Extended Syllabus
(2024 Spring Semester)

Course Title	Introduction to Computer Systems	Course Number	CSE3030-01
Credit	3.0	Enrollment Eligibility	Sophomore (2nd-year) and Junior (3rd-year)
Meeting Times	Tuesday and Thursday (09:00 ~ 10:15)	Classroom	TBD

Instructor: Prof. Euhyun Moon	Homepage: https://gordonmoon.github.io
Email: ehmoon@sogang.ac.kr	Telephone: 02-705-8487
Office: AS 813 Office Hours: Tu Th 11:00 ~ 13:00 or by appointment TA: Eunji Lee TA Office Hours: TBD	

I. Course Overview

1. Description

This course provides basic concepts of how computer systems execute programs, store data, and communicate. The topics covered in this course include bit-level representations, machine-level assembly language programming, memory organization and management, code optimization, and specialized architectures and systems.

2. Prerequisites

This course assumes that students are already familiar with the concepts of C programming language. If you have any questions about the prerequisites for this course, don't hesitate to reach out to the instructor.

3. Course Format (%)

Lecture	Discussion	Experiment/Practicum	Field study	Presentations	Other
100 %	%	%	%	%	%

4. Evaluation (%)

Mid term exam	Final exam	Quizzes	Presentations	Projects	Assignments	Participation	Other
35 %	35 %	%	%	%	30 %	%	%

II. Course Objectives

Upon course completion, students can be expected to:

- Be competent with fundamental concepts of computer systems – understand architectural characteristics of computers which directly affect performance of program
- Be able to find and eliminate bugs in the program efficiently
- Be able to improve the quality and performance of program
- Be prepared for other systems courses, such as Compilers, Operating Systems, Networks, Computer Architecture, Parallel and Distributed Computing, and Embedded Systems

III. Course Format

In each week, we will hold two classes: Tuesday and Thursday from 9:00am to 10:15am

IV. Course Requirements and Grading Criteria

The final course grade will be based on a composite score computed according to the following breakdown:

Midterm exam	35%
Final exam	40%
Programming assignments	25%

Both midterm and final exams are in-class exam covering all course content to date.
The programming assignments are graded according to the following scale and expectations, with scores then converted to a percentage equivalent on a scale specific to and provided with each assignment.

Project grade	Requirements
10	professional quality, correct and documented code, brief well-substantiated conclusions, thoughtful and neatly completed, correctly submitted
8	essentially correct in all aspects, lacking in quality of arguments or submission
6	contains one or two minor errors or omissions of key concepts
4	contains one significant or multiple minor errors or omissions
2	lacking multiple significant components
0	work not submitted

V. Course Policies

General policies:

- Following instructions are graded part of all assignments
- All exams are closed book, closed notes

Lectures and Attendance:

- Attendance is required for all students.
- Please get to know your classmates. Should you miss a lecture, it is your responsibility to review that material with someone who is willing to share their notes with you and then bring follow-up questions to office hours. It is your responsibility to ensure you have all course materials.

Exams:

- The midterm will be held during a regularly scheduled course lecture meeting time, thus there should be no scheduling conflicts. The midterm may not be taken early and may not be made up.
- The final exam will be held during the time slot scheduled by the university, thus there should be no scheduling conflicts. The final exam may not be taken early and may not be made up.
- Exams are due at the end of the examination period as announced by the instructor. Continuing to work on your exam after the examination period has ended may result in your work being considered late and a reduction in your score, up to and including receiving a score of 0 for the exam.
- In the event of an unavoidable unanticipated absence from an exam, the student should notify the instructor as soon as possible.

Electronic Media:

- Students are responsible for being aware of any announcements made via Cyber Campus or email.

- When communicating with your instructor via email, please be sure to include your name, class and section.

