Computer Science & Engineering Syllabi   College of Engineering & Computer Science

Winter 2012

CS 241-01: Computer Programming - II

Michael Ondrasek
Wright State University - Main Campus, michael.ondrasek@wright.edu

DRSS Dual Enrollment/Distance Learning CS 241 Computer Programming - II
WINTER 2011-2012
Last Update Tuesday November 22, 2011 at 3:00 p.m.

Lecture   **Lab**

CRN/Sect
63881/H90

Description: The CS 241 course is a continuation of CS 240. The emphasis in CS 241 is on solving more complex problems using object oriented programming. Prerequisite: CS240. Students must register for both lecture and one laboratory section.   4 credit hours.

CRN/Sect
63882/H90

Instructor: Mr. Michael Ondrasek Office RC 450 email is michael.ondrasek@wright.edu.

Textbook: Introduction to Java Programming, 8rd edition, Y. Daniel Liang, Prentice Hall - Pearson, 2011, ISBN – 13: 978-0-13-213079-0/ ISBN – 10: 0-13-213079-3.

Textbook Web Resources: See www.pearsonhighered.com/liang and www.cs.armstrong.edu/liang/intro8e. for answers to review questions, solutions to even-numbered programming exercises, source code for the book examples, self tests, errata, etc. You will also want to download the Integrated Development Environments JDK with NetBeans with the latest version of Java from: http://java.sun.com/javase/downloads/index.jsp

Pilot: Pilot allows you access to your grades as well as lab assignments and submittals. We will post much of the course materials on Pilot. To get to Pilot by type **https://pilot.wright.edu/** into your web browser. You can also use the link on Wings which is found under the Academics tab.

Grading: Students must demonstrate their ability to discuss programming issues as well as solve problems. The underlying metric for the determination of a student's overall grade in this course is the mastery of programming and introductory computer science. Students will be provided the opportunity to demonstrate their mastery through examinations, weekly laboratory assignments, and several programming projects. The overall course grade will be determined as follows:

Grades will be assigned on a standard A-90%, B-80%, C-70%, D-60%, F<60% scale. Clustering of grades may cause the thresholds to be lowered; they will not be raised. The instructor reserves the right to fail any student who does not a student attain both an overall passing grade (70%+) in the programming projects.

Programming projects	400 pts. [4 @ 100 pts.]
Laboratory assignments	160 pts. [8 @ 20 pts.]
Mid-term examinations	200 pts.
Final examination	300 pts.
TOTAL	1060 pts.

Policy: There are no late/early/makeup exams unless verifiable emergency and acceptable documentation in writing is provided to the instructor. Although verbal or e-mail notification can be provided, written documentation is required.

Academic Integrity :
1. Be honest at all times.
2. Act fairly towards others. For example, do not seek an unfair advantage over others by cheating with or by looking at other individual's work during examinations or laboratory assignments.
3. Take group as well as individual responsibility for honorable behavior. Collectively, as well as individually, make every effort to prevent and avoid academic misconduct, and reports acts of misconduct that you witness.
4. Know the policy -- ignorance is no defense. Read the policy contained in the student handbook. If you have any questions regarding academic misconduct, contact your instructor.
5.

All work must be your own; sharing of program code will result in a grade of "zero" for all those involved. Official university policy will be followed in cases of academic dishonesty.

What IS allowed: Students are allowed to discuss the general requirements of lab assignments to make certain that they understand the problem and its goal. Students are allowed to ask another student (who has completed the assignment) for (brief) help with a syntax error or other minor problem that does not require extensive exploration of the solution. If another student asks you for help debugging AFTER you have finished the lab assignment, then you may help them briefly, but you may NOT show them your solution. Students may use elluminate for more detailed help. If you work with other student in an allowed manner, you are required to acknowledge the collaboration and its extent in the lab

assignment's comments. This will allow the instructor to comment on and correct the degree of collaboration if necessary. Unacknowledged collaboration will be considered a violation of course policy.

What IS NOT allowed: Students may NOT discuss, look at, or debug other student's projects. Help on projects should come only from on-line help with the TA. Students may NOT work together on lab assignments - students can discuss the lab and/or provide certain help with debugging (see above) but may NOT work together for any extended period of time. Students may NOT use code created by other students or during previous offerings of the course. Students may NOT look at code created by another student (even to debug) until after they have completed the entire lab assignment themselves. Students absolutely may NOT turn in someone else's solution with simple cosmetic changes (say, changed variable names) to the solution -- this is a gross break of academic integrity and will result in a failing grade for the course. *You are responsible for ensuring that other students do not have access to your work* - do not give another student access to your files, do not leave printouts in the recycling bin or printer, do not leave your workstation unattended, etc. If you suspect that your work has been compromised notify your instructor immediately.

Conduct for Examinations: The academic code demands that no student should have an unfair advantage over any other student during examinations. It is strictly forbidden for any student to refer to information from previous offerings of this course unless this information is provided by the instructor to all students fairly. Thus, the use of test banks of previous quizzes or asking questions about examinations or laboratory assignments to prior students is strictly forbidden.

