

CS 683 Emerging Technologies
Fall Semester, 2004
Doc 1 Introduction
Contents

Reading Assignment	2
Crash Policy	3
Prerequisites	5
Warning about the course – Bleeding edge	9
Course Contents	10
Continuation-based Web Servers	11
Mobile Computing (JMEE)	15
Object-Relational Mapping	16
Application Framework	17
Exercises.....	18

Copyright ©, All rights reserved. 2004 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

Reading Assignment

CSS - For Thursday Sept 2

CSS Tutorial: Starting with HTML + CSS at:

<http://www.w3.org/Style/Examples/011/firstcss>

Chapter 2 of *Cascading Style Sheets, designing for the Web*,
by Håkon Wium Lie and Bert Bos at:

<http://www.w3.org/Style/LieBos2e/enter/>

Dave Raggett's Intro to CSS at:

<http://www.w3.org/MarkUp/Guide/Style>

Crash Policy

The crash list is ordered by seniority.

Crashers with the most graduate credit from SDSU will be accepted first

If there are more students tied at the top of the crash list than there are available seats, students will be selected at random from the top of the list

Role of the crashers will be taken each class period. Crashers attending all classes will be given priority over all crashers that do not attend all classes

Crashers must submit an unofficial SDSU transcript to be prioritized on the crash list

If you have not submitted a transcript at the time students are let into the course, you will not be prioritized

Students given an add code have a limited time to use the add code. (See time line below)

Students that do not use their add code in the allotted time will not be allowed to use that add code and will lose their priority in the crash list for this course.

When students will be added

If there is room in the course students will be added in class on

- Sept 7
- Sept 14
- Sept 16

Students not in attendance at the class are assumed not interested in crashing the course and their spot in the class will be given to someone else

Students added Sept 7 must use the add code by noon Friday Sept 10

Students added Sept 14 must use their add code by noon Thursday Sept 16

Students added Sept 16 must use their add code by noon Monday Sept 20

Prerequisites

Graduate status

Must know Java

- Java's syntax
- Basic Java api
- Basic development with Java

You will be writing the following things in Java

- Web Applications
- Programs to run on PDA/Cell phone
- Programs that interact with a database

I assume you have some database experience

You will be installing & using

- Tomcat - <http://jakarta.apache.org/tomcat/>
- Ant - <http://ant.apache.org/>
- Cocoon - <http://cocoon.apache.org/>
- Hibernate - <http://www.hibernate.org/1.html>
- Spring - <http://www.springframework.org/>
- JMEE - <http://java.sun.com/j2me/index.jsp>

As a student in an advanced graduate computer science course I assume that you can handle this

If you don't have a good Java IDE you should start using

- Eclipse - <http://www.eclipse.org/>

In dealing with CSS you will find the

Firefox - <http://www.mozilla.org/products/firefox/>

Java & Smalltalk

We will look at several technologies implemented in Smalltalk

We will also cover similar technologies in Java

Students can select which technology to work with

Student should be able to read Smalltalk

Grading

I do not give extra credit assignments

Grades in the course will be based on

- Homework
- Quizzes if needed

There will be between 3-5 assignments

Each assignment is given a numerical score

Assignments are averaged to get a course score

Course score is used to determine the course grade

I do not

- Give extra credit assignments
- Consider improving performance during the semester
- Consider the impact of your grade on your career
- Give international students special consideration

CS 683 is an advanced graduate course

If you dealing with new culture and educational system taking an advanced graduate course may be a bad idea

Warning about the course – Bleeding edge

Leading edge is called the bleeding edge

- Documentation is sparse or lacking
- Tools may be lacking & buggy
- APIs may not be fully developed

This course will cover a lot of material

Presentation of material will be rough

This course will be a lot of work

Don't fall behind

Course Contents

Applications in Complex World

We will look at

- Continuation based Web servers
- JMEE (Java for mobile devices)
- Object-relational mapping layers (GLORP, Hibernate)
- Framework for application development (Spring)

Continuation-based Web Servers

Seaside – Smalltalk based

Cocoon – Java, Javascript based

GUI applications started in 1970's

Web application development violate GUI best practices

- No MVC
- No callbacks

The begging of Continuation-base Web servers

- Paul Graham - <http://www.paulgraham.com/index.html>
- Yahoo Stores

Simple Seaside Example

Complete Smalltalk Source

```
renderContentOn: html
  html title: 'Continuation Example'.
  html heading: 'The First Page'.
  html
    anchorWithAction: [self tryMe]
    text: 'Start'

tryMe
  | name count message |
  name := self request: 'Your name'.
  count := 1.
  message := name , ' ready to stop yet? '.
  [self confirm: message , count printString]
    whileFalse: [count := count + 1].
  self inform: 'Good bye'.
```

Java pseudo Code

```
renderContentOn(WAHtmlRenderer html) {  
    html.title("Continuation Example");  
    html.heading("The First Page");  
    html.anchorWithAction( tryMe, "Start");  
}
```

```
tryMe() {  
    String name = request("Your name");  
    int count = 1;  
    String message = name + " ready to stop yet";  
    while ( !confirm( message + count) ) {  
        count = count + 1;  
    }  
    inform( "Good bye");  
}
```

Java Struts & Seaside Comparison

Simple Web log implementation

Designed by Java programmers to compare Java Frameworks

<http://www.waferproject.org/index.html>

<http://www.waferproject.org/weblog-prototype/index.jsp>

Contrast Java Struts implementation and Seaside implementation

	Java Struts	Seaside
Development time	~1 week	~6 hours

Mobile Computing (JMEE)

Mobile computing devices are everywhere

- Cell Phones
- PDA

Cell Phone development in San Diego

- Qualcomm
- Kyocera
- Nokia

Computer Chips are in many items

- \$20 bill
- Car keys
- Soon will replace bar codes

Object-Relational Mapping

Many applications use databases

Object-oriented programming is now common

Need to translate between objects and database tables

JDBC is a low level connection to database

GLORP & Hibernate translate between objects & tables

Hibernate Write Example

```
Transaction tx= session.beginTransaction();
```

```
Cat princess = new Cat();  
princess.setName("Princess");  
princess.setSex('F');  
princess.setWeight(7.4f);
```

```
session.save(princess);  
tx.commit();
```

Application Framework

Some think that EJB & J2EE are too complex

Applications are complex enough

Spring developers believe:

- J2EE should be easier to use
- Best to program to interfaces
- Checked Exceptions are overused
- Testability is essential

Exercises

1. Using Firefox or a similar tool view the CSS of

- <http://www.eli.sdsu.edu/courses/fall04/cs683/syllabus.html>
- <http://www.csszengarden.com/>

You might find the Web Developer Toolbar useful.

2. Using Firefox or a similar tool find the html error in

<http://www.eli.sdsu.edu/courses/fall04/cs683/syllabus.html>

3. Using Firefox or a similar tool find the deprecated elements in

<http://www.eli.sdsu.edu/courses/fall04/cs683/syllabus.html>

4. Using your Java IDE (or Eclipse if you do not already have a Java IDE) implement a stack and write JUnit tests for the stack.

See <http://junit.sourceforge.net/> for documentation about using JUnit.

See <http://www.junit.org/index.htm> to download JUnit

See Using JUnit With Eclipse IDE

<http://www.onjava.com/pub/a/onjava/2004/02/04/juie.html> and