Course Description

This is an entry course for CS majors and others with a serious interest in computer programming in which you will begin to learn to program computers. No prior exposure to computers or programming is required.

It is a course introducing the science of programming, and as such will concentrate heavily on the intellectual process of designing programs and verifying their correctness, since all modern large-scale programming demands considerable skill in this regard. You will also learn to operate some of the Department’s Unix computer systems along the way, along with details of the computer programming language called C++. Because of the emphasis on the underlying principles, much of the content of the course will be applicable to programming any computer using any of a variety of computer languages.

We will cover most of the material in the first eleven chapters of the Shiflet textbook. You will use the Anderson book to help you become familiar with the operation of our departmental computers.

If you have a C++ compiler at home you are encouraged to use it for additional programming practice, and you may find it useful for working on your individual assignments. However, you are required to attend the lab sessions in our

Departmental Lab, and use our machines for the laboratory assignments.

See the class web page for a links related to Linux, which you can install on your home PC to achieve a similar programming environment to our lab. The web page also links to an excellent free C++ programming environment from Cygnus Software which runs under Windows.

Schedule

We will normally meet in the lecture classroom on Fridays and Mondays, and in the computer laboratory on Wednesdays. Exceptions to this general rule will be announced in class.

Grading

Your grade will be determined from quizzes and exams, individual programming projects, and weekly laboratory work. Semester grades will be based upon the standard ABCDF system, without pluses or minuses.

Lab Work

Science, at its best, is an open and collaborative activity. International journals in every scientific discipline communicate theories and results, both for the benefit of the authors and fellow scientists. There is a strong tradition of preferring to work together, and secrecy
required by political or economic reality is
generally tolerated rather than welcomed. In
this course, we will model this tradition in the
weekly laboratory assignments.

You will do your weekly laboratory work in
collaboration with lab partners chosen at the
beginning of the term. You should feel free to
share your knowledge with your lab partners,
study together, and work together with them to
solve the laboratory problems from this course
(but not the individual projects—see below).

Each laboratory team will turn in one copy
of each week’s work, which will be credited
equally to all of the members of the team who
worked on the assignment. These will be graded
on the basis of completeness and timeliness, and
the total will count as 20% of your final course
grade.

Quizzes

As you work with teammates on your
laboratory work, you should take the respon-
sibility very seriously to be an informed and
contributing member of your lab team.
Together, you should treat the laboratory period
as a time for experimentation and learning about
how computer programming works. Obviously,
the learning part of that is necessarily each in-
dividual student’s responsibility. To test this,
ocasionally there will be unannounced quizzes
based on the laboratory work. These will be
given during the classroom segment of the
course, one or two class meetings after the lab to
which they apply. At the end of the semester, I
will discard each student’s lowest quiz score,
and the average of the remaining quizzes will
count as 15% of your final grade.

Individual Projects

Unfortunately, in the real world and espe-
cially in capitalistic societies, science is some-
times constrained by political or economic pres-
sures to keep certain results secret. In the class-
room this happens because of the need to assign
individual grades. Breaches of this secrecy in
industry may result in fines, imprisonment, and
loss of job and reputation. In the political arena
it (espionage) can be punishable by death! In
the academic world we also take this problem
very seriously, although our punishments are
thankfully less harsh: representing someone
else’s work as your own is called plagiarism,
and may be punishable by a failing grade or
suspension or expulsion from the university.

In contrast with the lab work, the individual
programming projects must be your own work.
Also, whereas lab work is expected to be some-
what experimental in nature, the individual
projects should be polished finished products.
They will be graded on the basis of complete-
ness, correctness, clarity, and promptness. To
obtain the highest grade, your program must be
free of “bugs”: it must not only produce com-
plete, correct output under any possible set of
conditions, but the logic must be so clear that it
is plain from the listing that this will be the case.
This clarity means it must be logically or-
ganized, effectively annotated with comments,
and generally easily understood by anyone who
understands C++. Finally, it must be turned in
at or before the appointed time. Programs sub-
mitted late will be graded down severely, and no
credit at all will normally be given for any
assignment which is five or more days late.

The aggregate individual project grade will
count as 25% of your final course grade.

Exams

There will be two exams: one midterm given
in approximately the sixth or seventh week of
classes, and one final, given during the
scheduled final exam period. Each exam will
count as 20% of your final course grade. Exams
will cover all of the course material covered as
of the date of the exam, with emphasis on
material covered subsequent to previous exams.
Questions will typically require short prose
answers, where it is understood that you might
be asked to answer in English, pseudocode, C++
or some combination of these.