Fall 9-1-2000

CS 131.00: Fundamentals of Computer Science I

Michael O'Conner
University of Montana - Missoula, michael.oconner@umontana.edu

Fundamentals of Computer Science I (CS 131) and II (CS 132)
Syllabus - Autumn 2000

Instructor: Mike O'Conner
Office: Social Science 413
Office Hours: TBA
Phone: 243-2217
e-mail: o_conner@selway.umt.edu

Teaching Assistant: Garrett Mosey
Office: TBA
Office Hours: TBA

Scheduled Labs: to be determined

Recommended Computer Labs on Campus:
Fine Arts 210
Liberal Arts 242
University Center 225
Liberal Arts 206
(see the CIS lab handout for schedules and policies)

Texts:

Java Software Solutions (Addison Wesley)
Second Edition (orange "color theme" on the cover)
- Lewis, Loftus

E.B.N.F.
- Fac Pac for CS 131 only

Fundamentals of Computer Science II
- Fac Pac for CS 132 only

Course Synopsis:

Fundamentals of Computer Science is intended to be an introduction to a diversity of topics in the field of Computer Science: a "breadth-first" overview.

Much of the "exploration" will be accomplished by learning elements of the Java programming language.

The Java language was chosen because:
- it can be used to illustrate most of the elements of computer science we wish to study
- it insulates the learner from many of the troublesome aspects found in other languages
- it incorporates the syntax used in C and C++ (two widely-used programming languages)
- it is freely downloadable
- applets - java programs that can be downloaded from the web and executed by a web browser - are popular

Our breadth-first exploration of computer science involves 4 views:

(1) One focus of computer science is software engineering: how humans control machines by means of commands expressed in some language. In learning such a language one also becomes familiar with paradigms, concepts, principles, models, patterns, conventions, tools, and skills employed in software engineering.

(2) Another focus of computer science is the machine itself:
- electronics and the "digital" concept(transistor, switch, gate, latch, flip-flop)
- components (memory, ALU, ...)
- architecture (the integration of components to form a machine)
- networking (interconnecting machines to enhance functionality, economy, and efficiency)

(3) A third focus of computer science is the application of the computer

programming, simulation,
Data Base Management Systems, and various artificial intelligence topics (robotics, machine learning, genetic algorithms and other problem-solving methodologies, speech recognition, and others). We will write programs which illustrate the principles we wish to study in those topics.

(4) Because of the potential for the abuse of the knowledge, skills, and privileges gained through the study of computer science, we will also focus on computer ethics. By examining ethical issues, we hope to raise each student's awareness of the issues, as well as to encourage ethically sound decisions throughout that person's computer science career.

Course Objectives:

Ten major objectives of the Fundamentals of Computer Science course are listed below. They are abilities the successful student will acquire.

1) become proficient in the creation of software, based on:
    problem analysis
    solution design
    program implementation in various paradigms
    program verification and modification
2) become adept in the use of UNIX's basic commands and file system
3) create interactive programs in both command line and graphical interfaces
4) understand the specification of a programming language, and the steps
    involved in translating a program into machine-executable form
5) manage data via static and dynamic structures
6) understand the use of recursion, dynamic programming, and simulation as
    problem-solving tools
7) implement the basic operations of a relational database
8) understand the basics of combinational and sequential logic circuits
9) employ simple artificial intelligence models, such as
    searching a solution space,
    utilizing machine learning, or
    creating model simulations,
  to complement problem-solving efforts
10) develop an appreciation of the importance of professional ethics in
    the field of Computer Science

Proposed Course Agenda 131:

Software engineering is the art/science of controlling a computer (a symbol-manipulating machine) by means of commands expressed in a language devised for that purpose. In Fundamentals of Computer Science I, we consider two programming paradigms (which we can demonstrate using java), software engineering, a graphical user interface, and ethics.

