Syllabus: CSE 220: Systems Programming

Department of Computer Science and Engineering
University at Buffalo

Karthik Dantu

Fall 2019

All students are expected to read and understand this syllabus. Failure to adhere to the policies in this syllabus may have consequences, including a negative impact on student grades, failure in the course, or administrative action against the student. *It is your responsibility to ask questions if anything in this document is unclear to you.*

This semester, we will have two separate sections being taught by two instructors. This is the syllabus for Dr. Dantu's section.

Instructor

Karthik Dantu
Davis 331
kdantu@buffalo.edu
Office hrs: MW 12:30-2 PM

0.1 Teaching Assistants

TA names and their office hours can be found on the course website for Dr. Blanton's section. https://cse.buffalo.edu/ eblanton/course/cse220/

1 Course Web Site

https://droneslab.github.io/cse220/

2 Lectures and Labs

Lectures will be held at **Knox 109** on **MWF 15:00-15:50 PM**.

Labs are at various times, as follows; every student is registered for particular lab section, and must attend their registered lab section unless prior arrangements have been made with the instructor. Lecture and lab times and locations are not scheduled by the course instructor and may be changed by the University for various reasons. Check HUB for the most recent information.

Labs are common to students from both sections, and you can register to any of the labs that are convenient to you based on your schedule.

Lab Section	Time and Day	Location
A1	Mo 15:00-17:00	Bell 340
A2	We 16:00-18:00	Bell 340
A3	Sa 08:00-10:00	Bell 340
A4	Fr 14:00-16:00	Bell 340
A5	Fr 16:00-18:00	Bell 340
A6	Fr 18:00-20:00	Bell 340
A8	Sa 16:00-18:00	Bell 340

3 Course Description

CSE 220 is a 4 credit course.

> This course is an introductory course on computer systems. It introduces computer systems from a programmer's perspective, rather than a system implementer's perspective, which prepares students for more advanced topics that discuss the internals of a computer system (e.g., operating systems or computer architecture). As a result, the focus of the course is teaching programmable interfaces of a computer system as well as how to use them correctly and effectively when writing a program. The topics mainly include hardware/software interfaces (e.g., data representation in memory) and OS/application interfaces (e.g., syscalls). In discussing these topics, the course gives an overview of a complete computer system, the hardware, operating system, compiler, and network, in order to guide students through various components that modern programs rely on to accomplish their intended purposes.

4 Course Materials

There are two required texts for this course. Students are expected to have immediate access to both of these texts throughout the semester.

Title	Authors	ISBN
Computer Systems: A Programmer's Perspective (Third Edition)	Bryant and O'Hallaron	978-0-13-409266-9
The C Programming Language (Second Edition)	Kernighan and Ritchie	978-0-13-110362-8

Lecture slides will be provided electronically throughout the course of the semester.

Various readings from external sources may be assigned, in which case they will be provided or available through University resources (*e.g.*, the library or periodical subscriptions).

References to materials not required or assigned may be made, and students are encouraged to follow up on these references, but this will not be required for successful completion of the course.

5 Communication

All electronic communication regarding this course must occur in one of two ways:

- Email using your *official UB email account*, or
- Messages on the course Piazza instance.

For topics of a sensitive nature, please email the course instructor directly from your *University-supplied email address*. Emails from non-University addresses will be disregarded due to privacy concerns and FERPA regulation. For all other contacts, please do **NOT** email the course instructors directly; instead, make a private or public post to Piazza, as appropriate. Private posts of non-sensitive nature should be sent to *all course staff*. This will ensure the most timely possible response. For matters regarding a specific lab section, contact the lab assistants first, as they may be able to help resolve your issue without involving the rest of the course staff.

Students are expected to monitor the course Piazza instance routinely, as important course announcements will be posted to Piazza. Schedule changes, lab and assignment handouts, homeworks, required readings, and other materials may be posted to Piazza, and it is the student's responsibility to keep track of these things. Failure to read Piazza messages will not be accepted as an excuse for missed projects, labs, exams, or other course requirements.

Students will be added to the Piazza instance by the course instructor. If you are not, please contact the instructor by email to rectify this.

6 Prerequisites

- CSE 116. SEAS approved or Intended Majors, CS Minors.

Students are expected to have a firm understanding of the material from CSE 116 at the University at Buffalo and its prerequisites.

7 Course Requirements

The following items are required of every student, and failure to complete them may affect student grades as described in Section 9, *Grading Policy*, below.

