Summer 2013

CS 1181: Computer Science II

Dan C. Wlodarski
Wright State University - Main Campus, daniel.wlodarski@wright.edu

CS 1181 — Computer Science II

Summer 2013

Last Update Monday, May 6, 2013 at 2:00 p.m.

CRN/Sect	Lecture Time	Day	Room
40973/C01	9:50 a.m. – 11:30 a.m.	MW	RC 152C

CRN/Sect	Laboratory Time	Day	Room
40974/C05	11:40 a.m. – 1:20 p.m.	M	RC 152C

Course Description: This is the second course in a two-semester sequence introducing fundamental concepts and techniques for computer science and engineering. The course focuses on problem analysis, advanced programming concepts using JAVA and fundamental data structures. Students learn to analyze problems and evaluate potential solutions with respect to choice of data structures and computational efficiency. Student are exposed to the underlying implementation of basic data structures available in JAVA libraries and develop the skilled needs to extend existing data structures and design new data structures to solve increasingly complex problems. This is an integrated writing course.

Computer Science/Engineering Learning Outcomes:

By the end of this course, a student will have developed:

- competency in recursion and recursive programming
- competency in fundamental data structures and algorithms
- ability to read, reuse and extend high-level code
- competency in analyzing problem requirements and developing program design
- understanding computation cost associated with alternative designs
- ability to understand and apply defensive programming techniques

Integrated Writing Learning Outcomes:

By the end of this course, students will be able to produce writing that

- demonstrates an understanding of course content
- is appropriate for the audience and purpose of particular writing task
- demonstrates the degree of mastery of disciplinary writing conventions appropriate to the course (including documentation conventions)
- shows competency in standard edited American English.

Instructor: Mr. Dan C. Wlodarski

Email: daniel.wlodarski@wright.edu
Office: JC 390
Office Hours: M: 2:00 – 4:30 p.m.,
T & R: 12:00 – 4:30 p.m., or by appointment

Textbook: **Introduction to Java Programming,** 9th edition, Y. Daniel Liang, Prentice Hall – Pearson; one of the 3 choices used in CS 1180: regular text with MPL ISBN – 0133050572; loose page version with MPL ISBN – 0133051463; or E-text version with MPL ISBN – 0132991705. Also, without MPL, listed under: ISBN-10: 0132936526 and ISBN-13: 9780132936521

Internet Resources: See www.cs.armstrong.edu/liang/intro9e 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 and install Oracle's most recent stable release of the NetBeans Integrated Development Environment (NetBeans IDE) for Java Standard Edition (Java SE; the rightmost choice) from: **http://netbeans.org/downloads/index.html**. This installer will include the most recent version of the Java Development Kit (JDK).

Pilot: Pilot allows you to access lab and project specifications, submit source code for credit, and view your grades. Most of the course materials are posted on Pilot. To get to Pilot 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:

Programming projects	450 pts. [3 @ 150 pts.]
Laboratory assignments	400 pts. [9-1 @ 50 pts.]
Quizzes	200 pts. [4 @ 50 pts.]
Mid-term examination	350 pts.
Final examination	600 pts.
TOTAL	**2000 pts.**

Grades will be assigned on a standard $A \geq 90\%$, $B \in (90\%, 80\%]$, $C \in (80\%, 70\%]$, $D \in (70\%, 60\%]$, $F < 60\%$ 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 attain both an overall passing grade (70%+) in the programming labs and 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. Do not cheat by looking at other individual's work during examinations or at another student's laboratory/project 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 laboratory assignment or project 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. Unless directed to do so, 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/project assignment themselves. Students absolutely may NOT turn in someone else's solution with simple cosmetic changes 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. Lab assignments will be issued in class, during the lab sessions, or on Pilot. Each assignment will state the due date. Lab assignments will usually be one (1) or possibly two (2) weeks in duration. The single lowest-graded lab assignment will be dropped from each student's final scores. Lab assignments are subject to changes specified by the TA during the laboratory period. All students are required to attend their scheduled laboratory session each week. In addition to lab assignments, programming projects will be provided by the Instructor to exercise students' ability to integrate several topics into one solution. These programming projects are designed for individual development over longer durations. All submitted source code must compile in Oracle's NetBeans IDE 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 that have 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.

Expectations of Students: Attendance at lecture is not required although it is strongly encouraged and expected. The Instructor considers it essential to your success in this course that you attend all lectures and lab sessions. Students are expected to study the text. **Even when you don't attend class, you are still responsible for material covered in lecture, lab, and in your text readings.** If you miss a lecture, you may also miss a quiz. If you miss an unexcused exam you will receive a zero score. Students are expected to be on time for lecture and lab sessions: lectures and labs start promptly. Early departure from lecture or lab may be unavoidable, but it is expected that this would be quite unusual. 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 do not have a computer, it is expected that you will use the computers in Russ Center Room 152B (or other campus locations) to practice programming skills. Questions are encouraged in lecture and lab; however, if there are no questions it is assumed that students understand the lecture, have read, and understand the text and lab materials. If you are having trouble with programs or text readings, it is expected that you will ask questions in class, come during office hours for help, or make an appointment to discuss your questions as needed. 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 lecture preparations. Lecture preparations may include use of the programming lab that is part of text package.

Suggestions: Get an early start on each programming assignment. Most often you will not complete the programming assignment in one 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 from the author's web site. 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.

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.

Weekly Topics and Assignments

Session	Topic	Reading
Mon, May 6	Introduction, JUG Recap	
Wed, May 8	Recursion *Project 1*	Ch. 20
Mon, May 13	Recursion, Generics, Lists	Ch. 20
Wed, May 15	Basic Data Structures (Java Collections), Stacks, Queues **Quiz 1**	Ch. 22
Mon, May 20	Iterators, Iterators	
Wed, May 22	ArrayList, List, and Linked Lists	Ch. 26
Mon, May 27	Memorial Day UNIVERSITY CLOSED	
Wed, May 29	Linked Lists, Sets, Maps *Project 2* **Project 1 Due Fri**	Ch. 23
Mon, Jun 3	Big Oh Notation	Ch. 24
Wed, Jun 5	Big-Oh Notation, Algorithmic Efficiency **Quiz 2**	
Mon, Jun 10	Algorithmic Efficiency, Midterm Review	Any topics for review
Wed, Jun 12	**Midterm**	
Mon, Jun 17	Intro to Sorting, Bucket, Radix, and Merge sorts	Ch. 27
Wed, Jun 19	Quick and Heap sorts	
Mon, Jun 24	Binary Search Trees	Ch. 27
Wed, Jun 26	Binary Search Trees *Project 3* **Project 2 Due Fri**	
Mon, Jul 1	Hashing Quiz 3	Ch. 28
Wed, Jul 3	Hashing	
Mon, Jul 8	Hashing	
Wed, Jul 10	Divide-and-Conquer, Brute Force Algorithms	Ch. 24 (review)
Mon, Jul 15	Backtracking Algorithms, Dynamic Programming Quiz 4	
Wed, Jul 17	Final Exam Review **Project 3 Due Fri**	
Final	Monday, July 22nd @ 9:50am in RC 152C	

* No lab session