Introduction, Syllabus, New Accounts
Ch 1 - Computer Systems: Hardware, Network, Software and Programming
Ch 2 - Objects and Primitive Data: Type, Assignment, Input, Output
Ch 3 - Program Statements: Sequence, Selection, Repetition
Ch 4 - Writing Classes: Methods (p186 to p194)
EBNF Specification of a Language
Ethics #1
Ch 4 - Writing Classes: Encapsulation, Overloading
Ch 5 - Enhancing Classes: References, Interfaces, Polymorphism, Events
Ch 6 - Arrays and Vectors: Sorting, Multi-dimensional Arrays
Ch 7 - Inheritance: Overriding, Polymorphism
Ch 8 - Exceptions and I/O Streams: Keyboard and File Input/Output
Ethics #2
Ch 9 - Graphical User Interfaces: user-application Interaction
    (via events and event handling)

To successfully complete CS 131, the student will:
- demonstrate a basic understanding of the "von Neumann" architecture, including the components and their roles in the fetch-decode-execute cycle
- acquire familiarity with the programming environment:
  basic operating system commands related to programming, the file

system, text editors, compiler/interpreter, remote login, ftp
- develop facility in software engineering skills (application of the waterfall and spiral models of program lifecycle), including: problem analysis, program design, prototyping, incremental implementation, debugging and testing strategies
- demonstrate programming ability in a variety of forms: batch and interactive, menu-driven and event-driven, application and applet, command line interface and graphical user interface, (pseudo-)imperative paradigm and object-oriented paradigm
- begin formation of personal ethics through the resolution of ethical questions/dilemmas by applying "analogy" or an ethical norm

Proposed Course Agenda 132:

In Fundamentals of Computer Science II, we consider ways to make our commands more efficient, more powerful, and more elegant.
Topics include:
- understanding the compiler (SLR Parser)
- tools/techniques for managing data
  (data structures, searching and sorting, databases)
- problem-solving techniques
  (Recursion, Dynamic Programming, Artificial Intelligence topics)
- Logic Design and "Programming in Hardware"

```
Greetings and Syllabus, new CIS Java Environment
Ch 10 - Software Engineering: Review Programming Models and Paradigms
Ch 11 - Recursion: Concept, Activation Record Stack, Call-Tree,
                   some useful "patterns"
Ch 12 - Data Structures: Abstract Data Types, Dynamic Data Structures
                         (List, Stack, Queue, Doubly-Linked List)
SLR Parser: Compiler, Finite Automata and Finite State Machine
Searching and Sorting: Lists, Trees, Big "O"
Dynamic Programming: CYK Parsing Algorithm
Relational Databases: 5 Orthogonal Operators,
                      C and C++ pointers vs Java references
Logic Design and Circuits
Artificial Intelligence Topics
```

To successfully complete CS 132, the student will:
- implement the parsing phase of a compiler, based on both the SLR algorithm (which is used to introduce finite automata and finite state machines) and the CYK algorithm (which is used to illustrate dynamic programming)
- obtain rudimentary skills in using recursion as a programming tool
- acquire facility programming with static and dynamic data structures (array, table, list, stack, queue, deque, tree), including the design and "Big O" analysis of several searching and sorting algorithms
- implement the 5 fundamental operations of Database Management Systems (Union, Set Difference, Cartesian Product, Projection, and Selection) and understand how they interact to provide more complex operations (Join, Intersection, Theta-join, and Quotient)
- design optimized special-purpose combinational logic circuits using the Karnaugh Map technique, and examine sequential logic circuits based on latches and flip-flops
- demonstrate the ability to relate the ACM or SWE Code of Ethics to issues arising in the workplace
- modify/complete code written by a third party
- work on a software project as part of a team

Student Evaluation

:S131 and CS132 will each have:
2 Ethics Essays
5+ Programming Assignments
Possibly 1 or 2 Non-Programming Assignments
midterm 1: at the end of week 5 (Fri 6 Oct)
midterm 2: at the end of week 10 (Fri 10 Nov)

CS 131_01   8:00-10:00 AM -- Fri 22 Dec
CS 131_02   1:10- 3:10 PM -- Mon 18 Dec

final exam (CS132):
CS 132_01   1:10- 3:10 PM -- Thu 21 Dec

No "late" work will be accepted. No exceptions. One homework grade will be dropped if we have 6 homework assignments (not including Ethics essays), and an additional homework grade will be dropped if we have 8 or more homework assignments (not including Ethics essays).

A student's grade for the course will be based on total points accumulated. The grade-assignment scale may be curved from the 90-80-70 scheme.

Receiving a grade of Incomplete is subject to the following criteria:
- special circumstances are involved
- coursework was at an acceptable (passing) level before the special circumstances occurred
- the last day to drop (by signatures on a Drop/Add slip) has passed
- a reasonable expectation exists that the student can complete the course work
- the CS Department paperwork (red form) has been completed, signed by student and teacher, and submitted to the CS Department office

"I do not give grades: you earn them."

