| | | | |
|---|---|---|---|
| Course Number/Section and Title: | COMP 2100-01 Abstractions: Data and Algorithms | | |
| Semester and Year: | Fall 2015 | | |
| Course Meeting: | MW | 1:40-2:50 | Towers 115 |
| | TR | 2:00-3:45 | Towers 107B |
| | Days | Time | Location |
| Credit Hours: | 4 | 2.7 | 1.3 |
| | Total Credit Hours | Lecture Credit Hours | Lab Credit Hours (if applicable) |


| | | | |
|---|---|---|---|
| Instructor: | Pete Sanderson | psanderson@otterbein.edu | |
| | First, Last | Email Address | |
| | Towers 125 | 823-1317 | @DrPeteOU on Twitter |
| | Office Location/Room # | Office Phone Number | Other Number or preferred contact information |
| | 10:00-11:00 MTWRF | | |
| | Set Office Hours (Days and Time) Also available by appointment. | | |


**Course Catalog Description,** including pre- or co-requisite course work or other required items.
An introduction to fundamental data structures and computing algorithms within an object-oriented context. Principles of data abstraction and representation are examined. Additional topics include specification, design, use, and implementation of abstractions; recursion; and intuitive analysis of algorithms. Prerequisite: COMP 1600

**Course Objectives** (learning outcome goals or student learning outcomes for the course)

- You will be able to describe and apply principles and techniques for implementing object-oriented software components. We will work from the viewpoint of the component developer more than that of its client.
- You will be able to describe the functionality of and apply implementation techniques for several collection abstractions such as lists, stacks, queues, trees, hash tables and graphs.
- You will be able to evaluate the performance characteristics of the collection abstractions and sorting techniques.
- You will be able to develop (design, implement, test) solutions that utilize collection abstractions appropriately.

**Program Learning Goals or Outcomes**, *as applicable* (major, INST, FYE, SYE, HNRS, WI)
We have defined a set of 11 Student Learning Outcomes (SLO) for the Computer Science major. Your work in this course contributes to the following SLOs:

2. Students can methodically solve algorithmic problems.
3. Students understand the recurring themes of abstraction and computation.
4. Students are proficient in at least one programming language.
5. Students are proficient in a software development paradigm.
7. Students can independently learn and apply new methods and tools.

**Required Texts and/or Ancillary Materials**
Data Structures: Abstraction and Design Using Java, Second Edition, Koffman & Wolfgang, Wiley, 2010. ISBN 978-0-470-12870-1. This text is also available as an eBook. See the textbook website for further information.

**Attendance and Participation Policy**
We will follow some active learning practices in the classroom and particularly in the lab, including collaborative activities and projects. Your attendance and participation is essential to not only your success but also to the success of your collaborators. You are however tuition-paying adults capable of assessing the consequences of not attending class and lab sessions. I do not assign points for attendance or participation.

**Method for determining course grade**
Your grade is based on a final course score in the range 0 to 800. The components of this score and their weights are as follows: midterm exams 200 (25%), final exam 200 (25%), programming projects 200 (25%), programming exercises 120 (15%), quizzes 80 (10%). Ranges in the chart represent a 90-80-70-60 scale with 2% plus or minus on either side.

| Range | Grade | Range | Grade | Range | Grade |
|---|---|---|---|---|---|
| 736 - 800 | A | 640 - 655 | B- | 544 - 559 | D+ |
| 720 - 735 | A- | 624 - 639 | C+ | 480 - 543 | D |
| 704 - 719 | B+ | 576 - 623 | C | 0 - 479 | F |
| 656 - 703 | B | 560 - 575 | C- | | |

**Assignments/Tests and expectations for out-of-class work**

*Exams:* There will be two midterm exams and a comprehensive final exam. Exams cover lecture and textbook material, lab exercises, and projects. Make-ups will be scheduled only for documented emergencies.

*Quizzes:* I will regularly post short quizzes on Blackboard. Most quizzes will be worth 5 points and must be completed *prior to* the classroom discussion of their topics. This will encourage you to read assigned material in advance.

*Lab Exercises:* I will periodically assign individual laboratory programming exercises. They are intended to give you experience with and improve your skills in software implementation. You will apply and further develop those skills through larger individual and team projects.

*Lab Projects:* I will assign several programming projects to be solved either individually or by teams. They will help you further develop not only technical skills from the homework assignments but also your teamwork and interpersonal skills.

**Deadlines for submitting work**
Online quizzes must be completed by the provided deadline, normally the start of the class period for that material. Deadlines are enforced automatically by Blackboard. Programming projects and quizzes are assigned a due date and will incur a 10% per day penalty for late submission.

