

CPS122 - OBJECT-ORIENTED SOFTWARE DEVELOPMENT

Professor: Russell C. Bjork
 russell.bjork@gordon.edu
Office: KOSC 242 x4377
Hours: MWF 2:10-3:10 pm; Tu 9:00 am-noon
 and by appointment
Course Site: Blackboard site + <http://www.cs.gordon.edu/courses/cps122>

Spring Semester, 2017
MWF 1-2 pm KOS 124
Lab: Tu. 1:15-4:15 pm
(beginning 1/24) KOS 118

PREREQUISITE: CPS121

CATALOG DESCRIPTION:

Introduces object-oriented analysis and design using a statically-typed programming language, encapsulation, inheritance, polymorphism, use cases, UML modeling, and testing methods. Continued development of design and programming skills using UML and Java through weekly laboratories and projects.

COURSE OBJECTIVES:

In general, this course has two major foci: to introduce the object-oriented paradigm, and 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. Explain fundamental object-oriented concepts, such as “object”, “class”, “method”, “encapsulation”, “inheritance”, “polymorphism”, and “event-driven”.
2. Write simple programs using most of the capabilities of the Java language.
3. Analyze a moderately complex problem and design a solution to it, using UML notation.
4. Implement and test such a design, using Java.
5. Carry out the process of designing, implementing, and testing a piece of software as a member of a small team.

TEXTS: Carol Britton and Jill Doake, *A Student Guide to Object-Oriented Development* (Oxford: Elsevier, 2005)
Peter Pasquale. *Java Backpack Reference Guide*. (Boston: Addison-Wesley, 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, and to do the short answer questions at the end of the chapter.

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, programming projects and 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 Britton/Doake text, 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 in Britton/Doake ends with a set of “Quick check questions” that are designed to be done when you read the chapter. You should answer them as part of you reading. On days for which there is a reading assignment in Britton/Doake, we will discuss the corresponding quick check questions, and they will be graded on a "done/not done" basis. I don't expect you to get the “right answer”; all I'm looking for is a good-faith effort prior to class. At the end of the semester, 5% of your final grade will be based on your faithfulness in having answered these questions when due - whether or not your answer is correct. (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. Some exercises done in class will also be included with the above.
4. Most chapters in Britton/Doake 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 when we are discussing topics in it.
5. The DePasquale book is basically a reference, rather than a traditional text book. There are no formal reading assignments in it.
6. Weekly laboratories will focus on gaining practical experience with the material covered in the book and/or in lecture. Lab assignments will be posted on Blackboard ahead of time, and **must** be read over carefully **before** coming to lab. In some cases, you will be explicitly directed to study certain material in preparation for the 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 any assigned pre-lab preparation) and/or a quiz based on the work done in lab given at the end of lab. Each lab with a formal writeup/quizz(es) will account for 2% of the course grade (20% total for ten such labs). The tentative lab emphases are as follows:

Lab	Tentative Emphasis (Subject to change)
1	Introduction to Objects
2	Completing Classes
3	Functional Testing
4	(Lab time used to work on Individual Project 1 - no writeup or quiz)
5	Creating Classes
6	Inheritance and Polymorphism
7	(Lab time used to work on Individual Project 2 - no writeup or quiz)
8	Java Collections
9	Implementing a UML Design
10	(continued)
11	(continued)
12	(Lab time used to work on Team Project - no writeup or quiz)
13	Graphical User Interfaces
14	(Lab time used to work on Team Project - no writeup or quiz)

- In the first half of the semester, you will do two individual programming projects designed to improve your programming skills and help you gain familiarity with using Java. These projects must be done in accordance with the handout "Guidelines for Computer Science Projects", which will be distributed with the first project. You are expected to read these carefully and comply with them exactly. Each project will be worth 10% of the final course grade.
- Throughout much of the semester, you will work on an ongoing design and programming project as part of a team of three or four students. Portions of this project will be due at different times throughout the term, and some lab time will be devoted to working on this project, as shown in the course schedule below. Each portion will be graded individually when it is turned in. In total, this project will be worth 20% of the final course grade.
- 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 exercises or projects. Exams will be open book (course text only), open notes.
- Your final grade will be computed on the basis of a weighted sum of the items listed above.

Summary:	Quick check questions and Exercise Set	5%
	Labs	20%
	2 Individual Projects	20%
	Team Project	20%
	Exams	<u>35%</u>
		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-

ACADEMIC DISHONESTY

From the Gordon College Student Handbook: "Academic dishonesty is regarded as a major violation of both the academic and spiritual principles of this community and may result in a failing grade or suspension. Academic dishonesty includes plagiarism, cheating (whether in or out of the classroom) and abuse or misuse of library materials when such abuse or misuse can be related to course requirements." For the purposes of this course, abuse or misuse of Gordon computer systems or networks related to course requirements will also be viewed as academic dishonesty.

Academic dishonesty will not be tolerated. You know better. Just don't!

POLICY STATEMENT ON EXTENSIONS AND INCOMPLETES:

1. Extensions of the due dates for homework or projects MAY 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.
2. A grade of Incomplete MAY 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. You must apply for this using the form provided by the registrar. 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.

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 undocumented absences during the semester should expect to see their final grade reduced by 1% for each undocumented absence over 3, and students who have more than 12 undocumented absences will fail the course automatically. The allowance of 3 undocumented absences will be reduced by one for each documented absence over 3 - e.g. a student who has 5 documented absences will be allowed only 1 undocumented absence without grade penalty. (This will not be applied retroactively, though) 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 in 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 also Grievance Procedures in Student Handbook).

