getAge();
```

after line 7 executes:

```java
myKid.anotherMonth();
```

what value is associated with lizAge?
In Java all data is tagged (that is, typed)

    int number;

double v;

Infant kid;
In Java: two broad kinds of data:

**objects** (roughly, things - defined by classes)

**primitives** (numbers, true/false values etc.)

Variables are associated with data

For primitives anyway, there is a simple **cell model** for variables and their values

```java
int num = 6;
```
Assignment Statements and Identifiers

An identifier is the name of a variable (or method, or class..)

```java
int number = 7; // number now "holds" 7
number = 4; // number now "holds" 4
number = number + 2; // number now "holds" 6
```

Assignment is NOT equality!

Assignment is an **action** operator: **Compute** the RHS,
Then **copy** the result to variable named on the LHS
int number = 7; // number now “holds” 7
number = 4; // number now “holds” 4
number = number + 2; // number now “holds” 6
The cell model and assignment

```java
int j = 5;
int k = 10;
int m = 2;

What values do j, k, m, below, hold after stmt?

j = j + k;  // 1
k = 2 * k + j;  // 2
m = m + k;  // 3
```
The cell model and assignment

```java
int j = 5;
int k = 10;
int m = 2;
```

What values do j, k, m, below, hold after stmt?

```java
j = j + k;   // 1 - after execution, j holds 15
k = 2 * k + j;  // 2
m = m + k;    // 3
```
The cell model and assignment

```java
int j = 5;
int k = 10;
int m = 2;
```

What values do j, k, m, below, hold after stmt?

```java
j = j + k;  // 1 - after execution, j holds 15
k = 2 * k + j;  // 2 - after execution k holds 35
m = m + k;  // 3
```
The cell model and assignment

```java
int j = 5;
int k = 10;
int m = 2;

What values do j, k, m, below, hold after stmt?
j = j + k;  // 1 - after execution, j holds 15
k = 2 * k + j;  // 2 - after execution k holds 35
m = m + k;  // 3 - after execution m holds 37
```
int num = 7;
num = num + num;

---

int value = 4;
value = value + 1;

---

int score = 10;
score = score + score;
score = score * score;
score = score / 2;
myKid.anotherMonth();

lizAge = myKid.getAge();
Data in Java

A quick tour
**Primitive Data Types**

- Objects are Java’s main currency
- Too tedious for them to be the only currency
- Primitive data types (8): integers (4), floats (2), char, boolean.

Statement like these are fairly common:

```java
long count = 0;     // long is like int, but larger range
double bigPapiAvg = 0.224;  // nums with decimal pts
boolean chewsTobacco = false;
char averageGrade = 'C';   // note the single quotes
```
Strings - A very important class.

String greeting; String greeting2;
greeting = new String("ola");
greeting2 = new String("howdy");
greeting = greeting2;
System.out.println(greeting); // prints howdy

Some caveats:
1) Strings are not primitives (unlike double, int, boolean) String is a standard (Java library) class

2) There's a shorthand for String creation:
   greeting = "ola"; // works fine

3) String class comes with extensive functionality
String pupName = “spot”;  
int len = pupName.length(); // len assigned 4  
char ch = pupName.charAt(1); // ch is assigned ‘p’  
char ch = pupName.charAt(0); // ch is assigned ‘s’  
String huh = pupName.concat(“less”); // spotless  
String bigHuh = pupName.toUpperCase(); // SPOT

Where do I find out about the String class... (hold on)
The entire Java API is online, at

http://download.oracle.com/javase/6/docs/api/index.html

This is the link we use in the online Book.

How about the API for the String class?
Our current view of the OO programming landscape..
public class LooseLeaf{
    // models a looseleaf notebook, counts blank sheets
    private int blankPages;
    private String name; // notebook owner

    public LooseLeaf(String who, int blanks){
        blankPages = blanks;
        name = who;
    }

    public int getBlankPages(){return blankPages;}

    public String getName(){return name;}

    public void setBlankPages(int amount){
        blankPages = amount;
    }
}
A story...

Let’s make two loose leaf notebooks, jack’s (50), jill’s(20)

Let’s move 20 blank pages from Jack’s notebook to Jill’s notebook
public class LooseLeafTester{
    public static void main(String[] args){
        LooseLeaf jacksBook = new LooseLeaf("Jack",50);
    }
}

public class LooseLeafTester{
    public static void main(String[] args){
        LooseLeaf jacksBook = new LooseLeaf("Jack",50);
        LooseLeaf jillsBook = new LooseLeaf("Jill",20);
    }
}
public class LooseLeafTester{
    public static void main(String[] args){
        LooseLeaf jacksBook = new LooseLeaf("Jack",50);
        LooseLeaf jillsBook = new LooseLeaf("Jill",20);
        int jacksBlanks = jacksBook.getBlankPages();
    }
}

public class LooseLeafTester{
    public static void main(String[] args){
        LooseLeaf jacksBook = new LooseLeaf("Jack",50);
        LooseLeaf jillsBook = new LooseLeaf("Jill",20);
        int jacksBlanks = jacksBook.getBlankPages();
        jacksBook.setBlankPages(jacksBlanks - 20);
        int jillsBlanks  = jillsBook.getBlankPages();
    }
}

```java
public class LooseLeafTester{
    public static void main(String[] args){
        LooseLeaf jacksBook = new LooseLeaf("Jack",50);
        LooseLeaf jillsBook = new LooseLeaf("Jill",20);
        int jacksBlanks = jacksBook.getBlankPages();
        jacksBook.setBlankPages(jacksBlanks - 20);
        int jillsBlanks = jillsBook.getBlankPages();
        jillsBook.setBlankPages(jillsBlanks + 20);
    }
}
```
public class LooseLeafTester{
    public static void main(String[] args){
        LooseLeaf jacksBook = new LooseLeaf("Jack", 50);
        LooseLeaf jillsBook = new LooseLeaf("Jill", 20);
        int jacksBlanks = jacksBook.getBlankPages();
        jacksBook.setBlankPages(jacksBlanks - 20);
        int jillsBlanks = jillsBook.getBlankPages();
        jillsBook.setBlankPages(jillsBlanks + 20);
        System.out.println(jillsBook.getBlankPages());
    }
}

public class Car{

private String make; // manufacturer
private double fuelCapacity;
private double fuelAmount;

// the Car constructor

    public Car(String what, double cap, double amt){
        make = what;
        fuelCapacity = cap;
        fuelAmount = amt;
    }
}
// the Car methods
public String getMake(){
    return make;
}
public double getCapacity(){
    return fuelCapacity;
}
public double getFuel(){
    return fuelAmount;
}

public void setFuel(double amt){
    fuelAmount = amt;
}

public double unusedCap(){
    return (fuelCapacity - fuelAmount);
}
public class CarTester{
    public static void main(String[] args){
        Car mine = new Car("Ford",15,9.5);
        double amt = mine.unusedCap();
        System.out.println("fill-up cost " + 2.65*amt);
    }
}
public class CarTester{
    public static void main(String[] args){
        Car mine = new Car("Ford",15,9.5);
        double amt = mine.unusedCap();
        System.out.println("fill-up cost " + 2.65*amt);
        mine.setFuel(mine.getCapacity());
    }
}
public class CarTester{
    public static void main(String[] args){
        Car mine = new Car("Ford",15,9.5);
        double amt = mine.unusedCap();
        System.out.println("fill-up cost " + 2.65*amt);
        mine.setFuel(mine.getFuel() + amt);
    }
}