Preparing People to Lead Extraordinary Lives
LOYOLA UNIVERSITY CHICAGO

Department of Computer Science
Water Tower Campus, Lewis Towers 525
820 North Michigan Avenue
Chicago, Illinois 60611-2196
Lake Shore Campus, Cudahy 415/417

Syllabus - COMP 413-001
Intermediate Object Oriented Development (IOOD), Fall 2016
Tuesdays 4:15-6:45pm, LSC Cuneo Room 217
Office Hours: TuTh 12:30-1 in Crown Center 103, & 2-3, in Cudahy Science Hall 417; also possibly for a while after class on Tuesdays

Catalog Description

Object-orientation continues to be a dominant approach to software development. This intermediate programming-intensive course studies the use of classes and objects with an emphasis on collaboration among objects. Expected outcomes are a thorough understanding of the principles of object-orientation: abstraction, delegation, inheritance, and polymorphism; exposure to and use of basic design patterns; and programming experience in a mainstream object-oriented language, primarily Java, with a focus on collections and generics, with application to Android development. More detailed course information is attached (see page 5).

Prerequisites

COMP 271 (http://luc.edu/cs/academics/courses/courselisting/comp271/) – strictly enforced.

Required Textbooks for Class Readings (document links are active)

- Object Oriented Programming Using Java (OOPUJ) by Simon Kendal, available online for free at bookboon.com
- Principles of OOD (compiled by Robert Martin)
- Coad: **Design with Interfaces** and **Archetypes and Color** – pdf versions of these documents can be found in the Sakai → Course Content → General Resources folder
- Agile Principles, Patterns, and Practices in C# (APPPC) by Robert C. Martin and Micah Martin, Prentice Hall, © 2007, ISBN-13: 9780131857254; publisher's site
  - **Note that you may be able to buy a used copy since other students have used this textbook for COMP 313/413 in the past, or you can rent this textbook from Amazon until December 19 for half price at this URL:** http://www.amazon.com/Agile-Principles-Patterns-Practices-C/dp/0131857258/ref=sr_1_1?ie=UTF8&qid=1463237059&sr=8-1&keywords=Agile+Principles%2C+Patterns%2C+and+Practices+in+C%23
- These two free web-based textbooks for Android and Android Studio: http://i-programmer.info/programming/android/5887-android-adventures-getting-started-with-android-studio.html and https://examples.javacodegeeks.com/android/android-tutorial-beginners-with-examples/

Other recommended texts can be found at http://laufer.cs.luc.edu/teaching/313/textbooks.

Course Objectives and Goals

Take your software development abilities to the next level by building on your knowledge of data structures. You will learn to design and implement more complex programs using good software engineering practices, including:

- Designing with interfaces and composition
- Design patterns
- Refactoring
- Test-driven development (TDD)

Course Grading

Your grade will consist of these components with relative weights as follows (I reserve the right to adjust the percentages **in your favor** if circumstances warrant). See in addition the sections on Timely Completion and Academic Honesty.

1. Programming Projects (50%). Java programming assignments throughout the course. These may be individual or team activities. Projects 1-2: 5%; Projects 3 & 5: 12.5%; Project 4: 15%.
2. Tests and Quizzes (45%). Quizzes, tests, and final examination scores. There will be no "make-up" tests or quizzes unless you have made arrangements prior to the dates for them. The initial percentages are: Tests 1-3 10% each; Quizzes 1-3 2% each; Test 4: 9%. Quizzes cover material from the Software Engineering (SE) Radio podcasts; Tests are broader.
3. Class Participation (5%). You are expected to actively participate in the Piazza discussion forum topics and threads, including asking questions about the assignments, sharing information that you have found that could be helpful to other students, and reviewing problems that your team is having with the class projects. In addition, I may have each project team make a short class presentation about any Java-related topic they have learned about on their own that's not covered in class.

I may assign homework in addition to the programming projects, in which case I will let you know how the weights above will be adjusted.

All course work will be graded numerically and your letter grade will be determined from total points earned, weighted as above. Your total points will be converted to a letter grade using **approximately** the following percentage ranges. **These ranges may be adjusted upward or downward based on class performance.** Calculations will round fractional percentages below .500 down to the full number.

90-100 = A	85-89 = A-	
80-84 = B+	75-79 = B	70-74 = B-
65-69 = C+	60-64 = C	55-59 = C-
50-54 = D+	45-49 = D	
44 and lower = F		

If for any reason you do miss a class session, it is your responsibility to determine what you missed, locate any handouts, determine any changes in assignments, course plans, or schedules, etc.

I will not always be covering items from the book in class; additional materials will be added and additional guidance will be given. Information and activities in class and labs that are not in the book will be on exams, quizzes, and helpful for your assignments and programming projects.

