

Wright State University

## CORE Scholar

---

Computer Science & Engineering Syllabi

College of Engineering & Computer Science

---

Fall 2009

### CS 400-01: Data Structures and Algorithms

Sarah Gothard

*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

Gothard, S. (2009). CS 400-01: Data Structures and Algorithms. .

[https://corescholar.libraries.wright.edu/cecs\\_syllabi/291](https://corescholar.libraries.wright.edu/cecs_syllabi/291)

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).

## CS 400-01 Fall Quarter 2009

### Syllabus

---

#### Professor

- Dr. Sarah Gothard
- Russ Engineering 437
- sarah.gothard@wright.edu

#### Lecture

- 6:05 p.m. - 7:20 p.m., Mondays and Wednesdays, Russ Engineering Center 154

#### Office Hours

- Monday, Wednesday 1:00 p.m. - 3:45 p.m.
- Tuesday, Thursday, 1:00 p.m. - 2:00 p.m.
- By appointment

#### Textbook

- *Data Structures and Algorithm Analysis in C++* by Mark Allen Weiss. (Any edition.) Addison Wesley.

#### Language

- C++ in Unix Environment

#### Grading

- 10-point scale: 90.0-100.0 A, 80.0-89.9 B, 70.0-79.9 C, 60.0-69.9 D, 0.0-59.9 F
- Programming assignments: 40%
- Midterm: 20%
- Final: 25%
- 1 Presentation: 10%  
*Presentation must be completed to pass the course with a C or better.*
- Homework/quizzes: 5%

#### Deadlines

- Work is due at the specified deadline. Late work will not be accepted. If submitting the assignment appears to fail, email a copy of the assignment to me before the deadline.
- If a project is only partially completed you should submit what you have by the deadline.

- All programs should correctly compile and have been tested. There should be no infinite loops, system locking, or invalid output. It is your responsibility to seek help for compilation errors or unreasonable output. The programs should run according to the handout requirements.
- If your program has a problem, attach a readme file containing all pertinent details of the problem, what steps were taken to correct the problem, and any ideas you have as to what is wrong. (source: Dr. M. M. Rizki)

### **Academic Integrity**

- You may help other students in this section of the class with minor debugging (e.g. "You are missing a semicolon."), but not with logical errors (e.g. "You need a loop instead of a branch."), provided you have already completed the part of the assignment you are helping to debug.
- You may discuss the assignments to the extent that 1) You do not problem solve for others and 2) You never discuss actual code.
- Code from any source other than your mind is prohibited unless 1) You have explicit or implicit permission to use the source (e.g., textbook, provided sample code) and 2) You document the source. Cheating will be taken very seriously, resulting in harsh penalties.

### **Instructor Late**

If the instructor is late for class, students are expected to wait for 15 minutes after the class period starts before leaving.