Fall 2011

**CS 240: Distance Learning 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](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/380

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
Fall 2011

Last Update Monday July 25, 2011 at 1:00 p.m.

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 a lecture and laboratory section. 4 credit hours.

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


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: 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 attain an overall grade average of at least 70% in both the laboratory programming assignments and projects.

The 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.

Programming Projects and Laboratory Assignments: Distance learning students must sign into Pilot's Illuminate system the first week. 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. 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 Assignments: 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.

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 three days late.
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.

Academic Integrity: All work must be your own; sharing of program code/quiz or exam 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. The labs and projects you submit must be coded by you. However, sharing ideas and general computer skills with others outside of class is encouraged.

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.
<table>
<thead>
<tr>
<th>Week</th>
<th>Topic</th>
<th>Readings*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to Computer Science and Java Programming, binary, abstraction, algorithms, stepwise refinement, and hardware/software. Introduction to NetBeans IDE</td>
<td>Chapter 1</td>
</tr>
</tbody>
</table>
| 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. | Chapter 2.1 – 2.17  
Chapter 9.2 – 9.3 |
| 3    | Boolean logic, truth tables, logical operators, relational operators, Selections (if, if-else, if-else-if, switch), Formatted output, operator precedence and associatively. | Chapter 3 |
| 4    | Iteration Statements (while-loop, do-while, and for)  
Examination 1 (Covers material Ch. 1-3)  
Exam 1: 6:45 – 8:15 pm Friday September 30th | 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: 6:45 – 8:15 pm Friday October 28th | Chapter 6.3 – 6.6 |
| 9    | Variable length argument lists, searching and sorting  
Course Evaluation | Chapter 6.7 – 6.11 |
| 10   | Multi-dimensional arrays, Using Java API classes (ArrayList class)  
Review for Final Exam | Chapter 7  
Chapter 11.11 |
| Finals Week | Comprehensive Final Exam:  
6:45 – 9:15 pm Friday November 18th 2011 |          |

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