

Instructor: Dr. Ben Keller
625 McBryde Hall
231-9367
keller@cs.vt.edu
Office Hours: 10:00-11:00 MWF
9:30-10:30 TTh
CRN: 11373, 114 Holden
GTA: Savyasachi Samal
ssamal@vt.edu

GTA office hours will be held in 124 McBryde (the Unix lab). The schedule will be posted on the course website.

Objectives: This course provides an in-depth study of the issues in the design, implementation, and application of programming languages. Topics will vary from basic to advanced in areas such as syntax, semantics, binding, data abstraction, exception handling, concurrency, and functional, logic and object-oriented programming.

Prerequisites: CS 2604. Computer Science majors and minors must have completed CS 2604 with a final grade of C or better. You will find aspects of the course easier if you have been through ECPE 2504, but this is not necessary for the course.

Text: Michael L. Scott, *Programming Language Pragmatics*, Morgan-Kaufman, 2000.
Please check the author's website for errata of the book.

Course Web Page: The course web page will be the primary source of information regarding assigned work in class. You should check the web page frequently. The web page is <http://courses.cs.vt.edu/~cs3304/spr01/>.

Course Activities: The primary activities in this course will be homework, and projects.

- Homework — There will be approximately one homework assignment per week, except for the week of the midterm (for a total of 13). A homework assignment may consist of questions (possibly from the book) requiring written answers or programming solutions.
- Projects — We will have two kinds of programming assignments: (1) programming in a specific language, and (2) programming language implementation projects. We will program in three languages: Standard ML (SML), Prolog, and Eiffel; implementation projects will be in C++ and/or SML. There will be five projects.

Grading and Evaluation: Your grade in this course will be based on evaluation of the homework, projects, handbook, a midterm test and a final exam. The final grade for the course will be determined using the following distribution.

Homework	20%
Projects	40%
Midterm test	15%
Final exam	25%

- Homework — All homework solutions must be submitted electronically to the curator system as either a Word or PDF document unless otherwise required by the assignment. In some cases, homework may involve programming and may be autograded by the curator system. In most cases, however, the GTA will grade the homework solutions.
- Projects — All projects will be submitted electronically to the Curator system. When possible projects will be auto-graded for correctness, but may also be evaluated for style, design and other factors. Any additional grading will be computed as a deduction from the correctness score given by the Curator. In some cases programming assignments may be evaluated by demonstrating to the GTA. In both cases, grade sheets will be mailed by the GTA to each student. (The programming standards used for evaluation are given on the web page.)
- Examinations — The midterm test will cover roughly half of the material in the course and is tentatively scheduled for 25 February. The final exam will be given on 6 May from 2:05 to 4:05 p.m. The final exam will be comprehensive.

Grading Policies: The follow policies are the rules by which we will operate this semester.

- Missed work — No make up work will be allowed for the course. If you have a formal excuse from a dean (either the Dean of Student's office, or the dean of your college), your final exam grade will be substituted for your midterm grade.
- Late work — Only projects may be submitted late, and then for a penalty of 10% per weekday up to a maximum of three days late. Saturday or Sunday submissions will be counted as a submission on the following Monday, and so will have a one day penalty. The time that the Curator records for the submission will be the official submission time, and while we will be liberal in interpreting the exact due date, you should not rely on this. No other work will be accepted late.
- Hardware/software failure, power failure, etc. — No extensions will be given for failure of your personal hardware or software. Extensions will also not be given for problems with CS lab machines. Extensions may be given for widespread outages of power or networks, but are not likely.
- Autograding — projects which are autograded for correctness by the Curator may also be evaluated for quality by the GTA. The quality grade will be deducted from the correctness grade to compute the project grade. If you are allowed more than one

submission, the first submission with the highest correctness grade will be evaluated for quality. Be aware: comment your code before submitting, and do not use the Curator to debug your program! No exceptions will be granted!

- Demonstration grading — some projects may be evaluated by demonstration to a GTA. Demonstrations must be given within two weeks of the due date, if a project is not graded within that time period, it will not be evaluated. A project that is not demonstrated will not be evaluated.
- Compiler restrictions — All program grading will be done on Mandrake Linux. Compilers or interpreters required for specific programming assignments will be specified in the corresponding assignment. The only place where the use of Linux may affect your choice of OS is when we have C++ assignments.
- Programming Standards — Simply having a program that produces the correct results is not sufficient in this course; it must also correctly implement the data structure, adhere to the programming standards of the course, and be well designed. The programming standards are given on the course web page. These standards apply to all projects, unless specifically stated in the assignment.
- Appealing grades — Questions or complaints about grading should be addressed to the person who graded the work; however, the instructor is the ultimate authority on grading policy for the course.
- Appeal Deadline — Any questions or complaints regarding the grading of an assignment or exam must be raised within one week of the return of the graded work by the instructor or GTA. This is not necessarily the same as when you pick it up.
- Final grades — Letter grades will be assigned according to the usual 10-point scale: with 90% guaranteeing an A-, 80% guaranteeing a B-, and so forth. A curve may be applied to determine the final letter grades at the instructor's discretion.

Honor Code: *All graded assignments must be your own work.* The instructor and GTA are available for assistance on completing solutions to homework and programming assignments. Assistance from anyone else regarding the specifics of your solution is a violation of the honor code.

A major portion of this course is involved in learning and applying new programming languages. You are allowed to, and should, work together to help understand how to use these languages and the programming environments used in class. At no point, however, should this involve showing your solution to someone else, or looking at someone else's solution to an assignment.

Special Accommodations: Any student requiring special accommodations is asked to meet with the instructor in the first week of classes.