Spring 2-1-2004

CS 131.01: Fundamentals of Computer Science I

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

Registration Information:

CS 131

    32124   131_01   MWF   SS 356   1:10 – 2:00 PM

Prerequisite: Computer programming experience in a high-level language such as BASIC, Pascal, or C

Co-requisite: Math 100 or consent of instructor

CS 132

    32125   132_01   MWF   JOUR 304   10:10 – 11:00 AM

    32126   132_02   MWF   SS 352   4:10 – 5:00 PM

Prerequisite: "C" or better in CS 131

Co-requisite: Math 121 or consent of instructor

    Any student who takes CS 131 or CS 132 without having the corresponding prerequisite and/or co-requisite does so at his/her own risk. It is not the responsibility of the instructor to help such a student gain the knowledge that he/she should have obtained from prerequisite/co-requisite courses.

Lab Information:

Two computer lab environments are available: the CS Department lab in SS419, and CIS labs in FA210, LA242, UC225, and LA206. See handouts in the CIS labs for schedules and policies regarding lab use.

Also, students with Internet access may configure and use their personal computers.

Instructor/TA Information:

        Instructor: Mike O'Conner

            Office: Social Science 413

   Office Hours: MWF 11:10 – 12:00

                   MWF 2:10 – 4:00

          Phone: 243-2217 (with voice-message capabilities)

         e-mail: o_conner@cs.umt.edu

  Teaching Assistant: (graders only)

        TA Office: TBA

   TA Office Hours: TBA

Course Information:

Required Text:

    Java Software Solutions, 3rd Edition, Lewis and Loftus, Addison Wesley, 2003

Helpful Additional Resources:

- to acquire a good working knowledge of Java in a short time (good for beginners):

    Murach's Beginning Java 2, Andrea Steelman, Murach, 2001   (or a more recent edition)

- a good introduction to using Java (good for near-beginners):

    The Java Tutorial, Mary Campione and Kathy Walrath, SunSoft Press (Prentice Hall)

    or an online version at

    java.sun.com/docs/books/tutorial/

- for someone who has programming experience:

    On To Java, Patrick Henry Winston, Addison Wesley, 1998   (or a more recent edition)

    or an online version at

    www.ai.mit.edu/people/phw/OnToJava

- for a balanced understanding of Java and Object-Oriented Programming – NOT FOR BEGINNERS! –

    Thinking in Java, Bruce Eckel

    or the freely-downloadable version (using sftp) – see Mike O'Conner for details

    Thinking in Java is well worth buying. Bruce Eckel has also written Thinking in C++, and other "Thinking in" books.

Course Synopsis:
Fundamentals of Computer Science is intended to be an introduction to the programming language Java. In the process of learning Java in CS 131, we will touch on a diversity of topics in the field of Computer Science. CS 132 will finish the Java text begun in CS 131, then introduce the topics of Human-Computer Interface and Intelligent Systems. We will also touch on some basic operations in Graphics.

    The Java language was chosen because:
      - it can be used to illustrate most of the elements we wish to study in computer science
      - 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
      - it is closely associated with web-oriented programming (applets, servlets, …)

Our exploration of computer science in CS 131 and CS 132 involves 4 views:
(1) One focus of computer science is software: the way humans control machines by means of commands expressed in a computer programming language such as Java. In learning and using a programming language one also becomes familiar with paradigms, concepts, principles, models, patterns, conventions, tools, and skills employed to create software.

(2) Another focus of computer science is the machine itself, including topics such as electronics and the "digital" concept, hardware components (memory, ALU, ...), architecture (the integration of components to form a machine), and networking (interconnecting machines for better functionality, economy, and efficiency). In CS 131 we will work with applets.

(3) A third focus of computer science is the application of the computer as a tool. Topics discussed under this view include Human-Computer Interfaces and Intelligent Systems topics (machine learning, genetic algorithms, …)

(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 that arise from the use of computers, 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 basic commands and the file system of Unix-like operating systems
    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 and modeling/simulation as problem-solving tools
    7) employ simple artificial intelligence models, such as
        searching a solution space,
        utilizing machine learning, or
        creating model simulations,
      to complement problem-solving efforts
    8) develop an appreciation of the importance of professional ethics in the field of Computer Science

Proposed Course Agenda - CS 131: Examine fundamental computer science concepts by writing programs that use java, a high-level structured programming language.

Software engineering is the art/science of controlling a computer by means of commands expressed in a language devised for that purpose. In Fundamentals of Computer Science I, we focus on the Java programming language, software engineering, simple computer graphics, and ethics.