Programming Projects and Laboratory Assignments The instructor will provide a number of opportunities for students to develop their mastery of the subject throughout the course through graded assignments. Laboratory assignments are subject to changes specified by the TA during the laboratory lecture. Assignments must compile to receive credit. Programs that do not compile will not be graded. All programs must have comments at the top that identify the student, the course, and the project type/number. **Late Labs:** All lab assignments must be completed by the due date and time that is specified on the course Web site. In this course we do not accept late labs unless instructor permission is given due to an emergency. Any lab portion which is submitted after the due date and time will receive zero points.
Late Projects: Points will be deducted for projects submitted late. The deduction will be 10% of the total possible points per 24 hours (or portion thereof) elapsed from the moment that the project was due. No points will be awarded for projects that are more than one week late. Begin your projects immediately to guarantee that you have time to get help if necessary and complete them on-time. Deadlines will only be extended for documented emergencies or pre-arranged special needs. Poor time management, corrupt files, or network outages will not be considered a sufficient excuse to extend this deadline. Important note: computers go down, networks fail, and data gets destroyed on the day that a project is due. Plan ahead. Back up your work.

Examinations: All Examinations will occur on the date and time posted below in the Course Schedule unless announced. Students may use one (two-sided) 8.5"x11" page of hand-written notes on the examinations.

Expectations of Students: The Instructor considers it essential to your success in this course that you view all lecture and lab sessions that are posted. Students are expected to study the text. If you miss an unexcused exam/quiz you will receive a zero score. The Instructor feels that it is important that you have your own copy of the correct textbook and edition indicated above. If you have a computer at home, it is important that you practice programming using software discussed in class. If you are having trouble with programs or text readings, it is expected that you will ask the TA during a posted help session. Corresponding with the Instructor or Teaching Assistants by e-mail is a good way to get help with text readings or programming assignments. Finally, it is expected that students will follow the Instructor's recommendations concerning printing of slides and other course materials. In order to minimize handouts, you are expected to print your own copies before lecture whenever possible.

Suggestions: ..Get an early start on each programming assignment. You are urged to budget your lab time wisely and expect to spend considerable time to complete your programming assignments. Get acquainted with the CD/online textbook materials: study the text Preface carefully. You should print, review, and study online materials recommended by the Instructor and Teaching Assistants. You can download the source code for the text examples to try them out. Whenever possible study your text in front of a computer and actively get involved in trying out the programming concepts on your own. You should try to do most of the text review questions and exercises. This can be the most effective way to be successful in the course. It would be a very good idea to get your own USB 2.0 compatible flash drive (also known as a "thumb drive" or "min-drive") to save your labs and projects. **Always backup your programs!** Keep copies of your work in several different places. You can email yourself a backup copy.

Programs: Programming lab assignments will be issued in Pilot. Each assignment will state the due date. Assignments usually will be one or possibly two weeks in duration.

Syllabus Changes: The Instructor will not make changes to this syllabus without notification and understanding of all the students. New copies will be posted and/or provided. Changes would be required for the following reasons: (1) to correct mistakes, (2) to improve student learning, (3) to clarify misunderstands, or (4) to correct serious inconsistencies in policies and/or content compared to other concurrent lecture sections sharing the same labs.

Schedule: See the table below. Topics and order of topics may vary. Exam dates are firm. The topics to be covered each week are listed, followed by the accompanying sections in the text. Not all sections listed are directly covered in detail in class. This schedule is subject to change.

	COURSE SCHEDULE (Liang 8th ed.)	
Week	**TOPIC / ACTIVITY**	**HOMEWORK ASSIGNMENT**
1	Course introduction and review of programming fundamentals	Review: Liang, Ch 1-7 & 9; No labs this week! Project 1 assigned.
2	Introduction to objects: semantics, syntax, and style	Read: Liang, Ch 8 and 10;
3	Object-oriented software construction No class on MLK Day (Monday 1/17)	Read: Liang, Ch 8 and 10. Project 2 assigned.
4	Inheritance and polymorphism	Read: Liang, Ch 11;
5	The object contract: abstract classes, interfaces, and multiple inheritance; Introduction to graphics	Read: Liang, Ch 14 and 12. Project 3 assigned.
5	Midterm examination includes Labs 1-4, Projects 1 and 2 **Friday January 13th**	Know: Liang, Ch 1 – 11 and 14.
6	Event driven programming	Read: Liang, Ch 12, 15 - 16.
7	Event driven programming; Objects and memory	Read: Liang, Ch 12 and 15-16.
8	Recursion, exceptions, and binary I/O	Read: Liang, Ch 13, and 19-20. Project 4 assigned.
9	Threads, concurrency, and Unit Testing	Read: lecture notes.
10	Recitation, Course evaluation, recitation and review	Study/Prepare questions; No in-lab this week.
FINAL	Final examination **Week of March 19th**	