DeVRY UNIVERSITY

Course Syllabus **Spring 2007**

Course Title:	Distributed **Computing Using Java With Lab**
Course number:	CET431
Credit/Contact hour:	4-2-5
Course Dependency:	**Prerequisite:** CET375, **EET301** **Required Co-requisite**: (none)
Instructor:	Tom Wheeler E-mail: twheeler@kc.devry.edu Telephone: 816.941.0430 x5211
Office Hours:	**Monday 11-11:50 am; Thursday 12-12:50pm; Friday 12-12:50pm** Other times possible by appointment.

Course Description

This course focuses on application of software principles involved in a distributed computing environment. Design issues in transport and session layers protocols, terminal and file transfer protocols, message handling protocols, flow control and buffering, network security methods and models of distributed computation are covered. Lab work introduces distributed programming applications requiring development and debugging of computer programs.

Textbooks and Materials

Textbook/s Title:	**Ed**	**Author**
„*JAVA How to Program* (Prentice-Hall)	**CURR**	Deitel & Deitel

Software

NetBeans 3.5.1 (Sun Microsystems) or equivalent integrated development environment

OR

Eclipse 3.2.0 (Open Source @ www.eclipse.org)

- This software is available free of charge from Sun Microsystems at `http://java.sun.com`.
- The course is taught using NetBeans, primarily. However, Eclipse has been tested and will also work just as well.

Terminal Course Objectives (TCOs):

Following are the objectives for this course. Individual faculty, based upon their experience and expertise may add to these objectives to meet local campus needs. Any such additions will be communicated to the class. While the instruction remains focused in helping students, accomplishing these objectives is a shared responsibility of students and faculty. The outcomes of this course will depend upon the motivation and capabilities of the students, sufficient time allocation for studying, and the effectiveness of that effort.

DeVry University is committed to the continual improvement of its curriculum and instruction and to meet the needs of students and employers in a rapidly changing global economy. Students, faculty, and the university must all be actively involved to accomplish these objectives, as well as the objectives of this particular course.

Every class is to some extent a unique interactive experience, which may cause some variance within the stated objectives, in either content or level. Individual faculty, based on their experience and expertise, are encouraged to add objectives, as they deem appropriate, and to communicate these directly to the class. The outcomes of the course will depend on the design of the course, the quality of instruction, and the motivation and capabilities of the students, including time available for studying and the effectiveness of the effort.

1...Given a distributed system such as the Internet or a large network of computers, explain its operation.
2...Given a simple application that uses a Java Applet such as memo-pad, design the application.
3...Given a simple application that uses a Java Applet such as one that displays the contents of a database by a few criterion, design the application.
4...Given a design concept such as a game of solitaire, design the objects needed to implement the game.
5...Given a client-server application such as the transfer of a simple message through the Internet, design the software required to reliably transfer the message.
6...Given a distributed computing application such as the game of battleship, implement the client-server relationship.
7...Given a distributed system that uses a computer and an embedded appliance such as a weather station develop a system that logs data using the embedded system as a client to the computer system that acts as a server to monitor up to 1,000 such weather stations appliances.

How this Course helps in Achieving Your Program's Objectives:

The following matrix illustrates how this course supports achievement of your Program Objectives.

CET431	Assessment Outcomes																							
Lecture	*Testing* & Instrumentation					*Create/Implement Programming*				*Hardware/Software Design, Implem, & Eval*						*Writing Speaking*		*Team*		*Research Prob. Solv*		*Tech./Society Linkages*		
TCOs	1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	3.1	3.2	3.3	3.4	3.5	3.6	4.1	4.2	5.1	5.2	6.1	6.2	7.1	7.2	7.3
1											x													
2											x													
3											x													
4											x													
5	x																		x					
6						x													x					
7							x																	

EET/CET Program Objectives and Outcomes:

(Student competencies achieved at the time of graduation)

1. Conduct experiments involving electronic systems using modern test equipment, interpret test results and use them to improve products or methodologies.
   1.1. Performs Needs Analysis – define the problem
   1.2. States goals and objectives of the experiment
   1.3. Identifies resources to conduct experiment (parts, equipment, data sheets, etc.)
   1.4. Develops a procedure and collect data using modern test equipment
   1.5. Analyzes test results and draw conclusions.

