

**UNIVERSITY OF NEBRASKA AT OMAHA  
COURSE SYLLABUS/DESCRIPTION**

Department and Course Number	CSCI 3830
Course Title	Advanced Java Programming
Course Coordinator	Raj Dasgupta
Total Credits	3
Date of Last Revision	November 4, 2008

**1.0 COURSE DESCRIPTION**

- 1.1 Overview of content and purpose of the course  
This course teaches students advanced computing techniques and tools using the Java language. Topics include distributed programming, JavaBeans; advanced graphical user interfaces; Java Database Connection; and security.
- 1.2 For whom course is intended  
Computer Science, Computer Engineering, Information Systems and Quantitative Analysis majors. This course will serve as an elective for the Bachelor of Science degree in Computer Science
- 1.3 Prerequisites
  - 1.3.1 CSCI 2830: Java Programming (prior to Fall 2008)  
OR  
CSCI 1620: Computer Science Programming II (from Fall 2008)
  - 1.3.2 CSCI 3320: Data Structures (can be taken as a co-requisite)
- 1.4 Unusual circumstances of the course  
None

**2.0 OBJECTIVES**

- 2.1 List the performance objectives of the course
  - 2.1.1 Familiarize students familiar with different programming techniques such as client-server programming, distributed programming.
  - 2.1.2 Focus on the different programming models required to design software systems that use the techniques mentioned in 2.1.1
  - 2.1.3 Encourage students to develop new protocols and new interaction models for Web-based computing
  - 2.1.4 Introduce the student to the literature in Web-based computing using Java.
  - 2.1.5 To stress the importance of distributed computing on the Web as opposed to stand-alone desktop programming.
- 2.2 The students will:
  - 2.2.1 Apply principles of distributed Web-based computing to develop distributed applications in Java.
  - 2.2.2 Design and test distributed applications in Java using the techniques taught in class.

- 2.2.3 Conduct evaluations (usability tests) of the programs that they themselves have written.
- 2.2.4 Produce user documentation for all of the programs.

### **3.0 CONTENT AND ORGANIZATION**

#### List of topics to be covered in chronological sequence

Topics marked with an asterisk(\*) will be covered selectively depending on class progress and time available

- 3.1 Course Introduction and review of Basic Java (1.5 hours)
- 3.2 Advanced Swing (5 hours)
  - 3.2.1 Lists
  - 3.2.2 Trees
  - 3.2.3 Tables
  - 3.2.4 Sliders and Progress Meters
  - 3.2.5 Tool Bars and Tool Tips
  - 3.2.6 Component Organizers (Split Panes, Tabbed Panes, Desktop Panes and Internal Frames)
- 3.3 Java Beans (5 hours)
  - 3.3.1 The Bean-Writing Process. The BDK and the BeanBox
  - 3.3.2 Naming Patterns for Bean Properties and Events
  - 3.3.3 Bean Property Types
  - 3.3.4 BeanInfo Classes
  - 3.3.5 Property Editors
  - 3.3.6 Customizers\*
  - 3.3.7 JavaBeans Persistence\*
- 3.4 TCP/IP Socket Based Networking (5 hours)
  - 3.4.1 Creating a Client-Server Application, Creating multiple clients
  - 3.4.2 Making URL Connections, Handling Forms
  - 3.4.3 Advanced Socket Programming
- 3.5 Remote Objects, Remote Method Invocation and CORBA (7.5 hours)
  - 3.5.1 Stubs, skeletons and parameter marshalling
  - 3.5.2 Setup for RMI
  - 3.5.3 Parameter Passing in Remote Methods,
  - 3.5.4 Server Object Activation
  - 3.5.5 Java IDL and CORBA
  - 3.5.6 Remote Method Calls with SOAP
  - 3.5.7 JINI
  - 3.5.8 Javaspaces
- 3.6 Java Database Connectivity (7.5 hours)

- 3.6.1 Introduction and Basic JDBC Programming
  - 3.6.2 Advanced SQL Querying
  - 3.6.3 Scrollable and Updatable Result Sets
  - 3.6.4 Metadata
  - 3.6.5 Row Sets
  - 3.6.6 Transactions
  - 3.6.7 Advanced Connection Management\*
  - 3.6.8 Introduction to LDAP\*
- 3.7 Internet Security (5 hours)
- 3.7.1 Class Loaders
  - 3.7.2 Bytecode Verification
  - 3.7.3 Security Managers and Permissions
  - 3.7.4 Digital Signatures
  - 3.7.5 Authentication
  - 3.7.6 Encryption

#### **4.0 TEACHING METHODOLOGY**

- 4.1 Methods to be used  
Teaching methods will include in-class lectures, hands-on lab exercises, in-class quizzes, homework assignments, case studies, demonstrations, and self-directed study (using materials distributed via the class web site, reference material distributed in class).
- 4.2 Student role in the course  
Students are expected to attend all lectures and labs, participate in class discussions, complete assigned homework, group project(s), and examinations.