Programming Assignments:

- Programming assignments will be assigned via posting to the Cyber Campus (https://cyber.sogang.ac.kr) in the "Assignments" section.
- All programming assignments must be submitted/uploaded to the Cyber Campus in the "Assignments" section.
- All programming assignments are individual exercises. The work submitted by a student is expected to be that student own original work.
- Copying the source code of another student is not allowed. Such a violation may result in academic penalties. For the first occurrence, you will receive a zero and reduction in one letter grade (e.g., A→B, B→C). For the second occurrence, you will receive an "F" in this course.
- It is expected that students are either proficient in C or have sufficient programming background and experience to become proficient through self-study.
- Office hours are not for general program development. Office hours should be used for specific design questions or debugging specific issues.
- Programming assignments will be accepted past the due date and time according to the following cumulative penalties:

24 hours late	-20%

- No late homework will be accepted after 24 hours from the due date, please plan accordingly.

Academic Integrity:

- It is expected in this course that students are familiar with the academic integrity guidelines of Sogang University as defined by the Office of Academic Administration.
- It is expected that students will only turn in work that is their own, or the work of team to which they have been assigned for a given specific assignment.
- It is expected that students will neither seek nor receive any form of aid, other than from the instructor or proctor, during any exam or quiz.
- In the event that there occurs reasonable doubt about the integrity of any student's work, then that student and said work will be referred to the Committee on Academic Misconduct for adjudication.

VI. Materials and References

Lecture slides will be posted to Cyber Campus (https://cyber.sogang.ac.kr), but are not intended to replace the textbook nor the lectures themselves. Please use the slides as a medium to organize your course notes and as study aids.

Textbook:

Randal E. Bryant and David R. O'Hallaron, Computer Systems: A Programmer's Perspective, 3rd Edition, Pearson, 2016.

VII. Course Schedule

Week 1	**Learning Objectives**	Introduction to Computer Systems
	Topics	Overview of Computer Systems
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 2	**Learning Objectives**	Bits, Bytes, and Integers I
	Topics	Representing information as bits, Bit-level manipulation
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 3	**Learning Objectives**	Bits, Bytes, and Integers II
	Topics	Integers, Byte order in memory, pointers, strings
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 4	**Learning Objectives**	Machine-Level Programming I: Basics
	Topics	Intel processors and architectures, Assembly basics, Arithmetic and logical operations, C, assembly and machine code
	Class Work (Methods)	Lecture

	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 5	**Learning Objectives**	Machine-Level Programming II: Control
	Topics	Control flow, Condition, Conditional operations, Loops
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 6	**Learning Objectives**	Machine-Level Programming III: Procedures
	Topics	Stack structure, Calling conventions
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 7	**Learning Objectives**	Machine-Level Programming III: Procedures
	Topics	Register Saving Conventions, Recursion
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 8	**Learning Objectives**	Midterm Exam
	Topics	
	Class Work (Methods)	

	Materials (Required Readings)	
	Assignments	
Week 9	**Learning Objectives**	Machine-Level Programming IV: Data
	Topics	One-dimensional arrays, Multi-dimensional arrays, Structs
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 10	**Learning Objectives**	Machine-Level Programming V: Buffer Overflows
	Topics	Memory layouts, Buffer overflow
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 11	**Learning Objectives**	Memory & Caches
	Topics	Memory abstraction, RAM, Locality of reference, Memory hierarchy, Storage technologies
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 12	**Learning Objectives**	Caches I
	Topics	Cache memory organization and operation
	Class Work (Methods)	Lecture

	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 13	**Learning Objectives**	Caches II
	Topics	Performance impact of caches
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides
	Assignments	
Week 14	**Learning Objectives**	Code Optimization
	Topics	Compiler optimization, Machine-dependent optimization
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 15	**Learning Objectives**	Review and makeup class
	Topics	
	Class Work (Methods)	Lecture
	Materials (Required Readings)	Lecture slides and textbook
	Assignments	
Week 16	**Learning Objectives**	Final Exam
	Topics	
	Class Work (Methods)	

	Materials (Required Readings)	
	Assignments	

VIII. Special Accommodations

IX. Aid for the Challenged Students

If you have a disability that may affect your success in this course and wish to discuss academic accommodations, please arrange a meeting with the instructor on the first day of class.