Winter 2012

CS 241-04: Computer Programming - II

Jay DeJongh  
Wright State University - Main Campus, jay.dejongh@wright.edu

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation
https://corescholar.libraries.wright.edu/cecs_syllabi/840

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
Room & Time:  6:05 pm – 7:20 pm, TR, MH 402

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.

Instructor: Dr Jay DeJongh, 341 RC, 775-2555.  E-mail: jay.dejongh@wright.edu  Office hours: 4:30-5:30 T, R.  Other hours by appointment; all you have to do is talk to me and we will find a time to meet.


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://www.oracle.com/technetwork/java/javase/downloads/index.html

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:

<table>
<thead>
<tr>
<th>Component</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming projects</td>
<td>400 pts. [4 @ 100 pts.]</td>
</tr>
<tr>
<td>Laboratory assignments</td>
<td>160 pts. [8 @ 20 pts.]</td>
</tr>
<tr>
<td>Mid-term examination</td>
<td>200 pts.</td>
</tr>
<tr>
<td>Final examination</td>
<td>300 pts.</td>
</tr>
<tr>
<td>TOTAL</td>
<td>1060 pts.</td>
</tr>
</tbody>
</table>

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.

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.

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 go to their TA, the CS help room, or the instructor 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 the course instructor and the CS help room. 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. Thus, 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 period. All students are required to attend their scheduled
laboratory each week. 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. 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 three
days 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:** Examinations will occur at the normally scheduled class time and location unless announced otherwise in
class. The final examination is cumulative and will take place during the university scheduled time period in the normally
scheduled class location unless announced otherwise in class. Students may use one (two-sided) 8.5"x11" page of hand-
written notes on the examinations.

**Absences:** Class attendance will not be a direct factor in your grade but will strongly affect the quality of your education.
Students who miss class are responsible for the material or announcements presented. Any extenuating circumstances
which impact on your participation in the course should be discussed with your instructor as soon as those circumstances
are known. Make-ups for examinations may be arranged ONLY if a student's absence is caused by documented illness or
documented personal emergency. It is the student's responsibility to provide a written explanation (including supporting
evidence) to the instructor in a timely manner. Students registering after the term begins are responsible for all missed
assignments and cannot expect that due dates will be altered. If you miss a lecture or plan to miss a lecture, you may be
able to make arrangements to sit in on the same lecture in another concurrent offering of the course.

**Suggestions:** _Get an early start on each programming assignment._ Most often you will not complete the programming
assignment in the lab sessions. You are urged to budget your lab time wisely and expect to spend additional time outside
of the formal lab to complete your programming assignments. _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 all text checkpoint, review questions, and exercises. _This can
be the most effective way to be successful in the course._ If you are uncertain about how you should do this, please discuss
with the Instructor or Teaching Assistant. _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") for use in labs and possibly at home. _See the Instructor or Teaching
Assistants for recommendations and usage._ _Always backup your programs!_ Keep copies of your work in several
different places. E-mail yourself a backup copy.
**Programs:** Programming lab assignments will be issued in class, during the lab sessions, or on 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 in the class. New paper copies will be 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.

<table>
<thead>
<tr>
<th>COURSE SCHEDULE (Liang 8th ed.)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Week</strong></td>
<td><strong>TOPIC / ACTIVITY</strong></td>
</tr>
<tr>
<td>1</td>
<td>Course introduction and review of programming fundamentals</td>
</tr>
<tr>
<td>2</td>
<td>Introduction to objects: semantics, syntax, and style</td>
</tr>
<tr>
<td>3</td>
<td>Object-oriented software construction</td>
</tr>
<tr>
<td></td>
<td>No class on MLK Day (Monday 1/17)</td>
</tr>
<tr>
<td>4</td>
<td>Inheritance and polymorphism</td>
</tr>
<tr>
<td>5</td>
<td>The object contract: abstract classes, interfaces, and multiple inheritance; Introduction to graphics</td>
</tr>
<tr>
<td>6</td>
<td>Event driven programming</td>
</tr>
<tr>
<td>6</td>
<td><strong>Midterm examination includes Labs 1-4, Projects 1 and 2 Thursday, Feb 9, 2012</strong></td>
</tr>
<tr>
<td>7</td>
<td>Event driven programming; Objects and memory</td>
</tr>
<tr>
<td>8</td>
<td>Recursion, exceptions, and binary I/O</td>
</tr>
<tr>
<td>9</td>
<td>Threads, concurrency, and Unit Testing</td>
</tr>
<tr>
<td>10</td>
<td>Recitation, Course evaluation, recitation and review</td>
</tr>
<tr>
<td><strong>FINALS</strong></td>
<td><strong>Final examination: Thursday Mar 15, 2012, 8:00 pm – 10 pm.</strong></td>
</tr>
</tbody>
</table>