CPS211 - OBJECT-ORIENTED SOFTWARE DEVELOPMENT

Professor:	Russell C. Bjork russell.bjork@gordon.edu	Fall Semester, 2009 MWF 1-2 pm
Office:	MacDonald 217 x4377	Blackboard site plus http://www.cps.gordon.edu/ courses/cps211
Hours:	MF 2:10-4:00 PM; Tu 2:30-4:30 PM and by appointment	Lab: Tu 9:45-12:45 (beginning 9/1)

PREREQUISITE: Computer Science 112

CATALOG DESCRIPTION:

Introduces object-oriented analysis and design, including use cases, CRC cards, and class design; UML; inheritance and polymorphism; methods for verifying, validating, and testing software; concurrency; distributed systems; relational databases; continued development of design and programming skills using UML and Java through weekly laboratories.

COURSE OBJECTIVES:

In general, this course is intended to set programming in the broader context of software development, which includes requirements analysis, design, verification and validation, and maintenance as well. In particular, upon completion of this course, you should be able to:

1. Analyze a moderately complex problem and design a solution to it, using UML notation.
2. Implement and test such a design, using Java.
3. Carry out the process of designing, implementing, and testing a piece of software as a member of a small team.
4. Apply principles of good user interface design.
5. Create simple pieces of software that utilize concurrency, relational database access.
6. Use simple design patterns

TEXT: Carol Britton and Jill Doake, *A Student Guide to Object-Oriented Development* (Oxford: Elsevier, 2005)

COURSE TECHNIQUES AND PROCEDURES

Since this course is primarily concerned with the development of certain skills and habits, regular practice with evaluation will be the heart of the course. For each unit of material, you will be asked to read a portion of the text book, to do the short answer questions at the end of the chapter, and to do assigned homework problems.

Class sessions will include a discussion and amplification of the material in the text and the presentation of further examples and supplementary material. You should not expect to grasp everything presented in the text when you first read it; however, you should note areas that are unclear to you and be prepared to raise questions about them in class.

As is true with any skill, the only way you can really learn software development is by doing it. For this reason, you will have opportunities to practice what you are learning by doing homework problems and through weekly laboratories. Additional practice will come from an ongoing design and programming project which you will work on as part of a small team of students, with various portions of the project being due throughout the term. You should look on this as being your key learning experience in the course.

COURSE REQUIREMENTS AND EVALUATION:

1. You will be expected to read most of the textbook, as assigned in the schedule below. (Reading assignments should be completed BEFORE the class hour in which the topic is discussed, as specified in the schedule below.) However, our classroom discussion will not rigidly follow the order of material in the text, nor will it be confined to material covered there.

2. Each chapter ends with a set of "Quick check questions" that are designed to be done when you read the chapter. You should answer them **in writing** as part of you reading. (Occasionally, I will distribute additional questions for you to answer). In most cases, I will not collect your answers, but we will talk about these as part of our consideration of the chapter in class, and you may see questions similar to them on the exams.

   On days when textbook reading is due, I will pass around an honor-system sheet for you to indicate whether and to what extent you answered the quick check questions (and any others I distributed) in writing. I don't expect you to necessarily get the "right answer"; all I'm looking for is a good-faith effort. At the end of the semester, 5% of your final grade will be based on your faithfulness in having answered these questions when the reading is due.

   (If the reading of a textbook chapter is broken up into multiple assignments, the schedule will indicate which quick check questions go with each portion of the book assigned. If there is no explicit statement in the schedule, you should do all the quick check questions.)

3. Most chapters in the book include a set of Exercises at the end. Though these will not be formally assigned, we will do a number of them in class sessions. For this reason, you should be sure to bring your book with you to class.

4. Weekly laboratories will focus on gaining practical experience with the material covered in the book and/or in lecture. Lab assignments will be given out the day before lab, and **must** be read over carefully **before** coming to lab. For most laboratories, there will be a writeup to turn in. There may also be a quiz given at the start of the lab hour (based on your reading of the lab assignment) and/or a quiz based on the work done in lab given at the start of class on the due date. Each laboratory writeup and quiz(zes) will be worth 3% of the final course grade (30% total for the 10 labs with formal requirements).

   The following are the tentative emphases for the lab sessions *(subject to change)*:

Lab	**Emphasis**
1	Using an Integrated Development Environment (IDE)
2	(part II)
3	Using a Debugger
4	Using UML Modeling Tools
5	Java Collections
6	Implementing a UML Design
7	(part II)
8	(No formal requirements - time available to work on term project)
9	Concurrency; Threads
10	(No formal requirements - time available to work on term project)
11	Relational Databases; SQL
12	Accessing a Database from a Program (JDBC)
13	(No formal requirements - time available to work on term project)
14	(No formal requirements - time available to work on term project)

5. Throughout the term, you will work on an ongoing design and programming project as part of a team of two students. Portions of this project will be due at different times throughout the term, as shown in the course schedule below, and each portion will be graded when it is turned in. In total, this project will be worth 30% of the final course grade.

