CS 5600 Computer Systems
Spring 2026 – The Roux Institute at Northeastern University

Instructor

Dr. Scott Valcourt	s.valcourt@northeastern.edu	603-380-2860 (mobile)

Class Meetings: Room 136, Tuesdays 9:00am-12:20pm ET

Office Hours: Wednesdays 11:00am-12:00pm ET
By Appointment via Email or Text
https://northeastern.zoom.us/j/91454590124

Teaching Assistants

		Office Hours (ET)
Krunal Ghanshyambhai Savaj	savaj.k@northeastern.edu	TBD
	https://northeastern.zoom.us/j/?	

Syllabus

Course Description

Operating systems are one of the foundations upon which the remainder of computer science rests. Very rarely do we think twice about starting new processes or threads, dealing with the complexities of memory management, or the vagaries of low-level hardware interactions. Instead, the operating system abstracts these complexities away, leaving us free to work at higher levels of the software stack.

Understanding the inner workings of operating systems is a fundamental skill for computer scientists. Whether you are interested in low-level system design, cloud computing, or application development, the principals of operating system design can help you understand the behavior of computers and build more performant applications. Furthermore, knowledge of threads, synchronization, and memory management are key for building the large-scale, distributed, data-centric applications of tomorrow.

This course will cover all the basics of operating systems: program loading, context switching, threads and synchronization, virtual memory, and block devices and file systems. We will also cover more advanced topics, such as operating system security and virtualization. This course will be project centric, with students building significant components of an operating system from scratch. This course studies the structure, components, design, implementation, and internal operation of computer systems, focusing mainly on the operating system level. Reviews computer hardware and architecture including the arithmetic and logic unit, and the control unit. Covers current operating system components and construction techniques including the memory and memory controller, I/O device management, device drivers, memory management, file system structures, and the user interface. Introduces distributed operating systems. Discusses issues arising from concurrency and distribution, such as scheduling of concurrent processes, inter-process communication and synchronization, resource sharing and allocation,

and deadlock management and resolution. Includes examples from real operating systems. Exposes students to the system concepts through programming exercises.

To learn these concepts, the instructor will employ the following methods known to be effective in transferring knowledge:

Lectures	will focus on developing a conceptual understanding of systems concepts.
Homeworks	will ask students to apply their conceptual knowledge via problems, theory, and code implementation.
Quizzes	will offer students the opportunity to verify their skills learned in each module with the option of retaking each quiz until mastery (85%+) is achieved.
Programming	will allow students to collaboratively develop coded solutions to typical programming problems that are found in operating systems studied in the theory via a programmatic user interface.
Exams	will demonstrate to the student and instructor to what level a student has mastered the material presented during the course that assesses the course learning outcomes have been achieved.

Learning Goals

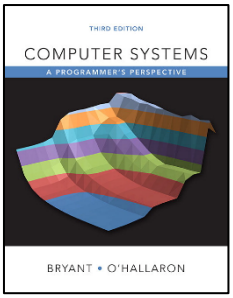
At the end of CS 5600, a student should be able to do the following:

1. Operating Systems (OS) Basics – trace the operation of context switching and program loading through the completion of a programming assignment demonstrating understanding of these principles;
2. Synchronization – use synchronization primitives (mutexes, condition variables) to synchronize threads as demonstrated in a programming assignment;
3. Virtual Memory – describe the operation of page tables and OS page faulting mechanisms for implementing demand allocation, demand loading, and copy-on-write;
4. Block Storage – understand performance characteristics of hard drives, demonstrate knowledge of RAID configurations via a programming assignment;
5. File Systems – describe the operation of journaled and log-structured file systems, demonstrate knowledge by writing a simple FUSE-based file system for Linux;
6. Security – recognize permissions and access control lists as implemented in Linux and Windows, describe common software exploits; and
7. Virtual Machines – describe and use the three common virtualization mechanisms in use.