7.1 General

Attendance in this course is **mandatory**, but will not necessarily be directly tracked. Students are responsible for attending every lecture and every meeting of their lab section. It will be assumed that students are familiar with all material presented in class, and any material presented in lecture or lab may appear on any test, quiz, homework assignment, or other evaluation. *Attendance and attention to lecture and lab materials are critical to success in this course.*

A quiz evaluating students' understanding of the University and Department academic integrity policies must be completed by all students, and all students must achieve 100% accuracy on this quiz.

7.2 Assignments

Several programming assignments will be required of all students. Programming assignments are intended to improve student understanding of the course material as well as demonstrate student mastery of certain core concepts.

Lab activities will be provided every week, which will range from small programming assignments to written activities to group discussions and beyond. Students are expected to participate fully in lab and will be evaluated accordingly. Course staff will be available in labs to assist students, guide tasks, and present additional course material.

Programming assignments and lab activities, unless otherwise explicitly specified, are *individual projects*, and collaboration between students to complete any such assignment is a violation of the course academic integrity policy.

Written Homework may be assigned, and students are expected to complete any such assignments in a timely fashion, although they will not be graded. These written assignments are intended to allow students to self-evaluate their level of preparedness and mastery of the course material, and students are encouraged to seek assistance from the instructor, teaching assistants, or each other in understanding and completing written assignments.

Readings may be assigned from time to time, and students are expected to complete these readings, although there may be no direct evaluation of any given reading. Readings are selected to improve student understanding of the course material and/or present auxiliary material that the instructor believes is relevant and important. Material from readings may appear directly or indirectly in assignments, on quizzes or on exams.

7.3 Tests and Quizzes

Quizzes may be introduced at any time by the instructor, covering any material previously covered in lectures, readings, or written homework assignments. These quizzes may or may not be announced in advance. (In particular, "pop quizzes" may be utilized to evaluate student attendance, engagement, and present understanding of course material.)

There will be four Lab Exams. Each lab exam will be given in lab, and must be completed during each student's assigned lab section. Lab exams will assess mastery of programming topics that have already been covered in class and that have been used in programming assignments. Lab exams may cover programming techniques or knowledge from any assignment previously submitted or currently outstanding, as well as any material covered in lecture to date.

There will be one midterm and one final exam. The midterm exam will cover all material presented in the course to date, including: lectures, labs, programming assignments, written homework assignments, and assigned readings. The final exam will cover all material covered in the course for the duration of the semester, including: lectures, labs, programming assignments, written homework assignments, and assigned readings.

7.4 Submission Policy

Programming assignments will be assigned with a deadline. All assignments are to be submitted by this deadline. In the event of any ambiguity in the deadline, times are assumed to be in the *current local time zone of the University.* Penalties for missing this deadline are as follows.

- Projects submitted before the deadline will incur no penalty.
- Projects submitted after the deadline, but within 24 hours of the deadline (excluding Saturday, Sunday, and University holidays) will incur a 20% penalty.
- Projects submitted more than 24 hours after the deadline as described above will not be accepted and will receive no credit.

Neither the instructor nor the teaching assistants will provide assistance for programming assignments after the assigned deadline.

Lab activities other than Lab 1 are due 36 hours after the start of lab. Lab 1 is due at 11:59 PM on the second Tuesday of classes (due to add/drop). No late submissions for lab activities will be accepted.

7.5 Programming Assignment Re-grading Policy

If you believe that a programming assignment has been graded incorrectly, you may submit it for re-grading. A request for a re-grade must be submitted within one calendar week of receiving the grade for a project, and must include:

- A copy of the score sheet for the assignment
- A description of the specific error in grading that is being contested
- Relevant code demonstrating the submitted code's correctness or the grading script's incorrectness, if available

Re-grading of programming assignments is intended *only* to address errors in grading. No grades will be improved for any other reason, although they may be reduced; in particular, note that *your grade on any part of the assignment, not just the portion being re-graded, may be reduced* if re-grading discovers additional errors. This includes automated evaluations that passed because they did not trigger bugs that were discovered in manual evaluation for the re-grade.

7.6 Exam and Quiz Re-grading Policy

If you believe that an exam or quiz has been graded incorrectly, you may submit it for re-grading. A request for a re-grade must be submitted within one calendar week of the exam or quiz being returned to you, must be submitted in writing to the instructor, and must include:

- The original, unmodified, exam or quiz
- A clear statement of the error *on a separate sheet of paper*

Re-grading of exams and quizzes is intended *only* to address errors in grading. No grades will be improved for any other reason, although they may be reduced if *errors are found in any portion of the assignment, not just the portion being re-graded.* Using re-grading as a bargaining tool to increase your score is likely to result in a lower grade, as the exam will be scrutinized in detail for errors that may have been missed the first time.