**Final Exam Date and Time**
Tuesday December 8, 2:30-3:30

**Course, major, program or department-specific formats, policies and procedures**, *as applicable* (e.g., specifics for field work, respect for others, lab safety, distance/online, discussion guidelines, student participation in co-curricular activities such as INST and FYS-approved events)

Does not apply.

**Academic Honesty**

This class will include both individual and collaborative assignments. In either case, you are encouraged to help each other learn the course material. Participants in these discussions enjoy the benefit of deeper and greater learning.

For individual assignments, the work you submit for evaluation must be your own; created by you while thinking it through. Any individual work submitted for evaluation (assignments and exams) that includes work done by another, copying of another's work, or the result of following another's step-by-step keystrokes and mouse clicks, violates the academic integrity policy of this course. For team assignments, the work you submit will be the product of the team. Taking credit for work that you did not participate in also violates the academic integrity policy of this course.

When academic misconduct occurs as described above, you will receive a zero grade for that assignment or exam. The misconduct may also be reported to the Office of Academic Affairs. If a previous academic misconduct offense is on your record, you will receive a grade of F for this course and a referral to the judicial system.

The complete statement on Plagiarism, Cheating and Dishonesty can be found in the *Campus Life Handbook*, page 33, at the following web link: http://www.otterbein.edu/public/CampusLife/HealthAndSafety/StudentConduct.aspx.

**Learning Differences**

If you have a documented learning difference please contact Kera McClain Manley, the Disability Services Coordinator, to arrange for whatever assistance you need. The Disability Services is located in Room #13 on the second floor of the Library in the Academic Support Center. You are welcome to consult with me privately to discuss your specific needs. For more information, contact Kera at kmanley@otterbein.edu, 614-823-1618 or visit the Disability Services at the following web link: http://www.otterbein.edu/public/Academics/AcademicAffairsDivision/AcademicSupportCenter/DisabilityServices.aspx.

**Statement on Credit Hour Definition/Expectation for Student Work**

For each credit hour of classroom or direct faculty instruction, students are expected to engage in two hours of out-of-class course-related work (readings, homework, studying, project preparation, etc.). Most Otterbein classes meet four hours per week in-class with expectation for an additional eight hours of out-of-class work. This class however combines classroom and lab activities. Lab projects are designed to be completed during the lab periods but depending on your learning style you may require additional time either in the lab during open hours or on your own computer. Since this course meets about six hours per week when combining classroom and lab, our expectation is you will spend an additional six hours per week in out-of-class course-related work.

**Fall 2015 Schedule**

| Week | Dates | Chapter | Topics |
|---|---|---|---|
| **1** | **Aug 24-28** | Appendix A | Introduction<br>Java |
| **2** | **Aug 31 - Sep 4** | 1.1-1.4<br>1.6 | OO Concepts<br>Exceptions |
| **3** | **Sep 7-11** | <br>2.11 | *Labor Day: No class Monday*<br>Testing |
| **4** | **Sep 14-18** | 2.1, 2.2<br>2.4 | Java Collections Framework, List<br>Runtime Performance |
| **5** | **Sep 21-25** | 2.5<br>2.6 | Single Linked Lists<br>Double Linked Lists |
| **6** | **Sep 28 - Oct 2** | | **Exam #1 over Chapters 1-2**<br>address issues revealed by exam |
| **7** | **Oct 5-9** | 3.1-3.3<br>4.1-4.3 | Stacks<br>Queues |
| **8** | **Oct 12-16** | <br>5.1, 5.2 | *Fall Break: No class Monday - Tuesday*<br>Recursion |
| **9** | **Oct 19-23** | 5.3, 5.6<br>6.1, 6.2 | Recursion<br>Trees and Binary Trees |
| **10** | **Oct 26-30** | 6.4<br>6.5 | Binary Search Trees<br>Heaps |
| **11** | **Nov 2-6** | 7.1, 7.2<br>7.3 | Tree Maps and Tree Sets<br>Hash Tables and Hashing |
| **12** | **Nov 9-13** | <br>8.1-8.5 | **Exam #2 over Chapters 3-7**<br>Sorting |
| **13** | **Nov 16-20** | 8.7, 8.9<br>9.1 | Sorting<br>Balanced Search Trees |
| **14** | **Nov 23-27** | 9.2 | Balanced Search Trees<br>*Thanksgiving: No class Wednesday - Friday* |
| **15** | **Nov 30 - Dec 4** | 10.1, 10.4<br>10.4, 10.6 | Graphs<br>Graphs |
| **16** | **Dec 7-11** | | Wrap-up and Review<br>**Final Exam: 2:30 p.m. Dec 8** |

*Useful Lecture Notes:* I have posted my Autumn 2007 CSC 205 (COMP 2100 under quarters) schedule with links to a complete set of lecture notes. They are from a different textbook but you will recognize nearly all topics. The URL is http://faculty.otterbein.edu/psanderson/csc205/schedule.html