Grading Criteria

Ethics Essays:

55	content relevant to assignment topic
10	responds to specific questions from the assignment
10	heading in upper left of page: name Ethics #(1, 2, 3, or 4) CS 131/132 - aut 2000 date
10	acceptable spelling and grammar
10	general appearance (margins, indentation, ...)
5	appropriate length

- Each Ethics assignment will be in a sub-directory of /class/CS131 or /class/CS132.
- The directory will contain the assignment description, plus any other files needed (or useful) for the assignment.

Programs:

5	file heading in comment(s) at top left of file text
	// file name
	// author's name
	// date

35	student-written code is present, appropriate for the assignment, and a reasonable attempt toward doing the assignment
10	proper indentation of code
5	heading information printed out at start of program execution
10	output values labeled appropriately
5	prompt is used for input values

70 - maximum grade if program doesn't compile
80 - maximum grade if it compiles but crashes during execution
90 - maximum grade if it runs to normal termination, but gives wrong answers or results
100 - works as described in the assignment

- Each programming assignment will be in a sub-directory of /class/CS131 or /class/CS132.
- The directory will contain the assignment and possibly several exercises. The exercises are not handed in, but they deal with separate aspects of the assignment. The experience gained by doing the exercises will greatly reduce the effort needed to do the assignment.

When possible, test your program execution on a CS Department computer before submitting it for grading. (You won't be able to test applets or GUI programs.)

Other Assignments:
- specified in the assignment

Important Dates:

Thu 7 Sep	late registration fee begins
Mon 25 Sep	last day to add/drop by DIALBEAR or Cyberbear
Mon 16 Oct	last day to drop by signatures on drop/add form
	last day to change sections or grading option
Mon 30 Oct	advising for spring 2001
Mon 6 Nov	spring registration begins (ends Tue 21 Nov)
Tue 7 Nov	Election Day - Holiday
Fri 10 Nov	Veteran's Day - Holiday
Wed 22 Nov	Thanksgiving Vacation (Wed, Thu, Fri)
Fri 8 Dec	last day to withdraw from Autumn 2000 semester
Fri 15 Dec	last day for "drop" petitions for Autumn 2000 semester
Mon 18 Dec	finals week begins
Mon 8 Jan	Intersession 2001 begins (ends Fri 26 Jan)
Mon 29 Jan	instruction begins for Spring 2001

Miscellaneous Notes

Courtesy:
- During lectures, please don't distract other students by carrying on 'side conversations.
- Be polite to CS Department staff in SS 401.
- If you arrive or leave during class, please try to do so in a way that will minimize any disruptive effects on other students.

Rules:
- When entering the CS Department (Social Sciences, floor 4N) after 5:00 PM, you must print your name, CS class, and destination room number in the security log book.
- You need to show your Student ID to office staff in order to take a test at other than the normal time and place. You will be provided with the best accommodations available at that time, but we can't guarantee normal exam conditions: you may experience interruptions and/or noise or distractions. Also, there may be no one present who is able to clarify a question you may have about the exam.

Collaboration:

The CS Department encourages students to share ideas and discuss assignments and projects. But when the time comes for "fingers to hit the keyboard", we expect everybody to do their own work. The current policy is to consider the person who provides the work and the person who accepts it to be equally accountable.

Lorrie DeYott of CIS asked me to convey the following comments to our students using the CIS Labs:

1) Sometimes a bit on a floppy disk is not set correctly for Windows NT, so students saving to disk should open the command prompt window (click the Start button, then select the command prompt menu choice), type "A:" (without the quotation marks) to change to the floppy drive, then type "chkdsk /f" (without the quotation marks). (This especially applies to the "green" disks sold at the UC.) You only need to do this once for each floppy disk.
(My understanding: NT has an image of the disk it expects to be in the floppy drive. If you switch disks in the middle of something, and try to save to a different disk, NT will write the image it has onto the

(incorrect) disk, and make a mess of whatever is there.)

2) Wait for the green light above the disk drive to go off before removing a floppy disk from the drive.

3) It is generally faster (especially if files will be saved to more than one disk) to work on the hard drive, then copy the file(s) to disk(s) once, just before leaving the machine. It is a good idea to then delete those file(s) on the hard drive (less clutter on lab machines).