

CSI 402 – Lecture 0
(Additional Course Information)

Course Information

- **Prerequisite:** CSI 333 (Programming at the Hardware-Software Interface) or equivalent with a grade of at least C.
- **Homepage:** www.albany.edu/~csi402
- Programming assignments must be done on the Unix machines supported by ITS.
- All programming assignments are **individual exercises**; **there are no team projects**.

Caution

The course is **programming-intensive**; it is **not** suitable for students who don't like to write large programs.

Some Good and Bad Reasons for taking CSI 402

Good Reasons:

- You are pursuing the B.S. degree in Computer Science and you have met the CSI 333 prerequisite.
- You are a graduate student in Computer Science completing the undergraduate deficiency in CSI 402 and you have met the CSI 333 prerequisite.
- You enjoy writing/debugging large C programs under Unix and you have met the CSI 333 prerequisite.

Bad Reasons:

- You don't have the CSI 333 prerequisite; however, the course "fits nicely into your schedule".
- You don't have the CSI 333 prerequisite; however, you have heard rumors that "the course is very easy".

More on Programming Assignments

- **Programs will have strict deadlines.** (Make it a habit to start working on each programming assignment soon after it is assigned.)
- **No credit will be given for programs which have compilation or linking errors.**
- All C programs must be split into multiple files. A `makefile` must be provided.
- Programs must have an adequate level of documentation. (See the handout on “Course Policies” for documentation guidelines.)
- **Program grading:** Approximately 85% for correctness and 15% for structure/documentation.

Guarding Against Accidental Deletion of Files

- Make backup copies frequently.
- Use a suitable alias so that the `rm` Unix command doesn't delete files silently.

Example: If you are using `bash`, add the following line to the file `.bash_profile` (which in your home directory):

```
alias  rm='rm  -i'
```

Background Assumed

- A thorough knowledge of programming and basic data structures.
- A thorough knowledge of C (including pointers, standard libraries, files, strings, macros, bitwise operators, command line arguments, etc.).
- Complete familiarity with Unix and associated software tools such as gcc.
- A thorough knowledge of various number systems, binary arithmetic and 2's complement representation.
- A thorough knowledge of assembly language programming.

Topics to be Covered (tentative)

- Developing C programs using multiple source files and `make`.
- Additional material on files.
- Assemblers.
- Linkers and Loaders.
- Files and directories under Unix.
- Process creation under Unix.
- Inter-process communication under Unix.
- Signals and signal handling under Unix.

Expected Coverage:

- Chapters 1 through 4 and Appendix A of the text by Beck.
- Chapters 1 through 7, 9 and 10 of the text by Haviland et al.

Suggestions:

- Skim the lecture slides and the handouts before each lecture.
- Bring a copy of lecture slides and handouts to class.
- After each lecture:
 - Carefully study the lecture slides, the relevant material from the texts and the handouts.
 - Do the exercises suggested in the lecture slides.