Instructor and Course Information

William D McQuain
Email: wmcquain@cs.vt.edu
Office: 631 McBryde Hall
Office Hours: see course website

CS 2606  Data Structures and OO Development II

Design and implementation of data structures, intermediate software engineering design principles, and object-oriented programming skills.

Emphasis on algorithm analysis, design patterns, testing, debugging, and organizing and managing larger programs.

Topics include sorting, searching, file processing, indexing, hashing, algorithm analysis, and advanced tree structures.
## Course TAs

<table>
<thead>
<tr>
<th>Srinivasa Santhanam</th>
<th>Mahima XXX</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Email:</strong> <a href="mailto:ssantha@vt.edu">ssantha@vt.edu</a></td>
<td><strong>Email:</strong> <a href="mailto:mahi83@vt.edu">mahi83@vt.edu</a></td>
</tr>
<tr>
<td><strong>Office:</strong> TBA</td>
<td><strong>Office:</strong> TBA</td>
</tr>
<tr>
<td><strong>Office Hours:</strong> TBA</td>
<td><strong>Office Hours:</strong> TBA</td>
</tr>
</tbody>
</table>
Prerequisites

CS 2605  Data Structures and OO Development I

All students must have completed the prerequisite with a grade of C or higher (C- is not acceptable).

Note: students are expected to have prior proficiency in the C++ programming language, including the design and implementation of object-oriented systems and linear and binary tree data structures using C++ templates.

There will be absolutely NO exceptions to these requirements.

Note  Lying about whether you meet the prerequisites is an Honor Code offense. Students who do so may be charged, at the discretion of the Department of Computer Science.
Text and Recommended References

Required:

Data Structures and Algorithms in C++, 3rd Edition
Adam Drozdek, Brooks/Cole, ©2005
ISBN 0-534-49182-0

Recommended:

CS 2606 Course Notes, Spring 2007 Edition
W D McQuain, ©2001-2007
(available ONLY at the course website)

courses.cs.vt.edu/~cs2606/Spring07/
Graded Work

Final grades will be based on the average achieved over the following:

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight</th>
<th>Tentative Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Projects</td>
<td>44%</td>
<td>See website</td>
</tr>
<tr>
<td>Homework</td>
<td>10%</td>
<td>See website</td>
</tr>
<tr>
<td>Quizzes</td>
<td>16%</td>
<td>Frequent</td>
</tr>
<tr>
<td>Midterm Test</td>
<td>10%</td>
<td>Thursday, March 1</td>
</tr>
<tr>
<td>Final Exam*</td>
<td>20%</td>
<td>07:45 Wednesday May 9</td>
</tr>
</tbody>
</table>

Grade Scale

The usual 10-point scale will apply (subject to any curve). A final average of 90% will guarantee an A-, 80% will guarantee a B-, and so forth.

* Exam score will replace the midterm score, if it is higher.
Grade Policies

Grade Scale

The usual 10-point scale will apply (subject to any curve). A final average of 90% will guarantee an A-, 80% will guarantee a B-, and so forth.

Curve

A grade curve may or may not be employed in this course. The application of a curve is dependent upon class performance on tests, projects and homework. The decision to utilize a curve rests entirely with the course instructor.

Statute of Limitations

Any questions about the grading of an assignment must be raised with your instructor within two weeks after the graded assignment has been made available to you.
Late Work

Late Days for Projects

Each student is allotted 6 (six) virtual late vouchers. Each late voucher may be used to "purchase" an extra day (relative to the due date) to turn in a programming project without any penalty.

Late submissions of a project will not be evaluated unless the student has enough late vouchers to "purchase" the necessary amount of extra time that was used.

Late vouchers will be applied automatically, as needed. Unused late vouchers simply expire at the end of the term; there will be no extra credit for them.

Extensions

Any request for an extension must be made, preferably by email, at least 24 hours prior to the due date. Reasons must be valid (generally events beyond your control) and documented.

Late submissions will not be given any credit if submitted after graded assignments or solutions have been released.

Homework

Usually, no late submissions will be allowed for homework assignments.
Evaluation of Correctness, Design and Implementation, and Documentation

The correctness of operation of your programming projects will be evaluated by executing your solution with test data constructed by the course staff. While some test data will be provided, there is no guarantee that data will cover all cases.