6. A mid-term examination (worth 15% of the final course grade) and a final examination (worth 20%) will be given as shown in the course schedule. Each exam will assume familiarity with material in the text, covered in lecture, and/or used in homework problems or the project. Exams will be open book (course text only), open notes.

7. Your final grade will be computed on the basis of a weighted sum of the items listed above.

Summary:	Quick check questions	5%
	Labs	30%
	Term Project	30%
	Exams	35%
		100%

The following are minimum guaranteed grades for the percentages indicated:

	93% - 100%: A	90% - 92.9%: A-
87% - 89.9%: B+	83% - 86.9%: B	80% - 82.9%: B-
77% - 79.9%: C+	73% - 76.9%: C	70% - 72.9%: C-
67% - 69.9%: D+	63% - 66.9%: D	60% - 62.9%: D-

POLICY STATEMENT ON EXTENSIONS AND INCOMPLETES:

1. Extensions of the due dates for homework or projects will be given in the event of extenuating circumstances (such as illness, personal emergency) IF you submit a brief written request to the professor as soon as possible after the circumstances arise. This request will be initialed (if approved) and will be returned to you. You must attach it to the piece of work for which the extension was granted.

2. A grade of Incomplete will be given without penalty IF you are unable to complete the course work by the last day of the term due to major illness or other similar emergency. Again, a written request should be submitted. Such a request will only be granted if you are substantially up-to-date with your course work and were making good progress in the course up to the time that the difficulty arose. Of course, you must complete all work for the course by the midpoint of the next semester in accordance with College policy.

3. A grade of Incomplete with a penalty of one letter grade to be applied in the final grade computation MAY be given if you are unable to complete all the course work for reasons other than those noted above. You must make a written request, and your progress in the course, class attendance etc. will be taken into consideration in determining whether to grant it. Again, you must complete all work for the course by the midpoint of the next semester.

ATTENDANCE POLICY:

Regular class attendance is expected of all students, and class attendance will be recorded. Absences from class will be classified as “documented” or “undocumented”. A documented absence is one where written documentation is submitted supporting an absence from class due to circumstances beyond the student’s control. An undocumented absence is any other absence, including one which could qualify as documented if proper documentation were submitted.

Students who have more than three absences (of any kind) during the semester should expect to see their final grade reduced by 1% for every undocumented absence and students who have more than 12 undocumented absences will fail the course automatically. Note that it is not necessary to document absences unless there are more than three total absences; for most students, this will avoid the need to submit documentation. A student who anticipates the need to miss more than three classes due to athletic competitions or other student activities should review the college's attendance policy on page 31 of the catalog, and should then discuss alternatives to class attendance with the professor at the start of the semester.

A student who is habitually late will have late arrival for class counted as a half absence for that class, and a student who sleeps through most or all of a given class session will be counted as absent for that class.

You may ask the professor to waive this policy for you if you earned an A in the prerequisite course, or if you have an A average in this course as of the mid-term exam. If you wish to take advantage of this exemption, you must so inform the professor. However, the attendance policy will be reimposed if your subsequent work deteriorates.

STUDENTS WITH DISABILITIES:

Gordon College is committed to assisting students with documented disabilities (see Academic Catalog Appendix C, for documentation guidelines). A student with a disability who may need academic accommodations should follow this procedure:

1. Meet with a staff person from the Academic Support Center (Jenks 412 X4746) to:
   a. make sure documentation of your disability is on file in the ASC,
   b. discuss the accommodations for which you are eligible,
   c. discuss the procedures for obtaining the accommodations, and
   d. obtain a **Faculty Notification Form**.

2. Deliver a Faculty Notification Form to each course professor *within the first full week of the semester;* at that time make an appointment to discuss your needs with each professor.

Failure to register in time with your professor and the ASC may compromise our ability to provide the accommodations. Questions or disputes about accommodations should be immediately referred to the Academic Support Center. See Grievance Procedures available from the ASC.

