

Wright State University

## CORE Scholar

---

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

---

Spring 2007

### CS 142: Computer Programming - II

Eric Maston

*Wright State University - Main Campus*

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

Maston, E. (2007). CS 142: Computer Programming - II. .  
[https://corescholar.libraries.wright.edu/cecs\\_syllabi/419](https://corescholar.libraries.wright.edu/cecs_syllabi/419)

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](mailto:library-corescholar@wright.edu).

# *Computer Science 142 - Computer Programming 2*

**Spring Quarter 2007  
Wright State University**

## **Course Description**

This course provides a continuation to the general introduction of computer programming begun in CS 141. Examples from and applications to a broad range of problems are given. The concepts covered will be applied to the Java programming language. (Students must register for both lecture and laboratory sections.)

## **Goal**

There are several goals to accomplish in CS 142:

1. Learning more advanced programming skills.
2. Conceptualize how computer programs are logically developed.
3. Develop an appreciation for systems and programming.
4. Write real and usable applications.
5. Have some fun!

## **Lecturer**

Eric Matson

Office: 336 Russ Engineering Center

Phone: 937-775-5108

Office Hours: M W 4:00 – 5:30

Email: [eric.matson@wright.edu](mailto:eric.matson@wright.edu)

Web: [agents.cs.wright.edu](http://agents.cs.wright.edu)

Class: MW(F) 12:15 – 1:20 Russ Engineering Center 148

## **Text**

Starting Out with Java, Tony Gaddis

## **Prerequisites**

For this class the official prerequisite is *CS 141*. Please let me know the first lecture if you do not meet this prerequisite, and we can talk about your preparation if it differs.

## **Grading**

Programming Lab Assignments 50%

Midterm Exam & Quizzes 20%

Final Exam 30%

The base scale is: A: 90-100, B: 80-89, C: 70-79, D: 60-69, F: 0-59. This is the highest requirement that will be used. The scales may be lowered or revised if necessary, **unless you get less than 70% of the possible points on your programming lab assignments in which case you fail the entire course regardless of your overall course average.**

## **Policies and Notes**

- Attendance: Attendance is not required, nor will it be taken after the first couple of lectures. If you are not a regular attendee, it will be your responsibility to seek out what material was covered in the lecture and learn it. Most of my exam questions will be taken directly from ideas covered during the lecture, so it greatly helps if you attend!
- I will utilize my CS web page ([agents.cs.wright.edu](http://agents.cs.wright.edu)) to post updates to the course, sample code, projects, announcements, schedule, etc. Get in the habit of checking it regularly.
- Always make back ups of all of you work. Never have just one copy of anything!
- If you are going to miss an exam, for any reason, discuss it with me in advance. If it is an emergency situation, please notify me as soon as possible.
- You can reach me a number of ways. Email is normally the best as I check it about 18 hours a day normally. You can also reach me by phone during the day at 775-5108. If you need human contact either stop in during my office hours, make an appointment, or just come by my office. If I am in and not on a deadline to get something else completed, I will normally try to help as much as possible.
- There are technologies we will use in this class that you may not already know, such as file transfer, command line, text editors, file systems, etc. We will cover some of these technologies as we go.
- The key to learning in this class will be spending time working through the problems. Don't wait until 2 hours before something is due to try to learn the concept and then write the program. This normally ends in a disaster! Stay up with the readings and try to work through some of the examples in the book. I will post what I call, "10 minute programs" which are exercises that you can work through to learn key concepts. And yes, they are programs you can write and execute in 10 minutes (unless you are a really slow typist, like me. In that case, they become "20 minute programs".)

### **Academic Misconduct**

In this class, the only way to truly learn the concepts to is do the work yourself. I encourage working with other people on the course concepts. When you begin to write the program, complete and submit your own work.

Work that has obviously been copied or in the more extreme case, when the original author's name has not even been changed, both parties will receive a 0 grade for that assignment. Both parties will also be turned over to the Office of Judicial Affairs.

**Schedule** (always subject to changes)

<b>Week</b>	<b>Topic</b>	<b>Reading</b>
1	Review, Objects and Classes	Chapter 6
2	Strings, Command Line, Inheritance	
3	Inheritance, Polymorphism, Abstraction	Chapter 11
4	Interfaces, GUI Programming	Chapter 7
5	GUI Programming	
6	Event Driven Programming	
7	Events, User Interfaces	
8	User Interfaces	Chapter 13
9	Interfaces, Applets	Chapter 14
10	Applets, Review	