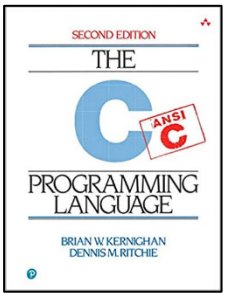
Required and Recommended Textbooks

- **Required: Operating Systems, Three Easy Pieces**
  - Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
  - Freely Available on the Web
  - https://pages.cs.wisc.edu/~remzi/OSTEP/

- **Recommended: Computer Systems: A Programmers Perspective**
  - Randal E. Bryant and David R. O'Hallaron, 2016, Third Edition
  - Pearson Publishing, ISBN 978-0-134-09266-9
  - https://www.pearson.com/us/higher-education/program/Bryant-Computer-Systems-A-Programmer-s-Perspective-3rd-Edition/PGM2476825.html



- **Recommended: C Programming Language**
  - Brian W. Kernighan and Dennis M. Ritchie, 2016, Second Edition
  - Pearson Publishing, ISBN 978-0-131-10362-7
  - https://www.amazon.com/Programming-Language-2nd-Brian-Kernighan/dp/0131103628



Additional Open Education Resources (OER) will be included as documents to be referenced or read as the semester progresses.

Grading

The goal of this course is to teach you the material to a certain level of mastery. Quizzes will provide you with rapid and frequent feedback about how well you understand the most recent topic. The homework assignments will give you practice with detailed problems associated with the material covered in class and related to the textbook readings. The programming assignments, which we will detail in depth during class sessions, will offer you the opportunity to apply your skills in a structured way to bring meaning to the theory studied in class.

Assessment will be based on programming assignments, weekly homework assignments, learning assessments (quizzes), and a course project and a final exam. Attendance and participation will not factor into your final grade. You are responsible for your own actions and the consequences of not attending class.

Programming Assignments (4)	10% each for a total of 40%
Weekly Class Homeworks (10)	1% each for a total of 10%
Quizzes (10)	1% each for a total of 10%
Course Project	20%
Final Exam	20%

Programming Assignments, Homeworks, and Quizzes will be graded and returned with feedback.

All Assignments will be graded out of 100 points. An ‘A’ grade requires clearly offered answers that indicate depth of understanding of the material in the assignment. A ‘C’ grade is presented when the assignment is completed with technically correct answers, but not much more is offered. A ‘F’ grade is earned when the assignment has too many flaws or blank responses to know if the material learning has been achieved.

Quizzes will provide you with rapid and frequent feedback about how well you understand the most recent topic. Students will be allowed to retake the quizzes as many times as required in the allocated window each week until mastery is achieved (a score of 85 or better).

The numerical to letter grade conversion for the final course grade will correspond to the following chart where shaded scores are considered passing within the graduate program:

Letter	Score		Letter	Score		Letter	Score
A	93-100		A-	90-92		B+	87-89
B	83-86		B-	80-82		C+	77-79
C	73-76		C-	70-72		D+	67-69
D	63-66		D-	60-62		F	Below 60

Additional Note: Canvas does not calculate the final course grade effectively in the same manner as the syllabus lists. The best way to determine your final grade is to follow the same process as the instructor: download all your grades, categorize all the grades together, calculate each category's average score, multiply all the category averages by the percentage of that category, then add all the category scores together to achieve a final grade.

Late Policy

Weekly homework assignments will be available by the end of the class meeting each week (~9pm Wednesdays) and Homework Assignments will be due by 11:59pm ET on the following Monday, offering nearly 12 days to complete. You may submit any assignment up to a week after the regular due date but may be subject to a late penalty of 10 points per day, meaning a perfect assignment (score of 100) submitted four days late would be subject to an actual score of 60 (4 days * 10 = 40 points).

During the semester, you have eight (8) no fault time travel days to use as you see fit. You may use your time travel days to hand in an assignment up to three days after the regular deadline as though you had handed it in on time. Note that you can use at most three days for any single assignment, and the time travel days are considered integers (no partial days). To use a time travel day, you must notify the professor before the regular deadline via email and include in your email at least one question about the assignment, justifying your use of time travel days.

Asking for Help

Computer Science, particularly programming in C, may be a brand-new topic for you if you are taking this course. One of the goals for this course is to show you that theoretical computer science concepts can be implemented in real code and can be fun. This course will be challenging, will stretch your understanding of concepts, can be difficult, and can be extremely satisfying when something works correctly. It should not be frustrating, because frustrating things are not challenging or fun.

