You are required to submit a signed and dated version of this syllabus to the TF in your first discussion session indicating that you have read, understood and agree to its contents.

Course Description

This course takes a programmer’s perspective to learn about the inner structure of computer systems, the design and implementation of abstractions that enable humans to use computers efficiently, the basics of C and assembly programming, the mapping between C and assembly, and between assembly and machine language, and the role of operating system software. Our goal is to learn what a "beautiful" computer system is and how it works. Quoting an Italian painter named Carlotti:

Beauty is the summation of the parts working together in such a way that nothing needed to be added, taken away or altered.

We will also learn how to become strong ("brilliant") programmers who write fast and reliable programs. Quoting Albert Einstein:

Computers are incredibly fast, accurate, and stupid; humans are incredibly slow, inaccurate and brilliant; together they are powerful beyond imagination.

CS 210 is a principal course for computer science majors. It provides background for courses in the systems area such as operating systems, compilers, networks, not to mention more advanced courses in computer architecture.
Prerequisites

This course assumes that students have a solid background in Java or C++ programming from CAS CS 111 or equivalent. CS 112 is also recommended, but not essential for students with strong programming skills. A solid working knowledge of operating systems, such as Unix/Linux and Windows, is also assumed. CS 113 or MA 293 is helpful for the material on Boolean logic and data representation, but is not essential.

We will make use of C as an example of high-level language because its syntax and semantics are closer to assembly language concepts. C is more suitable for exposing low-level system details and achieving higher performance in real implementations. If you know C++, you should not be distracted by this since C is mostly a subset of C++. C and C++ both share many of the same fundamental programming constructs. C, however, lacks support for object-oriented programming. On the other hand, if you know Java, there are aspects of C, particularly pointers, explicit dynamic memory allocation, and formatted I/O, that do not exist in Java. Fortunately, the classic K&R required text is an excellent reference to the relatively small C language.

Textbooks


All books are available from the BU bookstore.

Two optional books you might find useful are:


Online Organization

The primary resources for the class is the course piazza [http://piazza.com/class#fall2013/cs210](http://piazza.com/class#fall2013/cs210). Please ensure you are register on the site. If you have any questions or difficulties email the TF. A copy of the syllabus and some additional online resources will be available on the course webpage ([http://www.cs.bu.edu/~jappavoo/webpages/cs210.html](http://www.cs.bu.edu/~jappavoo/webpages/cs210.html)). Finally information regarding grades will be maintained on the course blackboard site ([http://learn.bu.edu](http://learn.bu.edu)).
Lecture Conduct

You are expected to attend all lectures and be an active participant. No messaging or surfing is allowed in class. All electronic devices including phones, tablets and laptops must be silenced and put away unless being used for note taking. If you are using them for other purposes you will be asked to leave class. If used for note taking you will be required to email an electronic version of the notes to the professor at the end of the lecture.

Once the lecture has begun and the door’s closed you will need to knock and ask to enter. Avoid these situations by arriving on time and planning appropriately.

Grading

Your final grade will be determined approximately as follows:

<table>
<thead>
<tr>
<th></th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Midterms</td>
<td>25%</td>
</tr>
<tr>
<td>Exam</td>
<td>25%</td>
</tr>
<tr>
<td>Assignments</td>
<td>40%</td>
</tr>
<tr>
<td>Quizzes/Attendance</td>
<td>10%</td>
</tr>
</tbody>
</table>

Midterms: Average of the two midterm exams. The midterm exam average will be tentatively weighted 60% of the best grade and 40% of the lower grade.

Exam: Final exam

Assignments: Several Programming Assignments/Labs and Problem Sets

Grading (except for the final exam) is done by a number of class graders, under the direct supervision of the Teaching Fellow and the professor. If you have an issue with a grade (homework or exam), please contact the Teaching Fellow. Only if the issue is not resolved to your satisfaction, please contact the professor. Grades must be appealed within two weeks of receipt.