Lecture Plan:
   Introduction, Syllabus, New User 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
   Ethics #1
   Ch 4 – Writing Classes: Methods, Encapsulation, Overloading
   Ch 5 – Enhancing Classes: References, Wrappers, Interfaces, Events
   Ch 6 – Arrays: Sorting, Multi-dimensional Arrays
   Ch 7 – Inheritance: Overriding, Polymorphism
   Ethics #2

To successfully complete CS 131, the student will:
- demonstrate a basic understanding of the "von Neumann" architecture, including the components that comprise a computer, 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 with ssh and sftp
- develop facility in software engineering skills (application of the evolutionary model of program creation), 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 recognition of ethical questions/dilemmas and their resolutions via "analogy" or the guidance of an ethical norm

Proposed Course Agenda - CS 132: A continuation of java programming, with additional computer science topics including recursion, basic data structures, Intelligent Systems, basic graphics operations, Human-Computer Interfaces, and ethical issues arising in computer science.

In Fundamentals of Computer Science II, we consider ways to make our programming more efficient, more powerful, and more elegant. Topics include:
- simple tools/techniques for managing data (data structures, searching and sorting algorithms)
- problem-solving techniques involving modeling/simulation, recursion, and Intelligent Systems

Lecture Plan
   Greetings and Syllabus, changes in lab environment, review of the final exam from CS 131
   Ch 8 – Exceptions and I/O Streams: Handle exceptional conditions ("exceptions") if they occur, and examine techniques for managing input and output – with a specific discussion of input and output involving files
   Ch 9 – Graphical User Interfaces: User-Application Interaction (via events and event handling) – a more in-depth discussion of graphics and GUI than what was presented in CS 131
   Ch 10 - Software Engineering: A view of programming models and paradigms
   Ch 11 - Recursion: The concept itself and its application (also including the Activation Record Stack concept, call-trees, and some useful "patterns" for solving problems by use of recursion)
   Ch 12 - Data Structures: Abstract Data Types and Dynamic Data Structures, with a discussion of dynamic data structures: List, Stack, Queue, Doubly-Linked List, and Trees
   - Intelligent Systems Topics, such as the Simple Genetic Algorithm or Machine-Learning,
   - Human-Computer Interface topics
   - basic operations in Graphics

To successfully complete CS 132, the student will:
- obtain .and demonstrate a rudimentary skill in using recursion as a programming/problem-solving tool
- acquire facility programming with static and dynamic data structures (array, table, list, stack, queue), including the design and "Big O" analysis of several searching and sorting algorithms
- modify/complete code written by a third party
- work on a software project as part of a team
- demonstrate the ability to relate the ACM or SWE Code of Ethics to issues arising in the workplace

Student Evaluation:

CS131 and CS132 will each have:
    2 Ethics Essays
    Practical exercises on important topics
    5+ Programming Assignments
    Possibly 1 or 2 Non-Programming Assignments
    Midterm Exam: during the week of Mar 15 – 19
    Final Exam (CS131):
        131_01 (1-2 MWF)   3:20-5:20   Mon 10 May

    Final Exam (CS132):
        132_01 (10-11 MWF) 10:10-12:10   Wed 12 May
        132_02 (4-5 MWF)   3:20-5:20   Tue 11 May

Most students submit homework assignments on or before the due date, often making sacrifices to do so. As recognition of their efforts: work not submitted by midnight on the due date will not be accepted. Therefore, begin projects as soon as possible, and try to complete and submit them well before the due date. A computer problem that delayed your work is not generally an acceptable excuse for late work.

Students who enroll in the class after any due dates have passed will be given two weeks to catch up with the rest of the class.

An average of all homework grades (including zeros) will be calculated, and used in computing the course grade. **Be sure to take every exam and do every assignment, because missing work receives a grade of zero!** If you are not finished by the deadline, hand in what you have! (That means you should be creating programs incrementally, so you have something that works – even if it is not completely finished.)

Accommodations are routinely made for disability-related issues, medical reasons, and work-related or family-related conflicts. In some cases, supporting documentation may be requested. Students utilizing the services of DSS should notify the instructor early in the course (preferably the first week), and mention any desired accommodations.

If you know you will miss an exam, notify the instructor beforehand, so you will be allowed to make arrangements for taking the exam at some other time. (Also see the second item in the "Rules" section of the "Miscellaneous Notes" section below.) If you miss an exam for a reason beyond your control, bring supporting documentation when you request an opportunity to take a "make-up" exam.

A student's grade for the course will be based on total percentage points (tpp) calculated, and a "90-80-70-60" grading scheme will be used to obtain a letter grade. Total percentage points are calculated from average points for assignments, for ethics essays, and for exams – each weighted by a contribution factor: 60% for assignments, 10% for ethics essays, and 30% for exams.
    Formula: tpp = (assigs_average * 60%) + (ethics_average * 10%) + (exams_average * 30%)

For the purposes of a "Pass" or "Not Pass" grade, a "C" or higher is considered passing.

