

Course Number, Course Title and Semester Hours

CSC 330 Computer Science III 4sh – Fall 2008

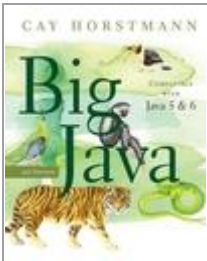
Course Coordinator

David J. Powell

Current Catalog Description

This course introduces concepts and methodologies to design and implement a distributed, multi-tier application. Students will cover advanced java features and look at advanced graphical user interface topics, multithreading, networking, java database connectivity and web applications. Prerequisite: CSC 230. Co requisite: MTH 206. Offered fall

Textbook

<p>Big Java, Third Edition Cay Horstmann, 2008</p> <p>ISBN: 978-0-470-10554-2</p>	
---	--

References

[Sun Swing Tutorial](#) – Superb tutorial on the extensive number of swing components

[Sun Java Tutorials](#) – Top page with specialized links to various topics that we will cover (e.g. internationalization, 2D Graphics)

[Javadoc Tutorial](#) - This is a link to a pdf file for a simple javadoc tutorial

[How to Write Doc Comments for Javadoc Tool](#) - Sun Microsystems

[Java Documentation Comments](#) – O'Reilly

[Writing Robust Java Code](#) – Scott Ambler

[Sun Coding Conventions](#): The company that has written Java and the Java class libraries has established a standard for coding style and conventions.

[XHTML and CSS Tutorial](#): Great beginner tutorial from www.html.net suggested by w3c. After reading, progress to use Macon State tutorial

[XHTML and CSS Tutorial](#): Superb Macon State tutorial with hot links to all needed references.

[JUnit](#): Java Unit Testing Tool tutorial. The software comes installed with Eclipse. We will use version 4 instead of 3.8 this year.

[User Interface Design for Programming](#): Joel is a well noted author on software development

Course Goals

1. Expand the material from CSC 230 to broaden the students' understanding of graphical user interface development design practices and advanced features.

CSC Course Document and Assessment Template – updated to be superset of ABET
7/24/07

2. Develop a solid foundational understanding of relational database design, query and interface with the java programming language.
3. Introduce concepts of multi-threading and interprocess communication.
4. Understand java client server programming for a two tiered architecture.

Prerequisites by Topic

CSC 230

Major Topics Covered in the Course

- PF5. Event-driven programming [core – 2 of 4 hours]
- NC1. Introduction to net-centric computing [core – 2 of 2 hours]
- NC4. The web as an example of client-server computing [core – 3 of 3 hours]
- NC5. Building web applications [advanced – 4 hours]
- PL6. Object-oriented programming [core – 1 of 10 hours]
- HC1. Foundations of human-computer interaction [core – 2 of 6 hours]
- HC2. Building a simple graphical user interface [core – 2 of 2 hours]
- HC3. Human-centered software evaluation [1 hour]
- HC4. Human-centered software development [1 hour]
- HC5. Graphical user-interface design [1.5 hour]
- GV1. Fundamental techniques in graphics [core – 2 of 2 hours]
- IM2. Database systems [core – 3 of 3 hours]
- IM3. Data modeling [core – 4 of 4 hours]
- IM4. Relational databases
- IM5. Database query languages [3 hours]
- IM6. Relational database design [1.5 hours]
- IM7. Transactional processing [1.5 hours]
- IM12. Hypertext and hypermedia [3 hours]
- OS3. Concurrency [core – 3 of 6 hours]
- SP2. Social context of computing [core – 1.5 of 3 hours]

Learning Objective/Outcomes with Cross Reference to CSC Program Outcomes

- Become familiar with the latest version of Eclipse Interactive Development Environment (IDE) and Java SDK 6.0 for developing standalone and client server applications. (PO 8, 11)
- Understand the Human/Computer Interaction design principles and best practice conventions for graphical user interfaces. (PO 5, 8, 11)
- Understand and build advanced graphical user interfaces using basic Swing components and advanced model based components such as JTable. (PO 8, 9)
- Manage components using advanced layout techniques. (PO 5, 8)
- Understand how to construct international applications. (PO 4, 9)
- Use a graphical layout tool for designing and developing a graphical user interface. (PO 11)
- Understand multithreading concepts and build multithreaded application programs. (PO 5)
- Understand and build client browser displays using HTML with an Apache HTTP server. Understand basic networking concepts using sockets.
- Work with files and I/O using serialization. (PO 5)
- Work with XML files and I/O using DOM. (PO 5)
- Understand how to design a relational database in third normal form. (PO 5, 11)
- Understand how to use ANSI SQL to define, query and modify relational database tables.
- Understand the JDBC API for connectivity between an application and relational databases. (PO 5, 9)
- Understand how to use a database in embedded and server mode for a single standalone application, a two tier application and a three tiered application. (PO 5, 8, 11)
- Gain familiarity with pairwise programming and working in teams (PO 2)

Laboratory projects (each project is approximately 1 week)

1. Xhtml/CSS (HW 1)
2. GUI Simple Components (Menus, buttons, text area, text buttons, labels) (HW 2 and 3)
3. GUI Internationalization (resource bundles, date, time and currency) (HW 6)
4. GUI Advanced MVC components (Table, SplitPane, TabbedPane) – (HW 4 and 5)
5. Concurrency (SwingWorker) (HW 7)
6. Networking: sockets (HW 7)
7. Database normalization and query (HW 9)
8. Database connectivity using JDBC (HW 9)
9. Distribution: jar files, applets (policy), web start (jar signing) (HW 8)

Estimate Curriculum Category Content in semester hours. Maximum number of semester hours per course is 4. Times should be in increments of .5. (Advanced is a topic requiring previous core materials – most likely found in a 300 or 400 level class.)

