CIS341
Computer Organization & Programming Systems
Syracuse University, Fall 2024

1 Basic Information

About the course

Core course for computer science; 3 credits

Prerequisite: CIS351

Course description: Digital logic, data type and representation, instruction set architecture, assembly language, program construction, processors, memory hierarchy, traps and interrupts, privilege and security, I/O

Time & location

Lecture (M001): TuTh 3:30pm – 4:50pm @ CST 4-201

Lab (M002): Fr 10:35am – 11:30am @ CST 3-116

Lab (M003): Fr 11:40am – 12:35pm @ CST 3-116

Instructor: Prof. Bryan S. Kim (bkim01@syr.edu)

Office hours TBD

Teaching assistants: Ziyang Jiao

Office hours TBD

Peer-learning assistants: TBD

Office hours TBD

Textbook

Title: *Computer Systems: A Programmer's Perspective*, Third Edition, Pearson, 2015. (ISBN: 9780134092669)

Authors: Randal E. Bryant and David R. O'Hallaron

Also available through Orange Inclusive Access (OIA)

2 Course Description

2.1 Course objectives

This course is designed to help you understand *how computers are built* and *how programs run on a system*. Although it may seem like computers are magical mystery machines to *users*, they are, in fact, very far from it. As computer scientists and engineers, we must understand what really happens when a program runs on a machine and think about more efficient ways to do so.

The learning objectives of this course include the following.

1. Explain common bit-level representations of numeric values and the consequent mathematical properties of arithmetic and bit-level operations on them.
2. Translate a C function into an assembly code including the implementation of expressions, control, and procedures by recalling the corresponding instruction set architecture.
3. Explain the organization of the classical von Neumann machine and its major functional units, and estimate the performance improvements of common performance optimization techniques in modern processors.
4. Modify a C function to maximize performance while retaining its functional correctness by assessing the effect of each expression on the processor and its memory subsystem.
5. Estimate the performance of cache memory, and explain the workings of a system with virtual memory management.
6. Examine the sources of conflict that can arise when multiple threads of execution share resources, and demonstrate the ability to use synchronization constructs to mediate those conflicts.
7. Explain the programmer's interaction with the underlying system through the different APIs and abstractions, including system support for process and thread control, virtual memory, and system I/O.

2.2 Schedule

The following is a tentative schedule and is subject to change.

Week	Tuesday lecture	Thursday lecture	Events
1	Overview	C basics	
2	C pointers & arrays	C memory management	`hw1` due
3	Integers	Floats	`proj1-C` due
4	Machine organization	Instruction control flow 1	`hw2` due
5	Instruction control flow 2	Procedures	`proj2-asm` due
6	Array & structs	Stack buffer overflow	`hw3` due
7	CPU potpourri	Midterm review	`proj3-stack` due
8	Fall break (no class)	Midterm	Midterm (Oct. 17, 3:30pm–4:50pm)
9	Cache memory 1	Cache memory 2	
10	Cache-friendliness	Virtual memory 1	`proj4-cache` due
11	Virtual memory 2	Memory potpourri	`hw4` due
12	Hard disk drives	Solid-state drive	`proj5-vm` due
13	Flash translation layer 1	Flash translation layer 2	`hw5` due
		Thanksgiving break	
14	Reliability	System-level I/O	`proj6-ssd` due
15	Final exam review		Final exam (Dec. 13, 5:15pm–7:15pm)

3 Course Components, Assessment, & Grading

This class will use a point-based system (maximum 500 points).

- Exams: Two 100-point exams
- Projects: Six 40-point projects
- Homework: Two 10-point & three 20-point homework assignments
- Lecture worksheets: Some number of 2-point worksheets
- Lab worksheets: Some number of 4-point worksheets

The course materials are divided into two large *modules*, with a maximum 250 points for each module. The two modules are as follows.

- **Data representation and instruction processing** (maximum 250 points)
  - Exams: Midterm exam (100 points)
  - Projects: `proj1-C`, `proj2-asm`, and `proj3-stack` (120 points)
  - Homework: `hw1`, `hw2`, and `hw3` (40 points)
  - Worksheets: At least six lecture worksheets and three lab worksheets (24+ points)
- **Memory and Storage** (maximum 250 points)
  - Exams: Final exam (100 points)
  - Projects: `proj4-cache`, `proj5-vm`, and `proj6-ssd` (120 points)

- Homework: `hw4` and `hw5` (40 points)
- Worksheets: At least six lecture worksheets and three lab worksheets (24+ points)

Unfortunately, Blackboard does not support this type of capped point tracking system, so the points shown on Blackboard may be greater than the actual points you have. A separate Excel spreadsheet will be used by the instructor to track the points.

The summation of the points from the two modules (with point capping) will be compared against the numerical thresholds below to assign the final letter grade.

Points	**Letter grade**
$450 \leq X$	A
$425 \leq X < 450$	A-
$400 \leq X < 425$	B+
$375 \leq X < 400$	B
$350 \leq X < 375$	B-
$325 \leq X < 350$	C+
$300 \leq X < 325$	C
$275 \leq X < 300$	C-
$X < 275$	F

No **incomplete** grade will be provided without a valid reason. Violations of academic integrity override the table above and could result in an F grade. Finally, the instructor reserves the right to change this grading scale.

