Lab 7 Classes and Objects – part 1:

You will write a Java class named ClassSection to represent a class section like our very own CIS 131. This class has a number of data members (aka instance variables, aka properties aka attributes) which we all know so well from looking at class schedules on the Pima web site:

1. CRN, like 20008
2. Department code, like CIS
3. Course number, like 131
4. Instructional mode, like online, classroom or hybrid
5. Campus, like East or West
6. Meeting days ("N/A" for online classes, 1234567 for Monday through Sunday. TTh would be 24)
7. Meeting times ("N/A", or free form such as: "12 noon to 2pm")
8. Capacity (maximum number of students who can enroll in it)
9. Enrollment (number of students actually enrolled in it)
10. Instructor's ID number (in schedules you see the instructor's name due to a database operation called a join. Take CIS162 and 182 for more info on this.)

The class representing a class section will have all of the attributes listed above along with getter and setter methods for each of them. It will also have a default constructor that takes no parameters. This constructor will assign "" (empty string) to String attributes, 0 to int attributes and 0.0 to double attributes, if any.

The ClassSection class will also have a constructor that accepts a parameter for each attribute. Lastly, it will have a toString method that returns a String containing labels and values for all of the attributes.

A sample of the String produced by toString() is shown below; notice that it uses new line characters to display parts of the String on different lines, lining up the data as is shown below:

```
**********************************
CRN :                20008
Department :         CIS
Course Number :      131
Instructional Mode : Online
Campus:              East
Meeting Days :       N/A
Meeting Times :      N/A
Capacity :           30
Enrollment :         30
Instructor's ID :    654
**********************************
```

After creating the Java file for class section, you need to create an application that uses it. We often call these applications driver programs. For our purposes, the driver program will

1) create one instance of a class section object using its default constructor.
2) Then it will use the object's setter methods to assign values to its attributes.
3) After doing this, the driver will use either println or printf to display the String returned by toString.
4) Next, it will create a second instance of a class section object using the constructor that accepts values for all of its attributes.
5) Again, it will it will use either println or printf to display the String returned by toString.

Provide your Java source code both the class section and the driver program.
Name the files for this lab: Lab_1_Driver.java and ClassSection.java

Since we have multiple files, compress them together into a file such as CIS131-DFreitag-Lab-7.zip and then upload it to D2L.