Course Syllabus
CS-220: Computer Systems II
Spring 2018

Credits /Contact Hours 4 credits, Three 60 minute lectures and one 85-minute lab per week

Webpage: http://www.cs.binghamton.edu/~tbartens/CS220_Spring_2018/

Instructor: Tom Bartenstein E-Mail: mailto:tbartens@binghamton.edu
Office Hours: See class Web Page.

Textbooks

Computer Systems: A Programmer's Perspective, Third Edition, Randal E. Bryant and David R. O'Hallaron, Prentice Hall. (There will be no specific reading assignments, but each lecture will identify the associated relevant sections of the text. Read these sections before the lecture to prepare for the lecture, and/or after the lecture to get a different perspective on the material.)

Supplementary Text: The C Programming Language, Second Edition, Brian Kernighan and Dennis Ritchie, Prentice Hall, 1988. (I strongly recommend that you purchase your own copy of this textbook and keep it even after this semester. This book will serve you your entire career as the single most useful and important reference book on the C language.)

Course Description We view programming as an abstract endeavor, but that abstraction hides much more detailed and meticulous low-level implementations. In this course, we investigate some of the implementation details that bleed into our abstract view of the world, and demonstrate that understanding lower level implementation can often lead to much higher quality software and user experiences.

We start with the C programming language constructs (control and data structures, pointers, arrays and functions), and delve into data representation; compilers, assemblers, macro-processors, interpreters, linkers, and loaders; assembly and machine language programming; instruction set architectures, encoding and addressing modes; the architecture and programming of digital computers; the relationship of C constructs to the underlying architecture; processor, memory and I/O organization, and if time permits I/O techniques and interrupts.

Supervised laboratory work involves programming and debugging using machine language, assembly language and C.

Prerequisites

- CS-120 COMPUTER SYSTEMS I: MACHINE ORGANIZATION
- CS-140 PROGRAMMING WITH OBJECTS

Required for Major: Successful completion of CS-220 is required for a Computer Science major.

Course Objectives Upon completion of this course students will understand:

- how to write application software.
- how to develop software using C.
- implications of other CS topics (architecture, languages, operating systems, algorithms, etc.) on application development.

Lecture Notes

Lecture Notes for each lecture will be posted on the class web-site in PDF format before lectures. *Lecture notes do not substitute for class attendance*, since they will not be complete and significant parts of lectures, including discussions and in-class exercises, may not come from the class notes.

Grading

Your grade will be based on**:**

Pop Quizzes, Attendance, Class Participation	10%
Homework	10%
Labs	10%
Projects	30%
Tests (2 tests 10% each)	20%
Final Exam	20%

Most of the components of your grade will appear on myCourses but your final average will include components that are not published. The mapping from a numeric average to a letter grade is not pre-defined, and will be determined by the instructor once all averages have been computed, based on class performance and comparison to CS-220 classes in previous semesters.

Participation Formal attendance will not be taken for each lecture or lab, but attendance is expected. There will be several un-announced quizzes administered throughout the semester during lecture or lab periods. Unexcused absence from a quiz will result in a zero grade for that quiz. If you cannot attend a lecture or lab, e-mail the professor *before* the lab or lecture, and make sure you consult with the professor or TA during office hours afterwards to ensure you know what you missed. Your participation grade will include of an average of the quiz grades, where the numerator is the sum of the quiz grades you received, and the denominator is the

number of *unexcused* quizzes given. Another component of the participation grade will be the professor's subjective evaluation of participation, including the number of times you answer/ask questions in class and lab, the number of times you ask for extra help during office hours, the number of times you contact the professor or TA's via email, and the basic interest and effort you put into the class.

Homework Short homework assignments will be posted on the class web page approximately a week before homework is due. Assignments should not take a long time (about an hour). Answers will be reviewed in class after the due date, and posted on the class web page. Late homework, turned in after the deadline, will get an automatic 50% deduction. Homework will not be accepted after the answers have been reviewed in class. Homework grades and feedback will be posted on myCourses as soon as your submission has been graded.

You may (and are encouraged to) seek help from others, including the instructor, TAs, and classmates. Help includes assistance with:

- using the software tools needed to complete an assignment
- understanding the specifications or requirements of an assignment
- evaluating strategies for solving a problem
- debugging code that you have written
- interpreting compiler and run-time error messages

However, assignments are individual assignments. This means that you must *write* all work submitted by yourself.

Labs Lab instructions will be posted on the class web page prior to the start of the first lab section. Either the TA or the Professor will be available to answer questions during the lab period. You are encouraged to work together during labs, but may work on your own. If you work as a team, make sure that every member of the team understands the lab material, and that each student writes his own code for submission! Material covered in the labs will appear on quizzes, tests, and the final exam. Lab grades will be posted on myCourses once they are available.

Projects There will be four long-term programming/debugging projects assigned throughout the semester. The lowest project grade will be dropped, and the remaining three will be averaged. The instructions for each project will be posted on the class web page. Each project will require a significant time to complete. Please get started on projects early – if you wait until a week before the projects is due, you will not get a good grade! Project assignments will be graded on CS LDAP computers, so please test them there. More specific submission and grading instructions will be included in the project instructions. Late projects will result in a 10% deduction for every day the submission is late.

Tests will be in class, closed notes, and closed book, unless otherwise specified (unlikely). The first test will take place near the beginning of March, the second, near the middle of April. You must complete the test in the time given. Unexcused absence from the test will result in a zero grade for that test. Test grades will be posted on myCourses as soon as they are available.

Getting Help Please utilize the instructors and TA's office hours for questions and discussion of course related material. Our job is to make you successful, and office hours are a great way to get help. E-mail the instructor or TA's if you need an appointment outside of office hours, or e-mail questions or discussions to the professor directly.

Partial Credit We commonly give partial credit to partially correct answers in this course. For that reason, it is always recommended to show your work in developing a solution for homework, test questions, lab questions, and projects.

Academic Honesty Expectations

Please review the academic honesty document and make sure that you understand it! The link is at: https://www.binghamton.edu/watson/about/honesty-policy.pdf. Cheating and copying will NOT be tolerated. For instance, any code turned in will be compared to other students' submissions. If there is significant similarity, all such similar code will receive a zero grade. Obvious violations of the principles of academic honesty include, but are not limited to:

- submitting another person's work (in whole or in part) as your own
- submitting the same work (with or without minor changes such as changes of names and the order of code sections) as another student.

Collaboration Students are encouraged to help one another and to form study groups. In Computer Science. You can learn more from your peers than from your instructors and teaching assistants. Please be generous with your time and expertise. Doing so is good for you and good for them. You are free to discuss assignments *in general terms* with one another. However, please do not share your work directly to other students. Each student must complete your assignments *individually* (unless indicated otherwise by the instructor). Each of you must write your own code, and you must write up all solutions individually. Students submitting solutions (including code) that are determined to be "too similar" will receive zero grades, and will be subject to Academic Honesty violations.

Academic Stress: If you are experiencing undue personal or academic stress at any time during the semester or need to talk with someone about a personal problem or

situation, I encourage you to seek support as soon as possible. I am available to talk with you about stresses related to your work in my class. Additionally, I can assist you in reaching out to any one of a wide range of campus resources, including:

1. Dean of Students Office: 607-777-2804
2. Decker Student Health Services Center: 607-777-2221
3. University Police: On campus emergency, 911
4. University Counseling Center: 607-777-2772
5. Interpersonal Violence Prevention: 607-777-3062
6. Harpur Advising: 607-777-6305
7. Office of International Student & Scholar Services:607-777-2510