Announcements:
Midterm will be released on Thursday evening. See OnlineExams link on course website for more details about exam format, etc.

Lecture Summary: Chapter 5, primary attention to methods, their mechanics, how they are deployed, the “art” of their organization.

Below: an old sample midterm, along with some supplementary problems.

Consider this class:

```java
class Office{
    private int number;
    private String occupant;
    private int area;

    public Office(int n, String occ, int area){
        // complete this constructor for part a)
    }

    public void setOccupant(String who){occupant = who;}
    public int getNumber(){return number;}
    public String getOccupant(){return occupant;}
    public int getArea(){return area;}
    public void enlargeOffice(int amount){area += amount;}
}
```

1. Suppose you are writing a driver class called OfficeDriver, which uses the Office class that is displayed above.

   a) The Office constructor is incomplete. Provide the constructor body. (Hint: use “this”).

   b) Now write a single statement that should appear in main in OfficeDriver, and which creates an Office object called office1. That object should refer to room number 20, should be occupied by Jill, and should have area 200.

   c) Write a single statement in the driver that creates an Office object called office2, which refers to room number 22, is occupied by Jack, and has area 240.

   d) Now add statements to the driver that, in effect, swap Jack’s and Jill’s offices (that is, after the exchange, Jack should be in room 20 and should be associated with office1, and Jill should be in room 22 and associated with room 22). Your answer should not alter the Office class in any way.

   e) Suppose someOffice refers to some Office object. Write a statement in the driver that increases the area of someOffice by 100. Your statement should use an existing method or methods in the Office class to enlarge someOffice by 100.

2. Consider the following driver class:

```java
class ParkingTester{
    public static void main(String[] args){
        // make a parking lot, owned by Dana, capacity of 200, 85 cars in lot
        Lot myLot = new Lot("Dana",200,85);
        myLot.changeOwner("Hilary"); // Hilary is new owner of myLot
        myLot.departing(15); // 15 cars have left the lot
        myLot.entering(20); // 20 cars have entered the lot
        System.out.println("fraction filled: ",
                          (double)myLot.getCarCount()/myLot.capacity()); }}
```
When executed, this driver class produces the following line of output:

Fraction filled: 0.45

Create a Lot class that will make the ParkingTester driver class run properly. (Be systematic here: what are the attributes, what are the methods, what must the constructor look like?)

Short answers

a) What do each of these statements print?
   i) System.out.println((int)3.999);
   ii) System.out.println((char)('a'+1));

b) Write a for loop that prints in a column all values, from largest to smallest, of the numbers from 1001 down to 101 that are divisible evenly by 5 (examples: 605, 770)

c) Write a method in the Lot class (problem 2) called emptyLot, with no input arguments, which returns true if the calling Lot object is empty (has no cars).

d) The code below is a complete program in a single class called CalcAverage. It reads two int values from the keyboard, and then prints their average. However the lines in the program have been scrambled. Unscramble the lines so that the program does its intended job. Your answer should be a sequence of line numbers, 1 through 11, that give an order that would make the program work correctly. For example your answer might look like this: 2-3-4-5-6-7-1-11-10-9-8.

Your program should show the following kind of interaction on the console

Enter two integers
12
13
Average: 12.5

1. }
2. System.out.println("Enter two integers");
3. public static void main (String[] args){
4. public class CalcAverage{
5. int num1 = s.nextInt();
6. import java.util.*;
7. int num2 = s.nextInt();
8. double avg = (num1 + num2)/2.0;
9. System.out.println("Average: "+ avg);
10. Scanner s = new Scanner(System.in);
11. }

EXTRA- a little harder.. Suppose s is a String consisting of all digits, for example 12096, or 445. Write a loop that prints the digits of s back to the console in a row – except for 5’s. 5’s should generate a new line. Thus, if s = 3456, then your loop should print this:

34
6

(Hint: do an easier problem first – just print the digits in a row, character by character.)