Computer Science 121 – Spring 2014 – Discussion #1

Announcements: Remember that lectures are available in “movie” form at course website. Due dates at website at CourseSyllabus link as well as in OWL. Read text Ch 3, do embedded problems for Wednesday. Second programming assignment due Tuesday. Hand in through OWL; Course Website: https://twiki-edlab.cs.umass.edu/bin/view/CS121Spring2014/WebHome

Lecture Synopsis–Talked last week mainly about the coordination between class definitions, and constructor and method invocations in driver class; also talked about Strings, data, primitives, cell model

Issues to review: how data is tagged(typed); class attributes; flow of control at the statement level; flow of control at class level; How attributes get values. Relationship between class definitions and class objects.

Here is a TubTester driver class [note: gallon of water weighs 8.6 lbs]

```java
public class TubTester{
    public static void main(String[] args){
        BathTub b = new BathTub("King",60);//60 gal capacity
        BathTub t = new BathTub("Queen",70);
        System.out.println(" capacity of both tubs: "+
                (b.getCapacity() + t.getCapacity()));
        System.out.println("filled wt of larger tub: "+t.totalWt());
    }
}
```

1. What outside class does TubTester make use of?
2. What files support these Java classes?
3. What are that class’s attributes? Methods? Mutator methods?
4. Can you write the constructor?
5. Can you write the getCapacity() method? The totalWt() method?

Here is the Car class – the questions at the end resemble what you need to do for programming assignment 2.

```java
public class Car{
    String make; // manufacturer
    double fuelCapacity;
    double fuelAmount;

    public Car(String what, double cap, double amt){
        make = what;
        fuelCapacity = cap;
        fuelAmount = amt;
    }

    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(){ // noteworthy: does a calculation
        return (fuelCapacity - fuelAmount); }
}
```
Now write a driver class (main class) that performs these actions

6. Make a car called myCar, a “Ford”, fuel capacity = 15.0, tank is half-filled.
7. Make a second car, called herCar, an Audi, cap = 16.0, amt = 12.0.
8. Write a statement that prints the unused fuel capacity of the Audi.
9. Write a statement that prints the sum of the capacities of the cars.
10. Steal exactly the right amount of gas from the Ford to fill the Audi.
10. Write a statement that copies into a variable called fordGas the gas now in myCar.

Here is the PaperPad class

```java
public class PaperPad{
    private int sheets; // sheets in pad
    private String owner; // pad owner
    public final double THICKNESS = .0093;                                       // final means a constant
    public PaperPad(String who, int count){
        owner = who;
        sheets = count;
    }

    public int getSheets(){return sheets;}
    public String getOwner(){return owner;}
    public void setOwner(String newOwner)
    {
        owner = newOwner;
    }

    public void tearOff(){sheets = sheets-1;}// tear off a sheet
}
```

Suppose PaperPad is on your computer and has been compiled. Write a complete class called PadTester that includes main, and which implements the following “story”: Make a pad of 100 sheets of paper called herPad, owned by Jill; make another pad, owned by Ike, with 80 sheets – call that one hisPad. Tear off 4 sheets of the pad owned by Ike, and then print how thick the pad is. Finally, Ike gives his pad to Jill (so that she is the new owner) – write a statement to accomplish this transfer.

Note: if this goes well, programming project 2 will be straightforward.