Frustration occurs when you are working on something and not making progress, especially if you are not sure about what you are doing. Frustration does not facilitate learning. If you find

yourself getting frustrated, stop. Go do something else--eat some dinner, take a walk--but most importantly, ask a question. Send an email to the professor or a TA, ask a question in class, talk with a classmate, or set up a 1:1 session to go over your work.

We do not expect you to complete the assignments entirely on your own.

Please follow the 30-minute rule: if you have been stuck on a problem for more than 30 minutes and have made no progress, despite your best efforts, please stop and get help. Email the professor, ask a TA, or consult a peer. If you do not get an answer immediately, do something else for a while. Please do not waste your time on one problem or bug in your code. Asking a question can both get you past the bug quickly and teach you how to fix it on your own next time.

Respect for Student Backgrounds and Perspectives

It is my intent that students from different backgrounds and perspectives will be successful in this course, that each student's learning needs are addressed both in and out of class, and that the diversity that each student brings to this class is viewed as a resource, strength, and benefit. I expect you to feel challenged and sometimes outside of your comfort zone in this course, but it is my intent to present materials and activities that are inclusive and respectful of all persons, no matter their gender, sexual orientation, disability, age, socioeconomic status, ethnicity, race, culture, perspective, and other background characteristics. We should all strive for these principles both inside and outside of the classroom.

The course meeting time is listed on the front page of this document as well as detailed in the tables that follow in the course calendar. If a class meeting conflicts with your religious observances, please let me know in the first two weeks of the class so that we can make other arrangements. Northeastern University respects the religious practices of its students, faculty, and staff and is committed to ensuring that all students can observe their religious beliefs without academic penalty.

Class rosters are provided to each instructor with each student's legal name. I will gladly honor your request to address you by an alternate name and/or gender pronoun. Please advise me of this early in the semester so that I may make appropriate changes to my records.

Stuff Happens Clause

My goal for this course is for you to learn the computer science concepts you need to be successful. The guidelines and rules are intended to facilitate your learning, and there is no substitute for putting in the time and effort to learn the material.

Life happens, though, and sometimes it will prevent you from focusing on CS. If something happens, please communicate with the professor as soon as is feasible. I don’t need details, but we do need to develop an alternative plan that will work for you and still ensure you will learn what you need to learn. We are partners in this academic adventure, and I want you to be successful.

Academic Accommodations

If you have a documented need for academic accommodation, please contact the professor within the first two weeks so we can have a conversation about how best to make appropriate arrangements.

If you require support during the course due to a disability, please ensure that you are already registered with the University's Disability Center, and contact your course instructors to coordinate any support needed during the course.

Mental health issues are real and can prevent you from doing your best work. Your Khoury advisor is your primary contact for accessing University resources. Do not hesitate to make use of them as needed.

Collaboration and Academic Honesty

Computer Science, both academically and professionally, is a collaborative discipline. In any collaboration, however, all parties are expected to make their own contributions and to generously credit the contributions of others. In our class, therefore, collaboration on homework and programming assignments is encouraged, but you, as an individual, are responsible for understanding all the material in the assignment and doing your own work. Always strive to do your best, give generous credit to others, start early, and seek help early from both your professors, teaching assistants, and classmates.

The following rules are intended to help you get the most out of your education and to clarify the line between honest and dishonest work. The professor reserves the right to ask you to verbally explain the reasoning behind any answer or code that you turn in and to modify your project grade based on your answers. It is vitally important that you turn in work that is your own. Follow the guidelines for academic honesty or we are done.

If you have had a substantive discussion of any homework or programming solution with a classmate, then be sure to cite them in your report. If you are unsure of what constitutes "substantive", then ask us or err on the side of caution. You will not be penalized for working together. You must not copy answers or code from another student either by hand or electronically. Another way to think about it is that you should be talking English with one another, not C.

The following rules apply to anything you hand in for a grade:

- You may not copy anyone else's code under any circumstances. This includes any online code sources you may have found.
- You may not permit any other student to see any part of your program.
- You may not permit yourself to see any part of another student's program.
- You may not post a public question to Piazza or any other community site that contains any part of your code.

- You may consult online resources as part of your course work, but you may not copy code from online sources. If you get an idea of how to solve a problem from an online source, include a short citation in a code block at the top of your source code file.