2. Create, implement high-level and Assembly language programs in support of technical activities.
   2.1. Analyzes the problem logically
   2.2. Designs the solution
   2.3. Implements the solution
   2.4. Tests and debugs the software

3. For EET: Use the principles of science, mathematics, and engineering technology to design, implement, and evaluate hardware and software solutions to complex technical problems,
   3.1. Selects and defines a meaningful problem taking safety, ethical, social, economic, and technical constraints into consideration.
   3.2. Devises process to solve problem
   3.3. Applies appropriate knowledge of scientific, mathematical, and engineering design tools toward the design and analysis of problem solutions.
   3.4. Identifies key issues in designing and building a prototype
   3.5. Builds, tests and troubleshoots prototype
   3.6. Optimizes prototype with a commitment to quality, timeliness, and continuous improvement.

For CET: Use the principles of science, mathematics, software engineering, and engineering technology to design, implement, and evaluate software solutions to complex technical problems.
   3.1 Identifies a meaningful problem and defines preliminary solution specifications taking safety, ethical, social, economic, technical constraints, and user requirement into consideration
   3.2 Designs and implements appropriate data structures and algorithms
   3.3 Prepares a plan of action to implement the system
   3.4 Applies scientific, mathematical, software, and engineering design tools toward the design and analysis of problem solution
   3.5 Writes and tests readable and maintainable code
   3.6 Optimizes code with a commitment to quality, timeliness, and continuous improvement

4. Communicate effectively both orally and in writing.
   4.1 Communicates effectively in wiring
   4.2 Communicates effectively orally

5. Work effectively in a team environment.
   5.1 Exhibits good dialoguing skills
   5.2 As part of a small group project, when assigned roles, performs roles effectively

6. Apply applied research and problem-solving skills to support learning at DeVry as well as life-long personal and professional development.
   6.1 Recognizes the need to know information beyond one's own expertise and has the ability to gather and synthesize the necessary information into the solution of a problem
   6.2 Uses engineering problem-solving methodology in solving problems

7. Evaluate the broader effects of technology and to identify connections between technology and economics, politics, culture, ethical responsibility, social structure, the environment and other areas.
   7.1 Identifies linkages and causal relationships between technology and social, political, economic, cultural, and environmental conditions.
   7.2 Works effectively in diverse environments and adapts technical solution to solution a diverse audience
   7.3 Pursues technical work within guidelines for professional, ethical, and social responsibility

Class Policies and Procedures:

Attendance

Each student is required to attend every lecture and laboratory session in which he or she is enrolled. A swipe-card terminal (ATS) in each classroom is used to record attendance electronically. Students are responsible for arriving before class begins, sliding their identification card through the wall-mounted reader, and remaining for the duration of the course meeting. Students who are absent for two or more days should notify their Professor or assigned Academic Advisor in advance. Students who miss more than five (5) consecutive days of school are in violation of the DeVry attendance policy and will be dismissed. ***Unexcused absence in excess of 5 hours will result in a reduction of final grade by one letter***.

Homework

Homework is due at the beginning of class (xx00 UTC). ***Late homework is not accepted***. Homework carries the weight of one major exam (100 points) in the course. Failure to do homework will do severe damage to your grade. (UTC=Universal Coordinated Time, or Standard World Time.)

Homework will be kept in a 3-tab folder, with the latest assignment in front. Your name, the course number, and the instructor's name must appear in clearly-readable form on the front of the folder.

Make-Up Exams

No make up exams are given in CET431 for any reason.

Course Grading Standards

The final grade in CET431 is based on the following items:

- The grades from the three course projects (100 points each, 18% of final grade for each)
- A comprehensive final examination (150 points, 27% of final grade)
- Quizzes and homeworks (50 points total, 10% of final grade)
- Participation and Attendance (50 points total, 9% of final grade)[1].

  Expectations regarding participation and attendance: DeVry University requires that each student attend every scheduled class meeting. Absence for more than 4 class hours will cause a zero (0) to be assigned for this portion of your grade. Participation means active, productive engagement in the classroom activities, including discussion, programming, and other activities.

  Come to class prepared to discuss and apply the material!