Performance Assessment:
The instructor (and TA's, if appropriate) may examine a student's overall performance during the course and decide whether or not the grade – as produced by the grading scheme above – accurately reflects that student's performance. If it is determined that the grade is not a fair assessment, then an increase of up to 2 total percentage points may be made.

One intent of this provision is to be fair to students whose total points are very close to the dividing line between grades. For example, if the 12 grades that fall in the range 80 to 90 lie between 80 and 83 – with the exception of two grades which lie between 88.5 and 89.5 – then the "performance assessment" criteria makes it possible to consider the work of those students more closely, and award a higher grade if that appears to be justified. (For example, submitting every homework assignment would indicate a serious student, and that would be taken into consideration.) An increase in total percentage points can also make the difference between an "A–" and an "A", or a "B" and a "B+", for example.

Incomplete:
Receiving a grade of Incomplete is subject to the following conditions:
(See the University of Montana Catalog for 2003-2004, page 20 – "Policy on Incompletes" – for university requirements.)
- you must find out the financial impact that an incomplete will have on you (the effect on financial aid, probation, ...), and inform the instructor
- special circumstances (beyond the student's control) are involved
- coursework was at an acceptable (passing) level before the special circumstances occurred
- a reasonable expectation exists that the student can complete the course work independently
- the CS Department paperwork (yellow form) has been completed, signed by student and teacher, and submitted to the CS Department office (or to the instructor) before the end of finals week (19 Dec).

Support Services:
Several of the CS Department floor monitors (who make it possible to provide after-hours access to the department) may be upper division students who can answer questions you may have about CS 131 and CS 132. The floor monitor schedule is typically 5-9 PM every night except Saturday.
Your instructor has office hours and expects students to come and ask questions at those times. Students are also welcome to "drop in", with the understanding that the instructor may be too busy at that moment to talk, but is available for help if that is not the case.

Grading Criteria:

Ethics Essays:
- 50%   content is relevant to the assignment topic
- 10%   responds to the specific questions (if any) stated in the assignment
- 10%   layout, including a heading with your name, and general appearance (margins, indentations, …)
- 10%   acceptable spelling and grammar
- 20%   follows the "organization" of presentation, as discussed in class

Programs:
basic information about the program should appear in comments at the top left of the text file:
```
// file name
// author's name
// CS 131 or CS 132
// date
// brief description of program, including any special instructions regarding execution
```
Maximum Deductions:
- 100%   if program doesn't compile (you get no points)
- 20% compiles but crashes during execution
- 10% runs, but gives wrong answers or results
- 10% improper (or non-existent) indentation of code
- 10% Input/Output handled poorly (no prompts for input, no labels on output)
- 10% poor software engineering practices
- 5%   inadequate commenting
Other reasons for deductions may arise in special circumstances (such as for a "group project").

Other Graded Work:
Grading criteria will be specified individually for each assignment.

Assignments will be announced in class.
Also, assignments will generally be posted on a CS Department computer, and in the directory (or a sub-directory of):
  /class/CS131-o_conner   for CS 131   or   /class/CS132-o_conner   for CS 132

A sub-directory may be used to contain the assignment, support material (if needed), and possibly several exercises related to the assignment. Those exercises are not handed in, but they do involve aspects of the assignment. The experience gained by doing them 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 on CS Department computers that you access via the Internet.)

Important Dates:
(see the back of the "schedule of classes" booklet for a more thorough listing)

Wed	28	Jan	late registration fee begins
Fri	13	Feb	last day to add/drop by Cyberbear, change to "audit", pay fees, or receive full refund
Mon	16	Feb	President's Day – Holiday
Tue	9	Mar	last day to drop by signatures on drop/add form, or change sections or grading option
Mon	29	Mar	Spring Break begins: Mon 29th Mar to Fri 2nd Apr
Mon	12	Apr	advising for Autumn 2004 begins
Mon	19	Apr	begin Cyberbear registration for Spring 2004
Fri	30	Apr	last day to withdraw from Spring 2004 semester
Fri	7	May	last (!) day for drop petitions or grade-option-change petitions for Spring 2004
Mon	10	May	final exam week begins (ends 15 May)

Miscellaneous Notes:
Preparation:
You will get more from the lectures if you have looked over the textbook material beforehand.

Attendance:
I assume students will attend all classes. If you miss a lecture, you alone are responsible for the course material you missed – including information not in the text – as well as any changes regarding assignments and exams. The instructor is not responsible for helping a student acquire material missed due to "cutting class".

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 disruptive effects on other students.
-- Profanity, obscenity, racial slurs, and hate-speech have no place in a scholarly community, and will not be tolerated, neither in class nor in assignments.