#### **5.0 EVALUATION**

- 5.1 Types of student products
- 5.1.1 Develop and implement a stand-alone Java application for a meeting scheduler system that has user-interface components designed using the Advanced Java Swing GUI.
  - 5.1.2 Implement the application as a Java Bean that allows end users to drag and drop components and build their own application.
  - 5.1.3 Extend the stand-alone application mentioned above to a distributed system in which multiple applications can communicate with each other using the client-server model over a network.
  - 5.1.4 Implement the distributed system mentioned above, using middleware programming such as RMI and CORBA as the communication technique over the Internet.
  - 5.1.5 Implement the distributed application using Java applets as the communication model.
  - 5.1.6 Implement the distributed application using mobile agents as the communication model.

5.1.7 Interface the front-end of the distributed application to a database at the back-end using JDBC

5.2 Basis for determining the final grade

In class quizzes: 10%

Homework assignment: 40%

Final Project Software Demo: 30%

Final Project Report and Oral Presentation: 20%

5.3 Grading scale

<b>Percent</b>	<b>Grade</b>	<b>Percent</b>	<b>Grade</b>
97 – 100	A+	77 – 79	C+
94 – 96	A	70 – 76	C
90 – 93	A–	70 – 73	C–
87 – 89	B+	67 – 69	D+
84 – 86	B	64 – 66	D
80 – 83	B–	60 – 63	D–

Below 60: F

## 6.0 RESOURCE MATERIAL

6.1 Textbook(s) or other required readings used in the course, or equivalent examples

6.1.1 Required Text:

- Horstmann, C., and Cornell, G., "Core Java, Vol. II - Advanced Features," (Eighth Edition) Prentice Hall, 2008.

6.1.2 Additional texts:

- Deitel, H., Deitel, P., and Santry, S., "Advanced Java 2 Platform: How to Program," Prentice Hall, 2001.
- Horstmann, C., and Cornell, G., "Core Java, Vol. I, Fundamentals," Prentice Hall, 2008.

6.2 Other sources for gathering information or equivalent

Students are directed in class to search the World Wide Web for relevant articles or case studies.

6.3 Current bibliography or equivalent

(See 7.1 and 7.2)

**7.0 Computer Science Accreditation Board (CSAB) Category Content (class time in hours):**

<i>CSAB Category</i>	<i>Core</i>	<i>Advanced</i>
Data structures		3
Computer organization and architecture		2
Algorithms and software design		15
Concepts of programming languages	3	22

**8.0 Oral and Written Communications:**

Every student is required to submit at least \_\_\_1\_\_ written reports (not including exams, tests, quizzes, or commented programs) to typically \_50\_\_\_ pages and to make \_\_1\_\_ oral presentations of typically \_\_30\_\_ minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

**9.0 Social and Ethical Issues:**

None

**10.0 Theoretical content:**

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

	Contact hours
1. Model-View-Controller Architecture	2.0
2. JavaBeans Architecture	3.0
3. Architecture of the different networking techniques outlined in sections 3.4 and 3.5 above	5.0
4. Agent oriented programming	2.0
5. Database design primer - ER diagrams, Relational databases and schema	2.5
6. Computer security techniques: cryptography, digital signatures, public key infrastructures	1.0

**11.0 Problem analysis:**

Students will be required to analyze the problems discussed in class along with the instructor.

Problems given in homework assignments should be analyzed by the student.

Problem analysis comprise the following steps:

1. Determine the most appropriate data structures for the problem
2. Determine and modify the Java based technologies taught in class to suit the problem needs

## 12.0 Solution design:

Please describe the design experiences common to all course sections. Students taking the course are expected to develop the software design for all the problems given as part of the homework assignments, quizzes and examinations. The design involves the implementation of software objects to implement the different technologies the students use to solve the problems, and the interaction method between those objects.

### CHANGE HISTORY

<i>Date</i>	<i>Change</i>	<i>By whom</i>	<i>Comments</i>
03/10/2002	Original Version	Craiger	
03/15/2002	Added sections on course description, objectives, content and organization, teaching methodologies, evaluation and resources/textbooks.	Dasgupta	
10/15/2002	Added time spent on each topic in content and organization. Refined grading scale. Removed section titled "Attached documentation" (was not there in the ABET sample sent by mchennuri). Added sections 7-12.	Dasgupta	
06/13/2003	Cleanup presentation	Wileman	
10/27/2008	Checked syllabus currency	Dasgupta	No changes made to syllabus, added mapping CS outcomes table below
11/04/2008	Revised syllabus,	Dasgupta	Removed java agents, updated socket programming, RMI, JDBC and security topics