

## CSI 402 – Systems Programming – Fall 2017

**Instructor:** Charalampos Chelmis

**Office Hours:** W 3:00 - 5:00 pm or by appointment

**Office Location:** UAB 424B

**Email:** cchelmis@albany.edu

**Course Description/Overview:** This course introduces students to the programming aspects of operating systems. Topics covered include implementation of storage management, resource allocation, multi-processing, scheduling, synchronization, inter-process communication, and terminal I/O. Programming assignments are designed to enhance subject understanding, problem solving, and programming skills through hands-on experience on real-life-like problems.

**Learning Objectives/Outcomes:** Throughout the course of this class, students will (i) work in an active learning environment that will enable them to be actively engaged with knowledge, and (ii) be provided with opportunities for critical inquiry into the assumptions, goals, and methods of designing, developing, and implementing system software. At the completion of this course, students will:

- be able to understand and articulate what system software is and does,
- gain a general understanding of fundamental software concepts, and the relationship between machine architecture and system software,
- understand the fundamental machine-independent aspects of system software design,
- be able to develop machine-dependent and machine-independent extensions to system software with desired features, and
- gain a general but not superficial understanding of implementation details, such as algorithms and data structures.

**Prerequisite:** ICSI/ICEN 333 or equivalent with a grade of *at least C*.

### Required Textbooks:

1. L. L. Beck, “System Software: An Introduction to Systems Programming”, Addison-Wesley, 3rd edition, 1997. (ISBN: 0-201-42300-6).
2. K. Haviland, D. Gray and B. Salama, “Unix System Programming”, Addison-Wesley, 1999. (ISBN: 0-201-87758-9).

### Recommended Textbooks:

1. H. M. Deitel and P. J. Deitel, “C How to Program”, Seventh Edition, Prentice-Hall, 2014.
2. S. Harbison and G. Steele, “C: A Reference Manual”, Fifth Edition, Prentice Hall, Upper Saddle River, NJ, 2002 (ISBN: 0-13-089592X.)

**Course Website and Blackboard:** Blackboard will be used to provide essential course materials, the most current syllabus, and assignment documents.

**Special Note:** At the first meeting of the class, a special “*entrance quiz*” will be administered. The quiz will not count towards the final grade. Please do come to the first meeting if you intend to take the course, whether you have already registered or are still interested in acquiring a permission number. The quiz will be completely closed-book. Consulting Internet or other electronic or printed resources is not permitted. The quiz will be graded and assessed by the Instructor. Students who do not meet the passing threshold will **not be permitted to take the course and will be automatically de-registered from the class**. Similarly, students who demonstrate significant deficiencies, but do not fail the quiz, will be advised to select, with the assistance of their academic advisors, other courses in which they are most prepared and will be more likely to succeed. Missing the “entrance quiz” will result in an automatic deregistration from the class.

**Assessment:** The accomplishment of the course objectives will be assessed by applying the concepts and tools for system software design in a combination of programming assignments, and exams as described below.

**Exams:** Two exams will be given, a mid-term and a final exam. Each of the two exams will cover material from the required textbooks, and may contain questions pertaining to the programming assignments. Exams will be individual effort, closed-book and closed-notes. One 8.5”x11” *handwritten* note sheet (front & back) will be allowed. No access to any other materials printed, online, or in an electronic device will be permitted. The final exam will be comprehensive covering materials from the entire course.

*Missing any of the exams will result in an automatic E grade for the course.* Makeup exams will be given only for valid and verifiable *extenuating* circumstances (e.g., a major medical situation). It is the student’s responsibility to contact the Instructor *ahead of time* and arrange to take a makeup exam at an alternate date/time. Makeup exams are not guaranteed and will be generally harder than the regular exams.

**Programming Assignments:** There will be five (5) programming assignments. These assignments must be done using the machines provided by the Information Technology Services (ITS) unit of the University. You can log on to these machines remotely. The programming assignments must be done using the programming language C. Additional information about programming assignments is provided at this end of this document.

**Grading:** A final grade will be determined as a weighted average of the programming assignments and exams scores using the following weights:

Midterm	October 17, 2017 (class meeting time)	In class	35%
Final	December 14, 2017 (10:30am–12:30pm)	In class	50%
Programming Assignments (5)	To be announced	(3% each)	15%

#### Grading Scale:

A: 100 – 95%	A-: 94 – 90%	
B+: 89 – 87%	B: 84 – 86%	B-: 80 – 83%
C+: 79 – 76%	C: 75 – 70%	
D: 69 – 60%	E: 59% or less	

Students must complete all requirements in order to pass the course. A grade of incomplete will be given only when circumstances beyond the student’s control cause a substantial amount of course work to be unfinished by the end of the semester. Whenever possible, the student is expected to make extra efforts to prevent this situation from occurring. The instructor will be the sole judge of whether an incomplete is warranted. Final grades are computed based on the above formulas

and are *NOT* negotiable. Per department policy, “*..students may not submit additional work or be re-examined for the purpose of improving their grades once the course has been completed and final grades assigned.*”

**Attendance/Use of Computers:** Attendance in lectures is *NOT* required, however, you are *strongly* advised to attend the lectures. Note that even though some of the material covered in class may not appear on the lecture slides, students are responsible for all materials covered in lectures. It is your responsibility to find out the material covered in a lecture you missed.