Midterms and Exam

There will be two midterm exams and one final exam, which will include all material covered from the beginning of the semester until the day of the exam. All exams will be closed books. You will be permitted to bring one 8.5 by 11 inch sheet of notes with you to the exams. There will be absolutely no make-up exams, except for medical emergencies. In that case, blue slips from Health Services will not be accepted; you must justify your medical problem with a letter from a doctor, specifying the period of time during which you were unable to attend one of the exams.

The two seventy-five minute in-class midterms held during the semester on Oct 1 and Nov 5. These dates are not flexible. The final will be held during the assigned exam slot. Please plan your work and travel plans at the end of the semester accordingly.

If you are found consulting any other material, than provided or specified, or electronic devices (including answering calls or texts) it will be considered as possible academic misconduct. If you need to leave the room for any reason including the restroom you are expected to ask for permission and you will not be allowed to return. Please plan appropriately.

Assignments

A large fraction of the course evaluation is based on several major programming assignments in C or assembly language. We will also post “pencil and paper” problem sets to help prepare you for
quizzes and exams. Problem sets may also involve smaller scale programming exercises, e.g., write or debug small programs, etc.

Assignments will be available on-line. **All assignments will be submitted electronically** **NO hard copies will be accepted.** Check for assignments regularly and start early! Bring problems, concerns and request for help to the UA, TF and professor as early as possible. The UA, TF and professor are available and want to help you but it is up to you to make use of us.

**Submission**

You will be required to submit all assignments, both problem sets and programming assignments electronically. In the case of hand written material you are expected to create a electronic copy by scanner. If you do not have access to one you may ask the CS office staff for direction. Additional information on what and how to complete the electronic submission can be found on the course web page.

**Late Policy**

Each assignment will have a due date. If late submissions are allowed for an assignment, there will be 10% penalty per day for late submissions. But, no late assignments will be accepted after one week from due date, and the last day to submit any late assignments is December 10, 2013. Extensions may be granted only for religious holidays and certified medical reasons.

No incompletes will be given, except for reasons of dire illness shortly before the end of the course, and only if a significant amount of work has been completed (e.g., attending lectures, handing in most assignments, and attending the midterms).

**Problem Sets**

At various points in the term you will be assigned problem sets. These problem sets are to confirm your knowledge of the material. In general they are good practice for the material, and its format, to be covered on the midterms and exams. These will be ungraded but solutions will be posted.

**Programming Assignments**

All necessary tools and software for completing the assignments are available on the CS machines – 32-bit Intel-compatible Linux machines: csa2, csa3. These systems will be used to grade your programming/lab assignments. You may access these machines from the undergraduate lab systems. The TF will provide information and help with accessing the systems. Although you may use your own machine, it is your responsibility to ultimately port your assignment to our CS machines to make sure they are graded correctly.

Again we will use the CS machines – 32-bit Intel-compatible Linux machines: csa2, csa3 – to grade your programming/lab assignments. **Although you may use your own machine, it is your responsibility to ultimately port your assignment to our CS machines to make sure they are graded correctly.**

Each assignment will specify what you need to submit and how it will be graded. If you are unclear what is required for a particular assignment or its grading consult the TF well before the due date.
Quizzes

Short quizzes will be given on a regular basis in discussion session. Make sure you are doing the readings on time and are prepared.

Office hours

The professor, TF and UA will hold office hours. The purpose of the office hours of the professor, TF and UA is to answer specific questions or clarify specific issues. Office hours are not to be used to fill you in on a class you skipped or to explain entire topics. Please come to class and to your discussion sessions.

To reach the professor, TF and UA at times other than office hours, please use piazza.

Teaching Fellow, Undergraduate Assistant and Discussions

Students are expected attend the weekly discussion section that they have been assigned to. The Teaching Fellow will lead the discussion sessions. The objectives are: to present material on “C” programming, reinforce the concepts covered in the lectures, and answer questions (or provide clarifications) regarding the homework and programming/lab assignments. The Teaching Fellow will also maintain a web page http://cs-people.bu.edu/kzhao.

In addition to the discussion the Teaching Fellow will hold her office hours in the http://www.cs.bu.edu/labs/ undergraduate lab EMA302.

The Undergraduate Assistant will additionally hold office hours. As need arises he will hold special focused session to help with particular topics. These sessions will be announced on piazza.