Project Due Dates: Projects are due on or before the specified deadlines below. You and/or your group are responsible for keeping the instructor informed of progress as well as any problems that may arise so that corrective action can be taken.

Project 1: Due Friday, Week 5 (Individual effort; not a group project)
Project 2: Proposal (with group roster) due Friday, Week 5.
Completed project due Friday, Week 9
Project 3: Proposal for special project #3 variations due Friday Week 9.
Final version of Project 3 Due Friday, Week 14

Late Projects:

Projects will be accepted up to one week late, with a grade reduction of 5 points per day (25 points maximum). After one week, no project will be accepted without prior approval of the instructor.

Note: The first Java project is done by ***individuals****, not by teams. Each student must work on the first project individually and submit their own and code and documentation for grading. Projects two through four are team projects.*

[1] These values are rounded to the nearest percent to allow eCollege to accurately weight them; the sum is 100%.

A final letter grade is to be awarded to each enrolled student in accordance with the 4.00 grading system:

Letter Grade	***Percent of Total Points***	***Grade Points***
A	90 – 100%	4.00
B	80 – 89%	3.00
C	70 – 79%	2.00
D	60 – 69%	1.00
F	Below 60%	0.00

Documentation Style Guide for Written Reports

The documentation you'll be writing for the projects in CET431 will be designed for *end-users*. This is probably different than other "lab reports" you've written for technical courses.

Documentation for end-users is generally designed to help people understand and use the software. What it *doesn't* do (usually) is provide internal operating details, flow charts, source code, or any other information of this type.

Usually, an end-user's manual is organized as follows:

- Cover Page
- Table of Contents
- Chapter or Section-Oriented Material, which typically includes sections such as:

  a) Installation Procedure
  b) Start-up Procedure
  c) Operating Procedure (with troubleshooting information as needed)

  -- Operating procedures should walk the user through each major feature of the program, providing screen shots and other relevant information as needed.

- Appendices (where needed). For example, if you feel that a glossary would be a useful addition, it would be placed in an appendix.

The document "WAVEMAGIC.PDF" (located at http://analogmagic.com/wavemagic/wavemagic.pdf) is one example of how you may format end-user documentation. (There are almost an infinite number of ways that you may format your own writing.) No matter how you write it, the following are important points to remember about the user's manual:

- It should have an attractive appearance.
- It should be well organized and easy to navigate.
- It should cover complete operation of the product.
- It should be easy to understand and free of serious grammatical mistakes that could obscure its meaning.

Evaluation Rubric For Projects:

- 60 Points - Program works correctly, according to either the specifications from the lab syllabus, or the student-provided proposal.
- 40 Points - Documentation Package
- A formal evaluation rubric is given under DocSharing; please attach this to each laboratory report.

Further information about the evaluation of written work is provided on the next page.

The A paper consists of the following:

Central Idea:

- Is clearly expressed, responds to the assignment, provides focus.
- Is explicitly and logically supported with concrete details and examples.

Structure:

- A plan of organization is given in which ideas are arranged in a clear, logical order.
- Ideas are clearly connected.

Development:

- Generalizations are supported or explained with concrete details.
- Smooth transitions are used between sentences and paragraphs.

Style:

- Varied sentence length and structure.
- Consistent and appropriate tone.

Mechanics:

- Grammar, punctuation, capitalization, spelling are correct.

A B paper consists of the following:

Central Idea:

- Is clearly expressed, responds to the assignment, provides focus.
- Is explicitly and logically supported with concrete details and examples.

Development:

- Concrete details usually given to support ideas.
- Transitions are given in most instances where needed.

Style:

- Contains some variation of sentence length and structure.
- Tone is consistent throughout.

Mechanics:

- No more than eight mechanical errors.

A C paper consists of the following:

Central Idea:

- May be slightly askew, but seems to be somewhat clear.

Structure:

- A clear construction is attempted, but does not measure up consistently, and ideas are usually connected via transitions.

Development:

- Writer has attempted to give enough information to support his/her ideas, but there are "holes" where the reader may be uncertain.

Style:

- The writer has attempted a few times to vary sentence length and structure, and tone shifts often.

Mechanics:

- No more than 10 mechanical errors.