Each project will have certain explicit requirements for design and implementation. Your solution will also be evaluated for adherence to those requirements.

You are expected to enter CS 2606 with considerable understanding of good software engineering practice, and you are expected to apply those lessons here. That means that you may be penalized for failing to make good decisions, even if there are no explicit guidelines in the specification of a project.

You are also expected to incorporate professional internal documentation into your projects. See the Programming Standards page on the course website for some suggestions and samples.
Test Environments

- All programming assignments submitted are required to compile under g++ version x.y.z running on Windows XP or Windows Advanced Server 2003, as installed in the McBryde 116/118 lab.
- Unless specified otherwise, programs will only be tested under that environment.
- It is the YOUR responsibility to ensure that YOUR programs execute correctly in the appropriate environment; programs that do not will receive substantial deductions.

Students developing with another C++ compiler are warned that there are many pitfalls and that they will be given NO compensation for those pitfalls.

Compliance with the ISO C++ Standard varies widely among older compilers, especially g++ prior to version 3.2.

Your programs WILL be tested with the environment listed above. If it fails to compile, or exhibits incorrect behavior, we don’t care that it may compile elsewhere, or appear to run correctly elsewhere.
Sources of Help for This Course

General Issues
- CS 2606 classmates
- CS 2606 Forum online at forum.cs.vt.edu
- CS 2606 TAs
- CS 2606 Instructors

C++ Language Help
- CS 2606 Forum
- texts from earlier courses
- alt.comp.lang.learn.c-c++
- gnu.gcc.help, gnu.g++.help

Lecture Instruction
Lectures will consist of presentations, applications, problems and solutions interspersed with classroom discussion.
Damage Control

Backups

**Students are responsible for making backup copies of all their work in this (and all) courses.**

Loss of work due to hard drive failure is **NOT** an acceptable excuse. Backup copies of files on the same hard drive are not backup copies. Backup copies of files on second hard drives are also risky. Backup copies should be maintained on two separate distinct storage mediums, (e.g., hard drives and Zip disks).

Backup copies should be maintained until after the end of the term and students have received their course grade. (The Army lives by triplicate for a reason.)

Remember: Computer systems are mechanical devices.

- Network connections break.
- Plan for it. It is inevitable!
Honor Code

An exhaustive list of Honor Code violations would be impossible to present here, but among other things, each of the following is a flagrant violation of the Virginia Tech Honor Code, and violations will be dealt with severely (Honor Court):

- Working with another student to derive a common program unless group work is explicitly allowed on that assignment.
- Discussing the details required to solve a programming assignment. You may not share solutions.
- Copying source code (programs) in whole or in part from someone else.
- Copying files from another student's disk even though they might be unprotected.

It is acceptable to discuss with classmates a programming assignment in a general way, i.e., to discuss the nature of the assignment. In other words, you may discuss with your classmates what your program is required to accomplish but not how to achieve that goal using C++. In no way should the individual statements of a program or the steps leading to the solution of the problem be discussed with or shown to anyone except those people cited in the following statement.
Honor Code

Feel free to discuss the homework assignments and your program source code with the teaching assistants assigned to CS 2606, the instructor, or the free tutors provided by UPE. The discussion of your program source code must be limited to these people. Note that this specifically excludes discussions of your program source code with other students (even if they are not enrolled in CS 2606), or with tutors except for those named above. Privately hired tutors are not an exception to this requirement, nor are athletic or other tutors provided by the University.

Copies of all submitted work are retained indefinitely by the Department. Submitted programs are subjected to automated analysis for detection of cheating.

If you have any question as to how the Honor Code applies to this class, remember that:

• Any work done in this class must be done on an individual basis.
• Credit will be given only for work done entirely on an individual basis.
• Do not make any assumptions as to who can provide help on a programming assignment.
• All submitted work is archived. All submitted programs will be subjected to automated cheat analysis.

Evidence indicating the violation of the policies stated above will be submitted to the Honor Court.

It is much easier to explain a poor grade to parents or a potential employer than to explain an Honor Court conviction.