Introduction to PAD2

Dr. Mark C. Lewis

1-17-2008

Opening Discussion

- At the beginning of every class we will do a brief discussion of some questions I provide and any questions you might have. Typically mine cover the previous lecture and the reading for the current one.
- What do you know about Java and Object-Orientated programming/design?
- Never reboot or turn off these machines!

Basic Information

- Log onto the machines and bring up a browser.
- Course web page: http://www.cs.trinity.edu/~mlewis/CSCI1321-S08
- Office: HAS 201K
- Phone: 999-7022
- e-mail: mlewis@trinity.edu
- Office hours: 4:30-6:00 M, 3:30-6:00 W, 9:30-11:00 R, and 1:30-6:00 R, or by appointment
- Afternoon times are open labs. Monday and Wednesday are in 228 while Thursday is in 340.

Text and References

- “Java Software Structures” by John Lewis and Joe Chase.
- There is also a “pamphlet” that I have written that we will be using. It is called “From C to Java” and you can get to the PDF from the links page.
- There are lots of books that you could buy on Java, but for most basic information the web is actually a great source, and it is free.
- If you are going to do much programming without net connection you can download the API in HTML or buy the Nutshell books.

Course Description

- This course is the second CS course for majors here at Trinity. It will continue to build your abilities to solve problems on the computer, but it will switch focus to object-oriented design and programming.
- We will learn how to use a number of simple data structures in an OO framework. We will also look at the construction of GUIs and the graphics APIs in Java.

Assignments/Project

- The most significant work you do for this course will be for the assignments. All the assignments will be based on a single program. With each assignment you will construct a bit more of the code to create your final result. You will also be using code that I have written as well as the general Java libraries.
- There are certain coding standards that you need to follow. You will also turn in a design for each assignment.
- More about the project in a few slides.

Grades

- Your grade in this class is determined by a combination of 5 parts.

Assignments (٨)	٤٠%
Tests (٢)	٣٠%
Quizzes (٦ drop ١)	١٠%
Interclass Problems	١٠%
Class Participation	١٠%

Extra Credit

- You can also receive extra credit in this course in a number of ways.
  - Quizzes and tests will almost all contain extra credit questions. These will typically be more challenging than the normal questions.
  - You can send me links to information you find that demonstrates that you have been thinking about course material outside of class.
  - Do some competitions at TopCoder.com. They have regular competitions where you attempt to solve a number of problems as fast as you can. They allow you to code in Java, C++, and C#.

Lectures/Schedule

- The web page also has a schedule for the entire semester. It includes not only the topics, but also readings and due dates for everything that receives a grade.
- On this page I will also provide links to the notes I put on in PDF format. I post them at least the night before class. I do this because you should never feel like you have to write down what is on the slides. Focus on listening so you can “grok” the material and take notes on interesting points not in the slide text.

Think More - Work Less

- For those of you who don’t know this yet, my overriding objective in this class is to get you to think. During the course of the semester, several lights should come on in your head as different ideas start to make sense. Object-orientation is not just a minor modification of the imperative style you learned last semester.
- The more you think through assignments the less time you will spend coding.

More on the Project

- The project that you will be doing for the semester is a game. Exactly what type of game it is will be up to you. There are certain restrictions on what you can produce though.
- Your game will consist of one of more different “screens” and possibly a player that navigates through them. Each screen is a grid of “blocks” and can have a number of “entities” on it the player can interact with or that interact with one another.
- People have made lots of different types of games. The only real limitation is that it needs to be 2-D.

Project Ideas

- There are many, many different games that could be written using this basic model and I want you to use your imagination in coming up with an idea.
- Two broad categories that fit this model are games similar to the “side scroller” or top-down view games like older RGPs. Even Pac-Man can be written using this model. Civ is a stretch, but not impossible. Tetris and break-out also.
- Board games can also be done, but the framework is less intended for that style of game.

Eclipse IDE

- We will use the an Integrated Development Environment called Eclipse. This is open source and is one of the most widely use professional Java IDEs. You can go to eclipse.org and download a version of Eclipse for your room if you want. It is installed on all of our machines here on campus.
- You can also work with Java on the command line and occasionally in class we will look at how you might do that.

Object Orientation

- There are lots of ideas that go into the concept of object orientation and many people would disagree with what parts are most important.
- The one aspect that is generally agreed upon is that an OO language needs to have encapsulation. This is the binding of data and the code that works on it together into entities called objects.
- The idea of OO comes from the real world where we interact with objects. In Java we call the information in an object the members and the things we can do with them the methods.
- Much of the power of encapsulation comes from the ability to hide things.

Class Based OO

- Java is a class based object oriented language. What we write in code is classes and objects are created that are based on the classes.
- An analogy that I like is that the class is like a blueprint and the objects are the actual objects that we build from the blue print.
- The class specifies what information the objects store and what the objects are able to do.
- The things we put in the classes can be given a visibility so that we can hide things that other people don't need to know about.

Example Coding

- For fun, we will now write some code in Java using Eclipse. The next two days we will look more at object orientation and Java and the things I write now will make more sense.

Minute Essay

- At the end of every class I will have you write me a “minute essay”. This is a few sentences that shouldn’t take you much more than a minute. I will typically ask some type of question for you to answer. You can also provide any form of feedback on anything from the class you want to. Make sure your names are always on these as I use them for attendance and feedback.
- What are your thoughts on the class description? What do you want to get from this course?