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 an exam at other than the normal time and place, if you are taking the exam under their supervision. You will be provided with the best accommodations available at that time, but normal exam conditions can not be guaranteed: 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 and Plagiarism/Cheating:
The CS Department encourages students to share ideas and discuss assignments and projects (collaborate with others).
The CS Department also expects the work you submit to reflect your efforts (be responsible for your performance).
So: "learn" by discussion, but "do" by yourself.

The ability to work together is a valuable tool for learning, and is the most likely scenario you will encounter in the workplace. Thus, you are encouraged to discuss assignments with other students and to seek outside help when you don't understand something. However, when it comes to doing an assignment - "when the fingers hit the keyboard" - you are required to do it on your own and in your own words. Further, you are responsible for understanding what you submit. A copy of someone else's assignment is not acceptable. The only exception is when an assignment is designated as a group project.

In this context, collaboration means to work with others for the purpose of sharing ideas. There are two rules regarding collaboration on assignments:
(1) If you collaborate on an assignment, then the instructor reserves the right to question you about your understanding of the assignment. In other words, the evaluation of your performance does not end with the submission of the finished assignment. The instructor reserves the right to lower your grade based on your lack of understanding of what you submitted.
(2) If you discuss an assignment with others, you MUST acknowledge this discussion when you submit your assignment, or it is likely your efforts will appear to be plagiarism or cheating. Further, if you used some outside reference or web page to find the answer to a question, you should mention this reference or web page in your assignment. There is no penalty for getting help in this way, except for the increased likelihood that you will be questioned about your understanding of what you have submitted. If in doubt, acknowledge collaboration or outside help.
Presenting the work of someone else as if it were your own is "plagiarism"(see the U of M Catalog, page 21). I define cheating as giving the impression of learning something you haven't learned (like copying someone's exam answers).

The CS Department will not tolerate cheating or plagiarism. We do not want students in our program who engage in these practices. Therefore, when there is clear evidence of cheating or plagiarism, the department will recommend suspension or expulsion of the student through the procedures specified in the Student Conduct Code.

In this course the current policy regarding plagiarism/cheating is to consider the person who provides the work and the person who accepts it to be equally accountable. The penalties are severe: don't get involved in it!

Note: leaving your work on a CIS lab computer creates a possibility for someone else to copy it – in which case you will find yourself in the position of being considered to be involved in plagiarism/cheating, and with no way to prove otherwise. Therefore, I recommend saving all work on the computer's hard-drive, transferring it to a floppy or zip disk when finished – and then deleting whatever you left on the hard-drive.

Another View of Collaboration and Plagiarism/Cheating:
One of the purposes for seeking a university education is to receive a fair evaluation of your performance in scholarly endeavors. (Others often use those evaluations to make judgments regarding your suitability for employment or your opportunities for further education.) If you present someone else's work as your own, that evaluation is worthless.

Another purpose of a university education is to enhance your ability to think clearly – abstractly, analytically, critically, productively, and so on – which requires an effort on your part. If you do not make that effort, you do not get the experience from which you can learn more efficient and effective modes of thought.

Another point about the work you do in school: one of the benefits of doing all work yourself is the confidence you acquire in your abilities. That confidence is what will enable you to go beyond the limits of what you know how to do, and apply your knowledge to solve problems you haven't attempted before. That is essential for the success we want you to have in the workplace, and it can only be acquired through experience.

Yet another benefit of a university education is the opportunity to interact with highly intelligent people (the "professors") and observe how they have organized the knowledge of their field. Acquiring an organized body of knowledge through such an interaction will save you years of experience and discovery. Plagiarism and cheating are indications that you are not interested in that.

If you involve yourself in cheating, you are depriving yourself of the benefits of being in a university community; you are subverting your potential for future successes in life and/or academia; and in general you are indicating you are not interested in doing what is done at a university.

If you are not interested in doing what university students do here, then there is no point in remaining here.

Besides: you are paying for a university education – get your money's worth!

Drops and Incompletes:
Be aware that drops and incompletes often have serious financial aid implications. Be sure you understand them before requesting to drop a course or request a grade of incomplete. See above for the dates of various deadlines. In particular, note the last day to drop with a refund, and the last day to drop without a refund.
The Computer Science Department follows the University policy on drops (explained on page 19 of the catalog or at http://www.umt.edu/catalog/acpolpro.htm#5): a drop after the deadline must involve (1) registration errors, (2) accident or illness, (3) family emergency, (4) change in work schedule, or (5) no assessment of performance in the class occurred before the drop deadline.

Comments regarding floppy disks:
(1) CS Department computers sometimes have a problem with floppy disks – and the contents of the disk can be destroyed. So be aware of that when planning to use a floppy drive in SS419.
(2) In CIS labs:
-- Wait for the green light above the disk drive to go off before removing a floppy disk from the drive.
-- 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 the floppy disk(s) once, just before leaving the machine.