The university's academic integrity policy discusses actions regarded as violations and consequences for students: https://osccr.sites.northeastern.edu/academic-integrity-policy/

Statement on the use of LLMs and Code Generation Tools

Large language models (LLM) and code generation tools such as ChatGPT, Claude, or co-Pilot can be useful for professional coders and for generating first drafts of documents. However, in my opinion they are not conducive to learning. The work and effort required to write explanations of a concept or implement an algorithm is critical exercise for your brain. You become a better coder and a better writer by doing it yourself, making mistakes, getting feedback, and fixing your mistakes. Mistakes are often the greatest teacher.

There are also several risks for you in using LLMs and code generators for coursework. The first is that if multiple people submit LLM generated text or code, it will often look virtually identical or contain unique turns of phrase or vocabulary, leading to serious questions of academic integrity and some potentially uncomfortable discussions with the professor. The second is that LLM generated text often contains slightly incorrect word choices or other small errors, which may end up costing you points you would have earned had you written it yourself. I find LLM generated answers also tend to be longer than they need to be and may not directly answer the question (which also loses points). The final point to consider is that when I am reading your work, I am looking for your voice. What do you bring to your writing or code that is uniquely yours? It's a little hard to describe, but when I'm reading 80 answers or explanations of the same concept, if there are a half-dozen answers that sound almost identical and all contain one or more of the same particularly smooth turns of phrase, I am disappointed, because I'm not convinced the work is their own. On the other hand, if your answer stands out as having a unique voice (even if it's not perfectly smooth) and is a concise response that directly answers the question or describes the concept, it makes me happy. If you want a good recommendation from me, make your voice unique and your own.

Tutoring and Workshops by Global Learner Support (GLS)

Global Learner Support (GLS) offers one-to-one tutorials for NU learners in the areas of academic writing, academic presentations, APA/MLA citation, English language conversation, and professional communication. To make a tutoring appointment, please visit the GLS booking page: https://gls.northeastern.edu/gls-tutoring/

Global Learner Support (GLS) also offers monthly virtual and in-person workshops on topics related to avoiding plagiarism, paraphrasing, APA/MLA guidelines, grammar and punctuation, academic presentations, writing professional emails, etc. Please visit https://gls.northeastern.edu/gls-workshops/ to register for upcoming workshops.

To view additional GLS services, visit our website at https://gls.northeastern.edu/

Title IX

Title IX of the Education Amendments of 1972 protects individuals from sex or gender-based discrimination, including discrimination based on gender-identity, in educational programs and activities that receive federal financial assistance.

Northeastern University's Title IX Policy addresses Prohibited Offenses, which are defined as sexual harassment, sexual assault, relationship or domestic violence, and stalking. The Title IX Policy applies to the entire community, including male, female, transgender students, faculty, and staff.

If you or someone you know has been a survivor of a Prohibited Offense, confidential support and guidance can be found through University Health and Counseling Services staff (http://www.northeastern.edu/uhcs/) and the Center for Spirituality, Dialogue, and Service clergy members (http://www.northeastern.edu/spirituallife/). By law, those employees are not required to report allegations of sex or gender-based discrimination to the University.

Alleged violations can be reported non-confidentially to the Title IX Coordinator within The Office for University Equity and Compliance at: titleix@northeastern.edu and/or through NUPD (Emergency 617.373.3333; Non-Emergency 617.373.2121). Reporting Prohibited Offenses to NUPD does NOT commit the victim/affected party to future legal action.

Faculty members are considered "responsible employees" at Northeastern University, meaning they are required to report all allegations of sex or gender-based discrimination to the Title IX Coordinator.

In case of an emergency, please call campus police.

Please visit http://www.northeastern.edu/titleix for a complete list of reporting options and resources both on- and off-campus.

Recording Policy

This course, or parts of this course, may be recorded for educational purposes. Prior to starting any recordings, the instructor will ask if anyone objects to recording the session. If anyone objects, no recording will be made. These recordings will be made available only to students enrolled in the course, the instructor of record, and any teaching assistants assigned to the course.

Only students who have arranged an accommodation with the Disability Resource Center may use mechanical or electronic transcribing, recording, or communication devices in the classroom. Students with disabilities who believe they may need such an accommodation may contact the Disabilities Resource Center.