7.7 Make-up Policy

No deadline extensions or make-up work will be permitted except for approved University absences. Please see the University attendance policy for more information.

No make-up exams will be given whatsoever except for **documented extreme circumstances**. *24 hours of advance notice via e-mail or telephone contact (voice mail is permitted) must be provided if at all possible* before missing an exam session. If advance notice is not possible, documentation supporting this must be provided. Absence from an exam session due to illness **must** be supported by a note from a physician specifying that the student was too ill and/or contagious to attend on the exam date.

You are responsible for remembering and attending exam sessions. Please use extra assistance to remind yourself if necessary.

8 Course Schedule

The course schedule, including exam dates and times, is provided here for convenience. Note that course staff do not schedule most of the following items, including final exam time and location. *You are responsible for verifying your final exam time and location on HUB.* Inclement weather, local emergencies, unsafe building and/or campus environments, or other circumstances may cause the University to change this schedule. Course progress and pedagogical concerns may cause rescheduling of other exams and activities. You will be notified via Piazza or UB Email of changes within the control of course staff. Course staff will attempt to keep you appraised of changes outside of staff control, but you are responsible for monitoring University communications to this effect.

The *weekly course schedule is tentative*, and will be adapted based on class progress and feedback from students from both sections. The instructors reserve the right to change the schedule as they see fit. Please follow the announcements on Piazza to be up-to-date on the schedule.

You are responsible for monitoring any changes to this schedule, according to communications from course staff or the University. Failure to be aware of schedule changes is not sufficient reason for extended deadlines, make-up exams, or other accommodations.

As shorthand, the C Programming Language textbook by Kernighan and Ritchie is called *KR* and the textbook by Bryant and O'Hallaron is called *BO* in the table below.

Date	**Description**	**Textbook**
8/26/19	Introduction	
8/28/19	Introduction to C	KR 1.1
8/30/19	C - Variables and Loops	KR 1.2, 1.3
9/2/19	Labor Day	
9/4/19	C - IO, Arrays and Scoping	KR 1.5-1.9
9/6/19	C - Control flow	KR 3.1 - 3.7
Fri, Sep 6, 2019	Programming Assignment 0 is due	
Week of Sep 9, 2019	Lab Exam 1 (in-lab)	
9/9/19	C - Pointers	KR 5.1 - 5.6
9/11/19	C - Best practices	
9/13/19	Tour of Computer Systems	BO 1.1-1.7
9/16/19	Integers and Integer representation	BO 2.1-2.2
9/18/19	Integer Arithmetic	BO 2.3
9/20/19	Floating Point	BO 2.4
Fri, Sep 20, 2019	Programming Assignment 1 is due	
9/23/19	Machine level representation - history and machine level code	BO 3.1-3.2
9/25/19	Data movement and Stack	BO 3.3-3.4
9/27/19	Arithmetic instructions and control flow	BO 3.5-3.6
Week of Sep 30, 2019	Lab Exam 2 (in-lab)	
9/30/19	Procedures	BO 3.7
10/2/19	Compiler basics	BO 5.1
10/4/19	Compiler Optimization	BO 5.4 - 5.6
Fri, Oct 4, 2019	Programming Assignment 2 is due	
10/7/19	Program Optimization	5.12-5.14
Wed, Oct 9, 2019	Midterm Examination (in-class)	
10/11/19	Midterm review/C best practices	

Date	Description	Textbook
10/14/19	Memory technologies and locality	BO 6.1-6.2
10/16/19	Memory hierarchy and caches	BO 6.3-6.4
10/18/19	Program performance	BO 6.5-6.6
10/21/19	Address spaces	BO 9.1-9.2
10/23/19	VM for memory management	BO 9.4
10/25/19	VM for memory protection	BO 9.5
Fri, Oct 25, 2019	Programming Assignment 3 is due	
Week of Oct 28, 2019	Lab Exam 3 (in-lab)	
10/28/19	Dynamic Memory Allocation	BO 9.9
10/30/19	Address translation	BO 9.4
11/1/19	Common memory bugs	BO 9.11
11/4/19	Exceptions	BO 8.1
11/6/19	Processes and process control	BO 8.2,8.4
11/8/19	Signals	BO 8.5
11/11/19	System-level I/O	BO 10
11/13/19	Concurrency with processes	BO 12.1
11/15/19	Concurrency with threads	BO 12.3
Fri, Nov 15, 2019	Programming Assignment 4 is due	
11/18/19	Concurrency and I/O	BO 12.2
11/20/19	Shared variables and synchronization	BO 12.4, 12.5
11/22/19	Threads and Parallelism	BO 12.6
11/25/19	Network Programming - client-server	BO 11.1, 11.2
Wed, Nov 27, 2019– Sat, Nov 30, 2019	Fall Recess	
Week of Dec 2, 2019	Lab Exam 4 (in-lab)	
12/2/19	Improving program performance examples	
12/4/19	Improving program performance examples	
12/6/19	Improving program performance examples	
Fri, Dec 6, 2019	Programming Assignment 5 is due	
Fri, Dec 6, 2019	Last day of regular classes	
Wed, Dec 11, 2019	Final exam	