Please *DO NOT* disrupt the class by entering late or leaving early without instructor’s approval. Computers may be used during class for note taking as long as the use is not disruptive or distracting. The use of cellphones or other electronic devices during class is not permitted.

**Responsible Computing:** Students are required to read the University at Albany Policy for the Responsible Use of Information Technology ([http://www.albany.edu/its/policies\\_responsible\\_use\\_of\\_IT.htm](http://www.albany.edu/its/policies_responsible_use_of_IT.htm)). Students are expected to apply the policies discussed in this document to all electronic communications in the course.

**Email Correspondence:** Make sure to use the Discussion Board on Blackboard for questions regarding the programming assignments. Your question may have been already answered. Send the Instructor an email only as a last resort. In that case, you must use proper business etiquette. Do not expect an instant response. Also, do not expect a response to emails that do not have the name of the person and class info from the person who sent it. Additionally, the Instructor may choose not to respond to emails that are not written properly. Re-read what you write before you hit send. The subject line of the email should contain the class number (i.e., 402).

**Students With Disabilities:** Reasonable accommodations will be provided for students with documented physical, sensory, systemic, cognitive, learning and psychiatric disabilities. If you believe you have a disability requiring accommodation in this class, please notify the Director of the Disability Resource Center (Campus Center 137, 442-5490). That office will provide the course instructor with verification of your disability, and will recommend appropriate accommodations. For further information refer to the University’s Disclosure Statement regarding Reasonable Accommodation found at the bottom of the document at the following website: <http://www.albany.edu/disability/docs/RAP.doc>. This website can be reached by following the link under Reasonable Accommodation Policy at the following webpage: <http://www.albany.edu/disability/faculty-staff.shtml>.

**Academic Honesty and Policy on Cheating:** Students have the responsibility to familiarize themselves with the Standards of Academic Integrity and Policies in the Undergraduate Bulletin ([http://www.albany.edu/undergraduate\\_bulletin/regulations.html](http://www.albany.edu/undergraduate_bulletin/regulations.html)) and behave accordingly. Students’ claims of ignorance, unintentional error, or personal or academic pressures cannot be excuses for violation of academic integrity. Faculty are responsible for teaching, modeling and upholding them. Anything less undermines the worth and value of our intellectual work, and the reputation and credibility of the University at Albany degree.

Plagiarism and other acts of academic dishonesty will be punished. Programming assignments in this course are meant to be the result of teamwork. While you may discuss a problem with other students (e.g., on the Discussion Board in Blackboard), the work you submit must be of your own team. Additionally, there will be **zero tolerance** for cheating. Cheating in any one examination, quiz, or programming assignment will result in an E grade for the course. Further, students involved in cheating will be referred to the Dean’s office for disciplinary action.

## Additional Information Regarding Programming Assignments

**Documentation:** The documentation for each program must include the following:

- (a) Header information (consisting of all team members' names, email addresses, and roles) at the beginning of the program. Each team member should assume the role of a leader, recorder, and monitor in a round robin fashion.
- (b) An overall description of what the program does along with descriptions of inputs to the program and the outputs produced by the program. Any assumptions that you make with respect to the inputs must be clearly stated.
- (c) The purpose of every constant, data type and variable declared in your program must be stated at the point of declaration.
- (d) For each function you must provide the following:
  - (i) A description of what the function does.
  - (ii) Descriptions of all the parameters.
- (e) In addition to the above, you must also include in-line comments (i.e., comments interspersed with code) to convey the logic behind your code.

In order for the Instructor to provide any help with your program, you must follow the aforementioned documentation guidelines.

**Submission information:** For each program, you must electronically submit the required file(s) by the deadline. You should **not** e-mail your source files to the Instructor.)

At the beginning of each of your source files, you must include (a) your name and (b) your email address in the form of comments. Failure to do so will result in a 5% **penalty**.

**Late Submission Policy:** The due date and the weightage for each of the five programming assignments will be indicated on the assignment sheet. There is no late submission period. The due time of 11:59 PM is EST time and is based on our clock. Even if you submit a few minutes after the deadline, your submission will be considered late. It is your job to be on time and not cut it too close. Remember Murphy's Law and leave time for things to "go wrong."

**Correctness of Programming Assignments and Grading:** For each programming assignment, ~ 85% of the grade will be for correctness and the remaining 15% will be for structure and documentation. *If your program has compilation or linking errors, the grade assigned will be zero.* To grade your program for correctness, a script will automatically generate an executable version of your program and run the executable on new input data sets prepared by the Instructor and/or the grader(s). Even when your program works correctly on all the sample data sets which were made available to you, it may fail on the new data sets. *It is your responsibility to test your program thoroughly (e.g., by preparing additional input data sets) before the program is submitted. It is not possible for the Instructor to identify or fix the bugs in the submitted version of your program.*