Please do not ask for personal "extra credit" to improve your grade as this is neither practical in the course nor fair to your fellow students. I may, however, make one or more extra credit assignments available for the entire class. I will be happy to discuss your performance in the course with you at any time, including discussing possible grade based on performance to date and ways to improve your performance during the remainder of the course. If you are concerned about your grade at any time during the course, or if you have questions or issues about any of the material, please don't wait to set up a time to discuss this with me!

Timely Completion

You are expected to complete all assignments, readings, and projects on time. In computer systems in the "real world", there is always a strong emphasis on getting projects done on time. Use this class to develop your own skills at timely completion.

Personal and team programming projects and other assignments will be due as described at time of assignment. See the class schedule for advanced planning.

You're welcome to ask questions on all assignments and course work, seek additional information on the assignments, and offer observations on the assignments to me either in or outside of class. There will be a class discussion forum called Piazza reachable from Sakai for student use to discuss the course and assignments. This forum will allow the exchange of student ideas as well as interactions with faculty and teaching assistants.

Office Hours and Help

See above for my Lake Shore office hours. I may also provide additional times for online interactions with me and other students (also see the discussion board information below). Additional times may be available by email appointment at ryacobellis@luc.edu, and I encourage you to set up times to talk with me about the course, questions you have, etc. The Department of Computer Science provides tutors for this and other courses; you can find information about tutors on the CS website at http://www.luc.edu/cs/academics/tutoring.

Continuous Improvement

I believe in a personal quality process of continuous improvement. Anything can be improved by applying the quality process of "Plan, Do, Check, Act" (PDCA). To improve the course and the learning of programming, Java, and computer science concepts, I welcome your feedback, comments, suggestions, and complaints at any time.

To support PDCA, I may also ask you to participate in surveys on your impressions of the course and any improvement suggestions; when time permits I will share the results with the class. Your inputs on these surveys are anonymous and will in no way affect your grade.

Academic Honesty

Students are expected to have read the statement on academic integrity available http://www.luc.edu/academics/catalog/undergrad/reg_academicintegrity.shtml. This policy applies to the course. The minimum penalty for academic dishonesty is a grade of F for that assignment. **Multiple instances or a single severe instance on a major exam or assignment may result in a grade of F for the course.** All cases of academic dishonesty will be reported to the CS department office and the relevant college office where they will be placed in your school record.

Academic dishonesty includes, but is not limited to, working together on assignments that are not group assignments, copying or sharing assignments or exam information with other students except in group assignments, submitting as your own information from current or former students of this course, copying information from anywhere on the web and submitting it as your own work, and submitting anything as your own work which you have not personally created for this course. If you do wish to use materials that are not your own, please check with me ahead of time and cite your source clearly. **When in doubt, ask first!**

Students with Disabilities

Students seeking academic accommodations for a disability must meet with Services for Students with Disabilities (SSWD) to verify the disability and to establish eligibility for accommodations. Students may visit SSWD in Sullivan Center - Suite 117, call 773-508-3700, email SSWD@luc.edu or visit LUC.edu/sswd to begin the process. Students should schedule an appointment with their instructor to discuss any academic concerns and/or accommodations. Students are encouraged to contact SSWD as early in the semester as possible.

COMP 413 Course Overview

How COMP 413 fits into the Loyola CS overall series of OO courses:

- **COMP 170 (CS1) – programming in general, classes, and simple objects representing scalars**
- **COMP 271 (CS2) – collections of simple objects**
- **COMP 313/413 – complex, interacting objects; collections/generics; basic design patterns**
- **COMP 473 – advanced design patterns and topics such as AOP (Aspect-Oriented programming)**

Course Topics

- **Data Structures of various types – linear vs. nonlinear, indexing vs. non-indexing, position vs. value-oriented**
- **Advanced Java, e.g. interfaces, annotations, exceptions, generics, collections, boxing/ unboxing, array objects**
- **Object Modeling – UML, use cases, activity and interaction diagrams, class diagrams**
- **Design by contract, interfaces, refactoring & generalization, design patterns (Adapter, Decorator, Composite, Strategy, Iterator, Factory Method, Abstract Factory, Visitor, …)**
- **Agile Development Process – incremental/evolutionary design, test-driven development, refactoring, …**
- **Tools – Eclipse, JUnit; Subversion or other version management tool; JMock, Ant, …**
- **Techniques – object pooling, garbage collection, performance profiling (NetBeans)**

Other Suggested Books:

- **Ken Pugh: Interface-Oriented Design, Second Printing, the Pragmatic Bookshelf, ISBN-10 0-976-69405-0**
- **Martin Fowler: UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition), Addison-Wesley Professional, ISBN-13 978-0321193681**
- **John Ferguson Smart: Java Power Tools, O'Reilly Media, ISBN-13 978- 0596527938**