CS 209: Computer Programming for Business II

David M. Hutchison
Wright State University - Main Campus, david.hutchison@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

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.
Syllabus

**Course Information**

Course title: Computer Programming for Business II  
Course number: CS 209 Section 01, Lab Section 01  
Course discipline: Computing Sciences  
Course description: CS 209 is the second in a sequence of two programming classes required for MIS majors. This course will continue teaching students to the basic concepts of programming. Examples are from business applications and emphasis is on problem solving with the computer as a tool.  
Course date: Wednesday, September 10, 2008 through Wednesday, November 19, 2008  
Location: 152A Russ Engineering Center  
Meeting day(s): Wednesday  
Meeting time(s): 6:05 - 9:35 (includes lecture and lab)  
Prerequisite(s): CS 208 or equivalent.

**Instructor Information**

Name: David M. Hutchison  
Email: david.hutchison@wright.edu  
Office location: 152A  
Office hours: Before and after class, as needed or by appointment  
Biography: B.S. Computer Science, Wright State University, 1990  
M.B.A. Project Management, Wright State University, 1995  
Previous (1997-2001) and current instructor of CS 208 and CS 209  
Instructor of various independent Java, C programming, and Oracle-related Java courses

**Teaching assistants:** TBD

**Course Goal**

Course goals: This course, in conjunction with CS 208, is designed to help students achieve a high degree of proficiency in intermediate level programming skills.

**Course Policies**

Introduction: All course policies are subject to change.

**Course Requirements**

Requirements: This course assumes successful (i.e., passing) completion of CS 208 or
equivalent. You are required to have a thumb drive or similar media. Although
not required, a backup is highly recommended as well.

Course Schedule
Schedule: The course schedule presented in the WebCT Calendar is tentative. We will do
our best to adhere to this schedule, but events beyond our control (e.g.,
weather) may impact the dates and content. I will do my best to not slip any
exam dates.

Class Attendance
Policy: It is your responsibility to attend all classes - attendance will not be taken.
Since this class is held only once a week, you stand to miss out on a lot of
information, quizzes, labs, project hints, etc. if you choose to skip a class. Of
course, prior notice of your missing class would be appreciated.

Grading Policy
Policy: Grading is on a straight 10-point increment scale. That is, >90 is an A, 80-89 is
a B, 70-79 is a C, 60-69 is a D, and <60 is an F. Weights of exams,
assignments, etc. are as follows:

- 25% - Mid-term exam
- 25% - Final exam
- 25% - Programming assignments
- 15% - Quizzes
- 10% - In-class labs

Remember, your grade is weighted - it is not a straight points-based
computation.

Academic Dishonesty
Policy: Violators will receive an F for the course and will be reported to the university
- official university policy will be followed (Click here for the policy). You are
required to work individually on your programming assignments. You are
permitted to exchange ideas with your peers, but you are not permitted to use
someone else's work. Additionally, you may not share your work with
someone else. If you choose to violate these rules, then all students involved
will suffer the consequences.

Course Lab
Lab: You must be enrolled in the lab associated with this class (i.e., CS 209 lab,
section 01). The WebCT site for the lab will not be used.

In-class Lab Assignments
Lab assignments will be provided during each lab session. These assignments are to be worked on during lab and submitted prior to the end of the night’s lab session, regardless of whether or not they are complete. These assignments will consist of straightforward coding problems such as writing source code for incomplete programs, or designing a complete Java application. There will be five lab assignments. Each lab will be worth 50 points.

**Course Exams**

Exams: All students are required to take both exams. Make-up exams are only given on a case-by-case basis. If you are unable to attend an exam, you are required to provide an acceptable and documented reason prior to the exam.

**Course Quizzes**

Quizzes: Five quizzes will be given during the quarter. Makeups will not be allowed without advanced coordination with the instructor (and no one other than the instructor).

**Programming Assignments**

Assignments: There will be three programming assignments over the course of the quarter. Each of these assignments is worth 100 points, each will state the required due date, and each will state the requirements for that assignment (e.g., provide a design, test cases, source code, etc.). You are required to earn at least 75% of the total points for all assignments. Failure to earn 75% of the total points will result in failure of the entire course. Late assignments will only be accepted for documented reasons, previously arranged with the instructor (i.e., not the lab TA). Please see the lab TA for a description of how points will be allocated for each programming assignment.

**Textbook**

Required reading: *Starting Out with Java: From Control Structures through Data Structures*, Tony Gaddis and Godfrey Muganda, Addison Wesley, 1 Pap/Cdr edition (August 14, 2006), 978-0321421029

**Midterm Exam**

Date: October 15, 2008
Content: Modules 1-5 (i.e., Chapters 5, 6, and 8).

**Final Exam**

Date: November 19, 2008 - 8:00pm til 10:00pm
Content: Comprehensive, with a concentration on modules 6-9 (i.e., Chapters 9, 11, and 12). I will do my best to schedule a review session from 7:00 til 8:00 that
evening. You are welcome to ask any questions during the review session. My intent of that review session is to allow you to ask any last minute questions about the material on the final exam. This time is for you - use it!

**Module 1**
Date: Wednesday, September 10, 2008
Topics:
- Course Overview
- Methods

Readings: Gaddis Chapter 5

**Module 2**
Date: Wednesday, September 17, 2008
Topics:
- Classes and objects
- Building a simple class
- Instance fields and methods

Readings: Gaddis, Chapter 6

**Module 3**
Date: Wednesday, September 24, 2008
Topics:
- Constructors
- Overloading methods and constructors
- Scope of instance fields

Readings: Gaddis, Chapter 6

**Module 4**
Date: Wednesday, October 1, 2008
Topics:
- Introduction to arrays
- Processing array elements
- Copying arrays
- Passing arrays as arguments to methods
- Useful array algorithms
- Returning arrays from methods
- String arrays

Readings: Gaddis, Chapter 8

**Module 5**
Date: Wednesday, October 8, 2008
Topics:  
- Searching arrays using the Collections framework
- Two dimensional arrays
- Ragged arrays
- Arrays with 3+ dimensions
- Command line arguments and arrays
- The ArrayList class

Readings:  
Gaddis, Chapter 8

Module 6  
Date:  
Wednesday, October 15, 2008
Topics:  
- Static class members
- Passing objects as arguments to methods
- Returning objects from methods
- The toString method
- The equals method
- The ‘this’ reference
- Garbage collection

Readings:  
Gaddis, Chapter 9

Module 7  
Date:  
Wednesday, October 22, 2008
Topics:  
- Inheritance
- The “is a” relationship
- The superclass constructor
- Overriding methods
- Access modifiers

Readings:  
Gaddis, Chapter 11

Module 8  
Date:  
Wednesday, October 29, 2008
Topics:  
- Multiple layers of inheritance
- The Object class
- Polymorphism
- The instanceof operator
- Abstract classes
- Interfaces
Readings: Gaddis, Chapter 11

Module 9
Date: Wednesday, November 5, 2008
Topics:
- Handling exceptions
- The Exception class
- Polymorphic references to exceptions
- Handling multiple exceptions
- The ‘finally’ clause
- The stack trace
- When an exception is not caught
- Checked and unchecked exceptions
- Throwing exceptions

Readings: Gaddis, Chapter 12

Overflow
Lesson: Overflow
Date: Wednesday, November 12, 2008
Objectives or Goals: This class is basically a buffer or overflow for the 9 modules of this course. I'm willing to bet that we'll need it!