9 Grading Policy

No "I" (Incomplete) grades will be given for this course except for **documented extreme circumstances** or situations required by University policy. *Failure to complete work on time does not constitute an extreme circumstance.*

Grades will not be changed at the end of the semester for any reason other than a documented error in grading according to the policies outlined in Section 7.5 and Section 7.6. No grade negotiation will be permitted. In particular, no grades will be changed to preserve scholarships, fellowships, University positions, immigration status, internship or job offers, or any other outside factor. Grades reflect student performance and mastery of course material.

The credit breakdown for the course will be as follows:

Course Requirement	Course Grade	Percent
Less than 100% on Academic Integrity quiz	F	*N/A*
Less than 100% on Lab 01	F	*N/A*
Less than 60% average on Exams & Lab Exams	F	*N/A*
Quizzes	1 Letter Grade	*N/A*
Programming Assignment 0		5%
Programming Assignment 1		5%
Programming Assignment 2		5%
Programming Assignment 3		5%
Programming Assignment 4		10%
Programming Assignment 5		5%
Lab Activities		10%
Lab Exams		20%
Midterm Exam		15%
Final Exam		25%

Note that this adds to 105%. This means that students who attend every lab and turn in every assignment have the opportunity to receive up to 5% extra credit in the course.

Failure to complete the Academic Integrity quiz with complete correctness (100% credit) or failure to complete Lab 01 with complete correctness (100% credit) will result in failure of the course. You may submit both of these assignments as many times as required to achieve complete correctness.

Failure to achieve a 60% or better average across all exams and lab exams will result in failure of the course. This is an *un-weighted average*; the final, midterm, and each of the four lab exams are weighted equally for this condition.

Quizzes, if any, will not be assigned a percentage of the final grade, but poor performance may detriment your final grade by (at most) one full letter grade. (That is, an A- would become B-, a B- would become C-, *etc.*)

Final grades will be assigned from the above percentages as follows, although a curve of the instructor's choice may be applied if the instructor deems it warranted. Lower percentages are inclusive, upper percentages (excepting 100%) are not; that is, a 90.0% would be an A-, not a B+.

A	95-+ %
A-	90-95%
B+	87-90%
B	83-87%
B-	80-83%
C+	77-80%
C	73-77%
C-	70-73%
D+	67-70%
D	63-67%
F	0-63%

10 Academic Integrity

Students will abide by the CSE Academic Integrity Policy, the University Academic Integrity Policy, and the Undergraduate or Graduate amendments thereof, as appropriate.

All resources used in completing assignments for this class *must be given appropriate attribution*,

and the *only resources allowed for the completion of programming assignments without specific permission* are as follows.

- The required course textbook *Computer Systems: A Programmer's Perspective*, Third Edition, by Bryant and O'Hallaron
- The required course textbook *The C Programming Language,* by Kernighan and Ritchie
- Lecture material from this course
- Required or recommended readings from lecture material
- Man and info pages from the course virtual machine

In particular, Stack Exchange, code from other students in the course or students who have completed this course or related courses at other universities in previous semesters, GitHub repositories, code or algorithms from other web sites or books, and other resources are not allowed without explicit permission from the instructor.

If there is any question about whether a resource is acceptable for use in completing a course assignment, students are encouraged to ask the instructor or a TA *before* making use of it. Asking about a resource is **not** a violation of academic integrity, even if the resource is not allowed for the course.

Violation of these policies will result in a failing grade for the course and referral upward for additional sanctions according to University policy.

10.1 Academic Integrity Amnesty

A student who has committed a violation of this academic integrity policy may receive limited amnesty for the violation by *notifying me, in writing,* of the violation **before I have begun to assess the violating assignment**. This notification must include the student's name, person number, UBITname, and state the assignment in question and the nature of the violation. Upon submitting such a statement, the student will receive no credit for the violating assignment, but *no further sanctions will be taken, and the violation will not be reported.* Once I have begun assessing the assignment in question, no such statements will be permitted. Since it may not be obvious to students when assessment begins, such statements should be submitted as soon as possible after the violation occurs. While assessment may begin at any time, in general I will not look at student submissions until a project deadline has passed.

