1. Basic Course Information

1.1. Objectives:
CS250 introduces the programming techniques required to scale up to larger problems than are treated in the introductory programming course. Topics include the software life cycle, methods of functional decomposition, design documentation, abstract data types and classes, common

1.2. Required Text: (5th or 6th edition is good for this semester)
data structures, dynamic data structures, and algorithmic patterns, and testing and debugging techniques. Students must be highly disciplined and motivated in the art of self-learning. This includes being able to efficiently read, absorb, and retain new information, as well as being fully capable of studying and understanding new concepts and technical ideas in a timely and independent manner. Students must be highly disciplined and motivated with respect to time management, and spending time working on assignments and course materials. This course is demanding, and may require a significant amount of time writing code, examining code, and fixing code.

1.2. Required Text: (5th or 6th edition is good for this semester)

  - This book is frequently sold with accompanying CDs. We won't use them, so don't worry about it. If you can get a used copy that is missing the CDs, or if you can get a better price on a new one without the CDs, go for it!
  - Not available through the bookstore, but you can order it.

2. Course Pre- and Co-requisites

Students must register for a lecture section, a lab section, and a recitation section of CS250.

The prerequisites for this course are:

- CS 150, Problem Solving and Programming I,
- Math 162M, Pre-Calculus I

or equivalents.

The co-requisite for this course is:

- CS 252, Introduction to Unix for Programmers

(A co-requisite is a course that must be taken before or during the same semester as this course.)

Although CS252 is, itself, a self-paced course, students in CS250 will be expected to have completed specific portions of it in preparation for some laboratory sessions, as indicated on the course Topics page.

3. Assignments

3.1. C++ compiler
Assignments for this course will include laboratory assignments and a design project. Both may involve programming in C++.

### 3.1. C++ compiler

The “official” compiler for this course is the Free Software Foundation's g++ (also known as gcc or GNU CC), version 4.5 or higher. This is the compiler that the instructor and/or grader will use in evaluating and grading projects. If you have access to other compilers, you may use them, but you are responsible for making sure that their projects can be compiled by the instructor and/or the course's grader using the official compiler.

You may want to develop your programs on the most convenient compiler and then port it over to the official environment. Please don't underestimate the amount of time that may be involved in coping with subtle differences among compilers.

You can do all work in this course using g++ on the CS Dept Unix servers via ssh/X. If you like, however, you can obtain the g++ compiler for free from a variety sources. Links will be provided on the course Library page.

### 4. Communications

#### 4.1. The Forum (may not be used this semester)

#### 4.2. Contacting the Instructors

#### 4.3. Office Hours:

**4.1. The Forum (may not be used this semester)**

The course website may include an online Forum, as described in Navigating The Course Website. Most course-related questions should be asked in class, during our regularly scheduled lectures. Office hours precede our lectures for your convenience. Email should be used only in cases where the question cannot be asked in class. (e.g., for help in entering the course website).

**4.2. Contacting the Instructors**

Chris Boyle  
Dragas 1100D  
683-6001 ext. 6289  
cboyle@cs.odu.edu

**4.3. Office Hours:**

Office hours for the instructor can be found on their respective home page. (Click the links on their names above).
5. Course Policies

5.1. Due Dates and Late Submissions:

Unless explicitly stated otherwise, assignments and project work are due by 3:00PM on the due date listed in the course schedule. For weekly assignments, this means that you must have submitted the assignment before the first lab session for the coming week (even if that is not the lab section for which you are registered).

Late assignments and make-up exams will not normally be permitted.

Exceptions will be made only in situations of unusual and unforeseeable circumstances beyond the student's control, and such arrangements must be made prior to the due date in any situations where the conflict is foreseeable.

"I've fallen behind and can't catch up", "I'm having a busier semester than I expected", or "I registered for too many classes this semester" are not grounds for an extension.

Extensions to due dates will not be granted simply to allow "porting" from one system to another. "But I had it working on my home PC!" is not an acceptable excuse.

To be clear: Late assignments and make-up exams are not permitted.

5.2. Academic Honesty:

Except where explicitly stated otherwise, everything turned in for grading in this course must be your own work. Some laboratory assignments may be done, in part or in whole, by teams of two students, in which case the submitted material must be the work of only those team members. Such assignments will contain clear and explicit instructions identifying them as team assignments. In the absence of any such explicit statement, any graded work must be performed by the single individual who submits it for grading.

The instructor reserves the right to question a student orally or in writing and to use his evaluation of the student's understanding of the assignment and of the submitted solution as evidence of cheating. Violations will be reported to Student Judicial Affairs for consideration for punitive action.

By CS Dept. policy, students found to be in violation of this rule will, at the very least, receive a failing grade in the course and may be subject to stiffer penalties. Students who contribute to violations by sharing their code/designs with others may be subject to the same penalty. Students
are expected to use standard Unix protection mechanisms \( \texttt{chmod} \) to keep their assignments from being read by their classmates. Failure to do so will result in grade penalties.

This policy is not intended to prevent students from providing legitimate assistance to one another. Students are encouraged to seek/provide one another aid in learning to use the operating system, in issues pertaining to the programming language, or to general issues relating to the course subject matter. Student discussions should avoid, however, explicit discussion of approaches to solving a particular graded assignment, and under no circumstances should students show one another their code for an ongoing assignment, nor discuss such code in detail.

5.3. Attendance:

Unless explicitly stated otherwise, attendance at lectures is generally required. Students are responsible for all material covered and announcements made in class. Consequently, if you are going to miss class, be sure to get notes, handouts, etc., from another class member. If you miss your lecture section, you may attempt to attend the other instructor's lecture. However,

- your ability to do so is subject to availability of seating in the classroom.

Each week, the course calendar will show a lab, an assignment, or both. Unless explicitly stated otherwise, attendance at laboratories is required. Lab activities are not always graded, but are often intended as an introduction to or practice in techniques that you are expected to employ in the week's assignment. Failure to attend may result in point deductions from the week's assignment.

If you miss your lab section, you may attempt to attend a later one. Again, however, your ability to do so is subject to availability of seating and machines in the lab. If too many people show up, the TAs will insist that students registered for that section get preference.

Recitations will be used primarily to review the solution to the prior week's assignment and to provide help on the assignment for the current week.

5.4. Grading:

Grades are calculated EXACTLY using the percentages in the table shown below.

<table>
<thead>
<tr>
<th>Weekly Assignments:</th>
<th>35%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Project:</td>
<td>20%</td>
</tr>
<tr>
<td>Exam One:</td>
<td>10%</td>
</tr>
<tr>
<td>Exam Two:</td>
<td>10%</td>
</tr>
<tr>
<td>Exam Three:</td>
<td>25%</td>
</tr>
</tbody>
</table>