TENTATIVE COURSE SCHEDULE

Date	Topic(s)	Reading	Written Work Due
	UNIT I: FUNDAMENTAL CONCEPTS		
W 8/26	Course Introduction; Introduction to Software Engineering		
F 8/28	(continued)	ch. 1	
M 8/31	Review of Fundamental OO Concepts	(Reading should be review): Portions of ch.4: pp. 76-89, 92-100: start at "Inheritance & Generalization"; quick check questions a,b,c + questions given out in class only	
W 9/2	(continued)		**LAB 1**
	UNIT II: REQUIREMENTS		
F 9/4	Requirements Elicitation, Specification, and Validation	ch. 2	
M 9/7	*(Labor Day - no class)*		
W 9/9	Use Cases and Initial Functional Tests	ch. 3	**LAB 2**
F 9/11	(continued)		
	UNIT III: ANALYSIS		
M 9/14	Identifying Objects and Classes	Rest of ch. 4 (pp. 89-92 , 100-111); Quick check questions l and q only	**PROJECT PRELIMINARY MILESTONE**
W 9/16	Class Diagrams in UML	ch. 5	**LAB 3**
F 9/18	(continued)		
M 9/21	(continued)		**PROJECT MILESTONE 1-1 - except Use-Case Diagram**
	UNIT IV: DESIGN		
W 9/23	Representing Associations in Java; Collections	http://java.sun.com/developer/Books/javaprogramming/javaobjects/ch06final.pdf: pp. 213-219; 234	**LAB 4; USE-CASE DIAGRAM FOR MILESTONE 1-1**

F 9/25	Identifying Responsibilities; CRC Cards	Portions of ch. 6 pp. 147-154; Quick check questions a-b plus questions given out in class only	
M 9/28	Modelling Dynamic Behaviors of Systems; Interaction Diagrams in UML	Rest of ch. 6: pp. 155-176; Quick check questions c-j only	**PROJECT MILESTONE 1-2**
W 9/30	State Diagrams in UML	ch. 7	**LAB 5**
	UNIT V: DETAILED DESIGN AND IMPLEMENTATION		
F 10/2	Cohesion and Coupling; the MVC Design Pattern	http://en.wikipedia.org/wiki/Cohesion_(computer_science); same site: Coupling_(computer_science)	
M 10/5	Detailed Class Design and Implementation	ch 10; Questions on Cohesion and Coupling given out last class	**PROJECT MILESTONE 1-3**
W 10/7	(continued)		**LAB 6**
	UNIT VI: QUALITY ASSURANCE		
F 10/9	Quality Assurance; Testing Strategies; Inspection; Correctness Proofs		
M 10/12	(continued)		
W 10/14	(continued)		**LAB 7**
F 10/16	*(Quad Break - no class)*		
	UNIT VII: ADDITIONAL CONCEPTS		
M 10/19	Concurrency; Java Threads; Activity Diagrams in UML	ch. 8	
W 10/21	(continued)		**(Nothing to turn in for Lab 8); PROJECT MILESTONE 1-4**
F 10/23	(continued)		
M 10/26	Architectural Design; Packages; Layered Systems	ch. 9 pp. 219-229; Quick check questions a-f only	
W 10/28	Review and Catch-up		**LAB 9; PROJECT MILESTONE 2-1**
F 10/30	**MIDTERM EXAM (through Unit V - text ch. 1-7, 10)**		

M 11/2	Introduction to Relational Databases	ch. 9 pp. 235-242; Quick check questions j-k only	
W 11/4	(continued)		**(Nothing to turn in for Lab 10); PROJECT MILESTONE 2-2**
F 11/6	Database Querying and Updating	(Handout)	
M 11/9	(continued)		
W 11/11	Java Database Connectivity (JDBC)	(Handout)	**LAB 11**
F 11/13	Relational Database Design		
M 11/16	(continued)		**PROJECT MILESTONE 2-3; PROJECT QUIZ**
Tu 11/17	*(No lab - day of prayer)*		
W 11/18	Distributed Systems; Remote Method Invocation; Deployment Diagrams in UML	ch. 9 pp. 229-231; Quick check questions g-h only	
F 11/20	(continued)		
M 11/23	(continued)		
	(Thanksgiving Break - Lab as usual on Tuesday before break!)		
M 11/30	The User Interface	ch. 9 pp. 231-235; Quick check question i only	**LAB 12 (Done Tuesday before Thanksgiving)**
W 12/2	(continued)		**(Nothing to turn in for Lab 13); PROJECT MILESTONE 3-1**
F 12/4	Design Patterns	ch. 9 pp. 243-245; Quick check questions l-n only	
M 12/7	(continued)		
W 12/9	Review and Catch up		**(Nothing to turn in for Lab 14); PROJECT MILESTONE 3-2**

THURSDAY, DECEMBER 17 - 10:30 AM - 12:30 PM - FINAL EXAM
(Cumulative, with particular emphasis on material since the Mid-Term)