TENTATIVE COURSE SCHEDULE

<u>Date</u>	<u>Topic(s)</u>	<u>Reading</u>	<u>Written Work Due</u>
F 1/20	Course Introduction; Introduction to Object-Orientation		
M 1/23	From Python to Java		
T 1/24	Lab 1 - Introduction to Objects		
W 1/25	From Python to Java (continued)		Start Individual Project 1
F 1/27	Introduction to Software Development; The Software Lifecycle	Britton/Doake ch. 1	ch 1 Quick Check Questions

M 1/30	(continued)		Individual Project 1 Reading Quiz
T 1/31	Lab 2 - Completing Classes		
W 2/1	Requirements Elicitation, Specification, and Validation	Britton/Doake ch. 2	ch 2 Quick Check Questions a-e only
F 2/3	Use Cases and Initial Functional Tests	Portions of Britton/ Doake ch. 3: pp. 39-55 only	ch 3 Quick Check Questions
M 2/6	(continued)		Individual Project 1 Milestone Due
T 2/7	Lab 3 - Functional Testing		
W 2/8	Identifying Objects and Classes	Britton/Doake ch. 4	ch 4 Quick check questions
F 2/10	Defining a Class		
M 2/13	Encapsulation, Inheritance, and Polymorphism		
T 2/14	Lab 4 - Work Session for Individual Project 1		
W 2/15	(continued)		Individual Project 1 Due
F 2/17	(continued)		Start Team Project
M 2/20	Class Diagrams in UML	Britton/Doake ch. 5	Individual Project 2 Reading Quiz; ch 5 Quick Check Questions
T 2/21	Lab 5 - Creating Classes		
W 2/22	(continued)		Team Project Preliminary Milestone Due
F 2/24	(continued)		
M 2/27	(continued)		Individual Project 2 Milestone Due
T 2/28	Lab 6 - Inheritance and Polymorphism		
W 3/1	Representing Associations in Java; Collections; Arrays		
F 3/3	(continued)		

M 3/6	Review and Catch Up		
T 3/7	Lab 7 - Work session for Individual Project 2		Individual Project 2 Due at end of lab
W 3/8	MIDTERM EXAM (through Class Diagrams)		
F 3/10- F 3/17	<i>(Quad finals and Spring break - no class)</i>		
M 3/20	Identifying Responsibilities; CRC Cards	Portions of Britton/ Doake ch. 6 pp. 147-154	ch 6 Quick Check questions a-b only
T 3/21	Lab 8 - Java Collections		
W 3/22	CRC Cards (continued); Project Class Structure		Team Project Milestone 1-1 Due
F 3/24	Detailed Class Design and Implementation; Unit Testing with JUnit; Version Control with Git	Britton/Doake ch. 10	ch 10 Quick Check Questions a-f only
M 3/27	(continued)		
T 3/28	Lab 9 - Implementing a UML Design part 1		
W 3/29	Modeling Dynamic Behaviors of Systems; Interaction Diagrams in UML	Portions of Britton/ Doake ch. 6: pp. 155-171	ch 6 Quick Check questions c-j only; Team Project Milestone 1-2 Due
F 3/31	State and Activity Diagrams in UML	read Britton/Doake ch. 7; skim ch. 8	ch 7 all Quick Check Questions; ch 8 Quick Check Questions a,b,g only
M 4/3	Graphical User Interfaces and Event- Driven Programming		
T 4/4	Lab 10 - Implementing a UML Design part 2		
W 4/5	(continued)		
F 4/7	Architectural Design; Components; Component and Deployment Diagrams in UML; the MVC and Client/Server Patterns	Portions of Britton/ Doake ch. 9 pp. 221-231	ch 9 Quick Check Questions d-h only' Team Project Milestone 1-3 Due

M 4/10	Design Patterns	Portions of Britton/ Doake ch. 9 pp. 242-245	ch 9 Quick Check Questions l-n only;
T 4/11	Lab 11 - Implementing a UML Design part 3		
W 4/12	(continued)		Team Project Milestone 1-4 Due; (Undividual) Quiz on material furnished by the professor for 1-3
F 4/14	<i>Good Friday - no class</i>		
M 4/17	<i>Easter Monday - no class</i>		
T 4/18	Lab 12 - Work Session for Team Project Milestone 1-4		
W 4/19	User-Interface Design	Portions of Britton/ Doake ch 9 pp 231-235	ch 9 Quick Check Question i only; Team Project Milestone 1-5 Due
F 4/21	(continued)		
M 4/24	Quality Assurance; Testing Strategies; Inspection; Correctness Proofs		
T 4/25	Lab 13 - Graphical User Interfaces		
W 4/26	(continued)		Team Project Milestone 2-1 Due
F 4/28	Cohesion and Coupling	http://en.wikipedia. org/wiki/ Cohesion_(computer _science) ; same site: Coupling_(computer _science)	
M 5/1	Reuse, API's		
T 5/2	Lab 14 - Work Session for Team Project Milestone 2-2		
W 5/3	Exceptions		Team Project Milestone 2-2 Due
F 5/5	Input-Output		

M 5/8 (continued)

T 5/9 *In the rest of the world, today is a
Tuesday - but - by administrative
decree, at Gordon, today is a
Thursday, so no lab*

W 5/10 Review and Catch up

**Team Project
Milestone 2-3 Due;
(Individual) Project
Quiz**

**TUESDAY, MAY 16 - 12:00-2:00 PM - FINAL EXAM
(Cumulative, with particular emphasis on material since the Mid-Term)**