See my online Academic Integrity Policy for an example scenario and more information.

11 Classroom Expectations

Students are expected to behave in a way that is respectful to their fellow students and the course staff. The University at Buffalo has a list of behavioral expectations that includes:

- Attending classes and paying attention. Students should not ask an instructor in class to go over material they missed by skipping a class or not concentrating.
- Not coming to class late or leaving early. If a student has to enter a class late, he or she should do so quietly and should not disrupt the class by walking between the class and the instructor. Students should not leave class unless it is an absolute necessity.

- Not talking with other classmates while the instructor or another student is speaking. If a student has a question or comment, he or she should raise a hand, rather than starting a conversation about it with a neighbor.
- Showing respect and concern for others by not monopolizing class discussion. Students must allow others time to give their input and ask questions. Students should not stray from the topic of class discussion.
- Not eating and drinking during class time.
- Turning off electronic devices including cell phones, pagers, and beeper watches.
- Avoiding audible and visible signs of restlessness. These are both rude and disruptive to the rest of the class.
- Focusing on class material during class time. Sleeping, talking to others, doing work for another class, reading the newspaper, checking email, and exploring the Internet are unacceptable and can be disruptive.
- Not packing bookbags or backpacks to leave until the instructor has dismissed class.

In addition to this list, students should minimize their use of laptop computers, tablets, and smart devices. Note-taking and the use of assistive technologies are appropriate, but web browsing, watching videos, chat programs, *etc.*, are inappropriate behavior for the classroom.

12 Program Outcomes and Competencies

This course is required in both the BS Computer Engineering program, accredited by the Engineering Accreditation Commission (EAC) of ABET, and the BS Computer Science program, accredited by the Computing Accreditation Commission (CAC) of ABET.

The course introduces students to the following CAC student outcomes, for which graduating students must demonstrate:

(CAC-1) Analyze a complex computing problem and apply principles of computing and other relevant disciplines to identify solutions.

(CAC-2) Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.

(CAC-6) Apply computer science theory and software development fundamentals to produce computing-based solutions.

The course introduces students to the following EAC student outcomes, for which graduating students must demonstrate:

(EAC-1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics

(EAC-7) an ability to acquire and apply new knowledge as needed, using appropriate learning strategies

Student outcomes will be evaluated as follows.

CAC 1	CAC 2	CAC 6	EAC 1	EAC 7	Assessment Types
✓	✓	✓	✓	✓	Programming Assignments
✓	✓	✓	✓	✓	Lab Activities
✓			✓		Exams

13 Accessibility Resources

From the UB Reasonable Accomodation Policy:

> The University at Buffalo is committed to providing equal access to individuals with disabilities, including physical access to programs and reasonable accommodations for members of the university community.

The UB Accessibility Resources Office provides assistance for students who require reasonable accommodations due to disability. They may be found at 60 Capen Hall or contacted by phone at 716-645-2608. Students must register with their office to receive accommodations for physical or learning disabilities.

14 Critical Campus Resources

Sexual Violence: UB is committed to providing a safe learning environment free of all forms of discrimination and sexual harassment, including sexual assault, domestic and dating violence and stalking. If you have experienced gender-based violence (intimate partner violence, attempted or completed sexual assault, harassment, coercion, stalking, etc.), UB has resources to help. This includes academic accommodations, health and counseling services, housing accommodations, helping with legal protective orders, and assistance with reporting the incident to police or other UB officials if you so choose. Please contact UB's Title IX Coordinator at 716-645-2266 for more information. For confidential assistance, you may also contact a Crisis Services Campus Advocate at 716-796-4399.

Mental Health: As a student you may experience a range of issues that can cause barriers to learning or reduce your ability to participate in daily activities. These might include strained relationships, anxiety, high levels of stress, alcohol/drug problems, feeling down, health concerns, or unwanted sexual experiences. Counseling, Health Services, and Health Promotion are here to help with these or other issues you may experience. You can learn more about these programs and services by contacting:

- Counseling Services:
  - 120 Richmond Quad (North Campus), 716-645-2720
  - 202 Michael Hall (South Campus), 716-829-5800
- Health Services:
  - Michael Hall (South Campus), 716-829-3316
- Health Promotion:
  - 114 Student Union (North Campus), 716-645-2837

Acknowledgments

Some language in this syllabus is drawn from University policies (as noted), the UB Course Syllabi Requirements document, department guidelines, and other University resources. Some language and structure in this syllabus is drawn from Steve Ko's CSE 486/586 syllabus from Spring 2017 and from Matthew Hertz's CSE 115/503 syllabus from Spring 2019.