

Wright State University

CORE Scholar

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

Winter 2012

CS 240-01: Computer Programming - I

Michael Ondrasek

Wright State University - Main Campus, michael.ondrasek@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

Ondrasek, M. (2012). CS 240-01: Computer Programming - I. .

https://corescholar.libraries.wright.edu/cecs_syllabi/843

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.

Distance Learning CS 240 Computer Programming - I

Winter 2012

Last Update Thursday December 22, 2011 at 3:00 p.m.

Lecture

CRN/Sect
62740/90

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: Mr. Michael Ondrasek's Office: RC 450; email: michael.ondrasek@wright.edu. Office hours: M-F: 10:00 – 10:45 am, TR 1:30 – 4:00 pm, and by appointment.

Textbook: Introduction to Java Programming, 8rd edition, Y. Daniel Liang, Prentice Hall - Pearson, 2011, ISBN – 13: 978-0-13-213080-6/ ISBN – 10: 0-13-213080-7.

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: <http://pilot.wright.edu> If you are new to Pilot log on and use the help button (upper right hand corner) withing a particular tab. The help desk (775-4827 can also help you get started. Pilot gives you access to your grades as well as lab and project assignments and submittals. We will post much of the course materials (lecture videos and slides) on Pilot.

Grading: 4 Programming Assignments (37.7%), 8 Laboratory Exercises (15.1%), 2 Examinations (18.9%), and a Final Comprehensive Examination. (28.3%). **The instructor reserves the right to fail any student who does not attain an overall grade average of at least 70% in both the laboratory and project programming assignments.**

Programming projects	400 pts. [4 @ 100 pts.]
Laboratory assignments	160 pts. [8 @ 20 pts.]
Mid-term examinations	200 pts. [2 @ 100 pts.]
Final examination	300 pts.
TOTAL	1060 pts.

Programming Laboratory and Project 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 session 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, thelab sectionm , the name of the TA and instructor, and the assignment type/number (e.g. Lab 1B, Project 1, etc.).

Late Lab Assignments: Although many assignments can be completed during the lab period in which they are assigned, they are not "due" until the following lab period. There is no penalty for turning in a laboratory during the following lab period. However, that is the final due date. Assignments cannot be turned-in/graded after the subsequent lab period. This deadline cannot be waived by the TA. The course instructor may waive the late penalty for documented emergencies.

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 project due date and time. 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. Plan ahead. Back up your work. Start early.

Final grade is based on the course average: A: 100-90, B: less than 90-80, C: less than 80-70, D: less than 70-60, F: less than 60-0.

Lab

CRN/Sect
62845/90

Policy: There are no late/early/makeup exams or quizzes 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. Quizzes and/or take-home exercises/quizzes/etc. may be unannounced. **All work must be your own; sharing of program code/take-home quiz solutions will result in a grade of "zero" for all those involved. Official university policy will be followed in cases of academic dishonesty.** Do not show others your programs and do not look at someone else's code. However, sharing ideas and general computer skills with others outside of class is encouraged.

Expectations of Students: The Instructor 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 for you to attend all lectures and lab sessions. Students are expected to study the text. **Even when you do not 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 an exam or quiz. If you miss an unexcused exam/quiz 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 152C (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 printing of slides and other course materials. In order to minimize handouts, you are expected to print your own copies before lecture whenever possible. The computers in Russ Center Room 152C provide all registered students with the ability to freely print their own copies. Please follow guidelines given in lecture on how to make the best use of the computing and printing resources.

Suggestions: Get an early start on each programming assignment. 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. 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 all text 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 WebCT. 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	Readings*
1	Introduction to Computer Science and Java Programming , binary, abstraction, algorithms, stepwise refinement, and hardware/software. Introduction to NetBeans IDE	Chapter 1
2	Java Fundamentals : data types, variables, constants, literals, operators, scope, casting, Unicode, Programming Style and Documentation , and errors(syntax/runtime). Introduction to Strings and Character classes/methods. Formatted output with <i>printf</i> .	Chapter 2.1 – 2.17 Chapter 9.2 – 9.3 Printf slides
3	Boolean logic, truth tables, logical operators, relational operators, Selections (if, if-else, if-else-if, switch), operator precedence and associativity.	Chapter 3
4	Iteration Statements (while-loop, do-while, and for) Examination 1 (Covers material Ch. 1-3) Exam 1: 7 – 8 p.m. Friday January 27th	Chapter 4.2 – 4.4
5	Using Loop constructs, numeric errors, break vs. continue; File Class, PrintWriter, and Scanner	Chapter 4.5 – 4.11 Chapter 9.6 – 9.7
6	Methods (definition, calling and passing arguments) Local variables, scope and visibility, overloading, returning values from methods, and the Math class	Chapter 5.1 – 5.11
7	Method abstraction, stepwise refinement, top-down/bottom-up design Introduction to arrays (indexing, initialization, and processing), for-each	Chapter 5.12 Chapter 6.1 – 6.2
8	Working with arrays and array algorithms Examination 2 (Covers material in Ch. 1-5 with emphasis on Ch. 4-5) Exam 2: 7 – 8 p.m. Friday February 24th	Chapter 6.3 – 6.6
9	Variable length argument lists, searching and sorting Course Evaluation, Multi-dimensional arrays	Chapter 6.7 – 6.11 Chapter 7
10	Using Java API classes (ArrayList class) Review for Final Exam	Chapter 11.11
Finals Week	Comprehensive Final Exam: 7 – 9 p.m. Friday Mar 16th	

*Whenever any Chapter is assigned, you should also study the corresponding Review Questions and Exercises found at the end of each chapter.