

Wright State University

CORE Scholar

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

Fall 2012

CS 1180-06: Computer Programming - I

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

DeJongh, J. (2012). CS 1180-06: Computer Programming - I. .

https://corescholar.libraries.wright.edu/cecs_syllabi/355

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.

CS 1180-06 Computer Programming - I

Fall 2012

Section 06 – TR 3:30 pm – 4:50 pm in Russ Engineering Center Room 346

Description: Basic concepts of programming and programming languages are introduced. Emphasis is on problem solving and object oriented programming. This course provides a general introduction to the fundamentals of computer science and programming. Examples from and applications to a broad range of problems are given. No prior knowledge of programming is assumed. The concepts covered will be applied to the Java programming language. 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: 11:30-1:30 M, W. Other hours by appointment; all you have to do is talk to me and we will find a time to meet.

Textbook: **Introduction to Java Programming**, 9th edition, Y. Daniel Liang, Prentice Hall – Pearson; one of 3 choices: regular text with MPL ISBN – 0133050572; loose page version with MPL ISBN – 0133051463; or E-text version with MPL ISBN – 0132991705. The same text will be used in the follow on CS 1181 course.
NOTE: YOU MUST USE ONE OF THESE TEXTS BECAUSE THEY GIVE YOU ACCESS TO THE MY PROGRAMMING LAB SYSTEM WHICH IS AN INTEGRAL PART OF THE COURSE AND THE FINAL GRADE. EARLIER TEXTS DO NOT GIVE YOU ACCESS TO THE REQUIRED MY PROGRAMMING LAB SYSTEM.

Textbook Web 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 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 to access your grades, project and lab assignments, and their submittals. 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:

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 attain both an overall passing grade (70%+) in the programming labs and projects.

Programming projects	700 pts. [7 @ 100 pts.]
Laboratory assignments	260 pts. [13 @ 20 pts.]
Quizzes and homework	140 pts.
Mid-term examination	400 pts. [2 @ 200 pts.]
Final examination	500 pts.
TOTAL	2000 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. 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. 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.

myProgrammingLab

Visit <http://myprogramminglab.com/> to access quiz/homework exercises. Information on how to use myProgrammingLab is available there, as well as under the myProgrammingLab content module in pilot.

Integrated Writing: This course is part of Wright States Integrated Writing Curriculum. You will see that all lab and project submissions require documentation which is worth up to 20% of the total lab or project grade. This is done so that Wright State students will be able to produce writing that:

- (1) Demonstrates their understanding of course content,
- (2) Is appropriate for the audience and purpose of a particular writing task,
- (3) Demonstrates the degree of mastery of disciplinary writing conventions appropriate to good program style and documentation, and
- (4) Shows competency in standard edited American English.

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 self-prepared notes on the examinations.

Expectations of Students: We will take attendance, but 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.

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.

Week	Topic	Text Chapters
1	Introduction to programming: Syntax, semantics, basic control flow	1 & 2
2	Representing information: variables, representation, scope	2
3	Strings, Character Class, Tokens	9
4	Conditional structures	3
5	Iteration structures	4
6	Methods: parameter passing; structured decomposition Midterm 1 Thursday Oct 4	5
7	Text and Binary I/O	14.10-14.11 & 19
8	Arrays and lists	6-7, & slides
9	Objects, Classes, and static vs instance	8
10	Unit Testing, immutable objects, aggregation, and this	50 & 10
11	Inheritance and polymorphism & GUI basics Midterm 2 Thursday Nov 8	11 & 12
12	GUI, abstract classes, interfaces, and inheritance	13 & 15
13	Event driven programming	16 & 17
14	Summary and Review	
Final	Tuesday, Dec 11, 2012, 2:45 pm - 4:45 pm	