Tutoring

In addition to the discussion sessions, UA and TF office hours and the professor’s office hours you may also get extra help during the TF tutoring hours scheduled in the CS lab http://www.bu.edu/cs/resources/tutoring/. Note that terminal assistants in the CS lab are not supposed to help with course material, but to maintain the lab environment.

Topics

The course is broken down in two 3 major parts:

1. Fundamentals of Representing and Manipulating Information
2. Memory and Address Spaces
3. Input and Output

A rough outline of the topics that will be covered is presented below. The exact topics and order will be adjusted as needed. In general you will find equivalents of these topics in the text (CSAPP). The rough correspondence is as follows:
Part I: Chapters 2 & 3

Part II: Chapters 6 & 9

Part III: Chapters 7 & 10

Lectures, however, will not be restricted to text material or presentation. Lectures may cover additional or alternative material. **You will be responsible for all material covered in the lectures.**

Discussions will cover aspects of “C” programming. This will include material from the K&R text as well as material from the CSAPP that you will be responsible for.

**Part I: Fundamentals of Representing and Manipulating Information**

1. The Bit, Boolean Algebra/Logic, and bit manipulation and logic in “C”
2. Bytes and Hexadecimal Notation
3. Fundamental Data Types (multi-byte units directly supported by the hardware):
   (a) Fixed Width Integers: Sizes, Endianess, Alignment, Signed vs Unsigned, Arithmetic, Overflow and Underflow, Casting, and Promotion/Demotion
   (b) Floating Point: Basics and Dangers
4. Representing Programs:
   (a) Instructions: opcodes, operands, addressing modes, memory format and CPU resources/anatomy
   (b) Variables: Registers, Memory: explicit globals and implicit locals
   (c) Flow Control: Conditionals and jumps/gotos
   (d) Functions/Procedures: stack based changes in flow, arguments/calling conventions
   (e) Aggregate Data Types: Arrays, Structures, and Classes
   (f) Language Level Virtual Machines and Interpreters (eg. Java and Python).

**Part II: Memory and Address Spaces**

1. Caches
   (a) Locality
   (b) Organization and Associativity
   (c) Read and Write Access
   (d) Cache Friendly Code
2. Virtual Memory
   (a) Physical vs Virtual – Address Spaces
   (b) Paging and Memory Mappings
   (c) Address Translation and TLB’s
3. Address Space Layout: text, stack, heap
4. Heap Management: Memory Allocators and Garbage Collectors
5. Linking

Part III: Input and Output
1. Operating Systems, Devices and Device Drivers
2. Programmed and Interrupt Driven I/O
3. Read and Write File Interfaces
4. Stream vs Block oriented I/O
5. Memory Mapped I/O
6. Event Driven/Asynchronous I/O

Academic Conduct

Academic standards and the code of academic conduct are taken very seriously at our university. Please take the time to review the CAS Academic Conduct Code if you are unfamiliar with the contents. See the [http://www.bu.edu/academics/resources/academic-conduct-code/](http://www.bu.edu/academics/resources/academic-conduct-code/) for the CAS Academic Conduct Code, in particular regarding plagiarism and cheating on exams. Copies of the CAS Academic Conduct Code are also available in room CAS 105. A student suspected to violate this code will be reported to the Academic Conduct Committee, and if found culpable, the student will receive a grade of "F" for the course.

Assignments must be completed individually. Discussion of issues in computer systems is encouraged, but representing the work of another person as your own is expressly forbidden. This includes "borrowing", "stealing", copying programs/solutions or parts of them from others. We may use an automated plagiarism checker. Cheating will not be tolerated under any circumstances. Handing in your own work a day or two late will affect your grade far less than turning in a copy of someone else’s work on time!

Any resources, including material from other students (current or past), that are used, beyond the text or that provided by the TF or professor must be clearly acknowledged and attributed. Using such material may at the discretion of the TF or professor result in a lower grade. However, if such material is used and not acknowledged and attributed, it will automatically be considered as possible academic misconduct.
Signature

The following signature confirms that I have read and understood the material in this syllabus.

Full Name: ____________________________________________

Date: __________________