3.1 Exams

There will be two exams: one midterm in class on Oct. 17 and a final exam during finals week on Dec. 13. These exam dates are fixed and will not change. Exams are closed books and closed notes.

For the exams, there are no extensions or make-ups, except for university-accepted reasons. If a student misses exams without a university-accepted reason, a zero will be assigned. Students are responsible for contacting the instructor as soon as possible if they are unable to take any exams due to university-accepted reasons.

3.2 Projects & Homework

Both the projects and homework share a number of class policies with exceptions on how they are turned in. Projects are designed to provide hands-on experience in understanding the learning objectives of the course, and the homework problems are designed to reinforce the concepts learned during lectures and help students prepare for the exams.

3.2.1 Common policies for projects and homework

Teamwork and size. For projects and homework, students may form a team of up to 3 members. You may work alone or in a pair, but having fewer members will not grant extensions, bonuses, or

advantages.

Contribution for teamwork. All team members are expected to contribute, and their contribution factor, from a scale of 1-5, must be stated at the beginning of the written report. The contributions factors must be agreed upon by all the members, and not stating them in the report will result in a significant penalty.

Availability and due dates. In general, all projects and homework will become available by Fridays 11:59pm and will be due on Wednesdays at 11:59pm. Students are given at least 12 calendar days for the projects, and at least 5 calendary days for the homework.

Written report. A written report must be turned in via Blackboard for both the project and homework. Only one team member needs to upload the report to Blackboard. We will not accept work turned in via e-mail. A guideline to what needs to be included will be stated in the project handout or the homework assignment.

Late turn-in. Aside from the due date, there is a 24-hour *grace period* for all assignments, both projects and homework. Beyond this grace period, each late day will reduce the maximum attainable points for the assignment by 20% of the assignment. Even one minute after the deadline will count as a full day late once the grace hours are exhausted.

Extensions. For projects and homework, extensions may be granted only if all team members have a well-reason cause. Please e-mail the instructor as to why your team would need an extension. Keep in mind that fairness is a prime consideration for granting extensions. Furthermore, no extensions will be granted after the due date for that assignment (no last-minute extensions).

Grading. Taking the contribution and late turn-in into consideration, the points (P) you receive for an assignment are as follows.

- x is the raw graded points based on the grading guidelines for the assignment.
- t is the maximum number of points for the assignment: 20 for homework, 40 for projects.
- l is the number of days late beyond the grace period, value between 1 and 5.
- c is your contribution for the assignment, value between 1 and 5.

$P(x, t, l, c) = min(x, t \times (1 - l \times 0.2)) \times (0.2 \times c)$

3.2.2 Policies specific for projects

The project will involve some hands-on experience in design, implementation, and evalution. The `C` and `C++` programming language plays a critical role in the projects, but the level of proficiency required is minimal.

All projects will be made available on our class Linux server (`lcs-vc-cis341.syr.edu`). To log in to the server, connect via the `ssh` (secure shell) protocol from a terminal, and use your NetID and its password.

The team will be asked to turn in any software artifacts they worked on for the project, in addition to the written report. The written report must specify the location of the software artifact so that the staff can inspect the code.

We will not accept work turned in via e-mail. The submissions will be cross-checked through a code-similarity analysis tool. Please do NOT cheat.

3.3 Lecture & Lab Worksheets

Group-based worksheets will be done during some of the lectures and labs. Each group consists of up to 4 members. This group may be different from the team formed for the projects or homework. The group worksheets are based on that day, week, or unit's lecture or lab contents, and the group's written response will be graded.

4 Course Policies

4.1 Piazza

For any class-related questions, please post them on Piazza (rather than e-mailing the teaching staff). If you have any problems or feedback for the website, e-mail team@piazza.com. Our class page is `https://piazza.com/syr/fall2024/cis341`.

4.2 Academic Integrity

For general expectations and policies of the University, please refer to `http://class.syr.edu/academic-integrity/policy/`.

Specifically for the projects in this course, we will use moss (`https://theory.stanford.edu/~aiken/moss/`), a system for detecting software similarity, extensively to detect plagiarism among this semester's submissions as well as previous years' submissions.

4.3 Data Collection

As part of the regular ABET (Accreditation Board for Engineering and Technology) process for the undergraduate programs in our department, we will be collecting samples of students' work in this class. As a result, some of your work (programming assignments, weekly assignments, exams) may be photocopied, scanned, and saved. Your registration and continued enrollment constitute your permission.

4.4 Attendance and Participation

Attendance is expected in all courses at Syracuse University, and we will have in-class problem solving activities to further encourage attendance. If you miss a lecture, it is your own responsibility to obtain all course content and announcements presented in that lecture. It is also your responsibility to go through the covered material (including, recitations, lectures, homework) on your own.

5 University Policies

Please refer to the following web page for our university's policies.

`https://academicaffairs.syracuse.edu/important-syllabus-reminders/`