A D paper consists of the following:

Central Idea:

- Is somewhat unclear, but is stated.

Structure:

- The ideas are somewhat "rambling" in nature, and few transitions are given.

Development:

- Many ideas have little concrete information for their support. Thus, they often fade into mere opinion rather than rather than expressing "facts."
- Few transitions are given.

Style:

- Leaves the reader feeling unsure of the writer's own attitude toward the topic.

Mechanics:

- Has more than 15 grammar, punctuation, spelling errors.

An F paper consists of the following:

Central idea is missing, and writing wanders from topic to topic without a clear focus.

Structure -- no clear structure -- becomes a jumble of ideas without a stated reason given for why it was written.

Development -- very little development or support given for any discernable ideas.

Mechanics -- writer evidences very little basic understanding of grammar, punctuation, or spelling skills. Many errors of each kind.

Academic Integrity Policy

Ideas and learning form the core of the academic community. In all centers of education, learning is valued and honored. No learning community can thrive if its members counterfeit their achievement and seek to establish an unfair advantage over their fellow students. The academic standards at DeVry are based on a pursuit of knowledge and assume a high level of integrity in every one of its members. When this trust is violated, the academic community suffers injury and must act to ensure that its standards remain meaningful. The vehicle for this action is the Academic Integrity Policy outlined in the *Student Handbook*.

The Academic Integrity Policy is designed to foster a fair and impartial set of standards upon which academic dishonesty will be judged. All students are required to read, understand, and adhere to these standards, which define and specify the following mandatory sanctions for such dishonest acts as copying, plagiarism, lying, unauthorized collaboration, alteration of records, bribery, and misrepresentation for the purpose of enhancing one's academic standing:

- The ***first recorded offense*** will result in the student receiving zero credit for the entire paper, exam, quiz, lab, homework assignment, or other graded activity in which the incident of academic dishonesty occurred. No partial credit may be given. Where the incident involved a graded assignment normally subject to a "drop" option, the student may not exercise that option.

- The ***second recorded offense*** will result in the student receiving a failing grade for the course in which the second offense occurs. The second offense need not be in the same course, program, or term as the first offense to invoke this sanction.

- The ***third recorded offense*** will result in the student being permanently expelled from the DeVry system. Again, the third offense need not be in the same course, program, or term as either the first or second offense to invoke the sanction.

All assignments you turn in for this course (coding, homework, laboratory, or other assignments) are expected to be your own original efforts. Do not share your work with other students. If you quote from an outside source, you must properly acknowledge the source of the information using MLA or APA style. The following sites have examples of proper usage for these styles:

```
http://owl.english.purdue.edu/owl/resource/557/01/
http://owl.english.purdue.edu/owl/resource/560/01/
http://www.apastyle.org
```

Changes to Syllabus:

The contents of this syllabus are subject to change with appropriate notice to the students.

Weekly Course Schedule:

Week	Topics	TCO's	Homeworks
1	Java Intro, IDE, JAR mgmt, Fundamental data types -vs- objects	2,3	
2	Data and references; instantiation; static/instance variables, methods; access control; constructors, finalizers, garbage collector; document-view model	2,3	
3	Streams; formatting; subclasses; super()	2,3	
4	Arrays; arrays & functions; interfaces; card deck simulation problem	2,3,4	
5	Files and object serialization; Introduction to GUI design	2,3	HW#1 due
6	GUI Elements & Controls: JPanel, JFrame, JButton; listeners; content panes, layout managers	2,3	
7	Forms editor; GUI events w/form editor; anonymous classes; drawing in a window during paint(); mouse events (Java Sketch problem)	2,3	
8	Multimedia: Image, MediaTracker; GUI implementation of card simulation	1,5	HW#2 due
9	Multithreading (rabbitthread); Communications in multithreaded programs using interfaces; Transport system (TCP, UDP) properties; networking with Java Sockets;	1,5,7	
10	UDP/TCP serialization; Instant Messenger problem	1,5,7	
11	Theory & Design of distributed protocols	5,6	
12	UNIX / text-based protocol design and implementation	4,5,6	
13	Design of client & server applications	4,5,6,7	HW#3 due
14	Java applets, wrap-up	1-7	

This schedule is tentative and subject to change based on class group response and performance.