Category	Core	Advanced
Data Structures		
Algorithms		
Software Design		.5
Computer Architecture		
Programming Languages		3.5

Oral and Written Communications

Every student is required to submit at least __0__ written reports (not including exams, tests, quizzes, or commented programs) of typically __0__ pages and to make __0__ oral presentations of typically __0__ minute’s duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

Social and Ethical Issues

Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

Software internationalization – Understand the GUI principles for developing software to support international end users and international development. This will consider GUI labeling, log messages, currency, time, date and number representational differences. Approximately two class periods will be spent discussing internationalization and implementing a GUI to support international framework. We will use Google translation for translation issues.

Theoretical Content

Please list the types of theoretical material covered, and estimate the time devoted to such coverage.

1. Model View Controller 3 hours
2. Database table normalization to 1, 2nd and 3rd normal forms (2 hours)
3. Polymorphism (1.5 hours)
4. Concurrency: deadlock, mutual exclusion (3 hours)

Problem Analysis

Students will have 9 homework assignments. Each homework is a standalone assignment. They are given no code but rather need to design and implement a solution to solve the problem. However, the individual homework assignments are limited in scope and are designed to primarily involve a problem that can be solved by the particular topic under discussion. The nine homework assignments will focus on the following problem areas:

1. XHTML and CSS (separation of structure and format)
2. Swing GUI Simple components and internationalization with resource bundles
3. Swing advanced layouts and MVC components
4. Concurrency
5. Network sockets for interprocess communication
6. Deployment security
7. Database design and query
8. Database connection using JDBC
9. XML

Solution Design

1. Design patterns: MVC, Proxy
2. Separation of structure from presentation (xhtml 1.0/CSS)
3. GUI Internationalization
4. Standalone, Client server (two tier)
5. Multithreaded
6. Deployment (Applet, WebStart, Jar)
7. Separation of structure from presentation (xhtml 1.0/CSS)

Course Assessment

Proposed changes from last offering (***Pre assessment – August 1, 2008***):

This is the second offering of the course at Elon. I will implement all of the proposed changes recommended by the post assessment of the course's first offering. The changes are:

1. Three more sessions will be spent on databases. The additional sessions will cover topics on advanced SQL queries, Object Relational Mapping using IBATIS and using Data Access Objects as a design pattern for the persistent storage layer.
2. Remote Method Invocation using the Java RMI protocol will not be covered. It is covered in CSC 462.

3. Servlets and JSP will not be covered.

Learning outcomes to be assessed, how they will be assessed and success criteria (**Pre assessment**)

Two learning outcomes have been selected that tie in with the department program outcome 9 that is being assessed during AY 08-09. Program Outcome 9 is “Students will be able to solve problems using procedural, functional, and object oriented programming paradigms”. The two learning outcomes for the course that relate to this program outcome are:

- Understand how to construct international applications. (PO 4, 9)
- Understand the JDBC API for connectivity between an application and relational databases. (PO 5, 9)

The design and development of international applications using sun provided object oriented java library calls will be assessed by a dedicated homework (homework 6) on development of a GUI that supports at least two different languages. Success will be an average homework grade of 70 from the class.

The understanding of how to use and interact with a underlying database using the Sun JDBC object library will be assessed by a dedicated homework (homework 9) on the design and development of a java application with a GUI front end and a relational database backend. The interaction between the two will be using JDBC. Success will be an average homework grade of 70 from the class.

Assessment data and analysis (Post assessment - January 2009)

Program outcome 9: Internationalization was assessed by homework 6 and the students developed a full internationalized GUI for a Password Manager. The GUI by default came up in English and had to have a menu option to switch to a different language. When the language was selected then all aspects of the GUI had to be switched to that language (e.g. accelerators, short cuts, tabs, file choosers, labels, and table headings). A variety of languages were selected by the students to include French, Spanish, Dutch, Italian and Greek. The average score on the assignment was 88.60 that easily surpassed the goal of 70. Figure 1 shows the grade distribution and 90% of the students (nine of ten) surpassed the goal score.

Grade Distribution	
greater than 100	0
90 - 100	7
80 - 89	1
70 - 79	1
60 - 69	0
50 - 59	0
40 - 49	1
30 - 39	0
20 - 29	0
10 - 19	0
0 - 9	0
less than 0	0

Figure 1: Internationalization Homework Averages

Understanding the JDBC api was assessed by homework 9 where the student had to design, normalize and implement a relational database of recent movies in Derby and connect to it from a

CSC Course Document and Assessment Template – updated to be superset of ABET
7/24/07

GUI front end. The average score on the assignment was 91.80 easily exceeding the goal of 70. Figure 2 shows the distribution of grades and all ten of ten students exceeded the goal of 70. This was the only required pairwise team assignment and working as a team provided additional motivation to the student who did not do well on Homework 6. Based on the performance on Homework 6 and Homework 9, Program outcome 9 is clearly being met.

Proposed changes for next offering (Post assessment – January 2009)

I am very pleased with the class. The extra class on SQL proved valuable and should be kept. The ORM class on IBATIS proved to be too much. The class immediately follows a discussion of DAO and the true value of a tool gets lost when working with the small tables used in a class session. In addition, there are many competing ORM tools and selecting one is not representative of the field. A better use of the one class period would be to discuss MySQL or another database and have the students port an application to the database.