Welcome to CS 2704

CS 2704
Object-Oriented Software Design and Construction

Fall 2000
D Barnette
Dr. Ben Keller

"There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult."

-C.A.R. Hoare

8/20/2000

Virginia Tech © 2000
Table of Contents

- Syllabus: Instructor Info
- Syllabus: Description & Texts
- Syllabus: Evaluation
- Class Organization
- Development System
- Trademark Notice
Instructor:

Instructor: N Dwight Barnette
Email: barnette@vt.edu
Office: 624 McBryde Hall
Office Hours: 9:00 – Noon Mondays
4:00 – 6:00 Mondays
and by appointment

Course Website: http://courses.cs.vt.edu/~cs2704/

Course Description

Credits: 3
Prerequisites: CS 1704 > C for CS Majors/Minors
ECPE 2574 ≥ C- for ECPE Majors
There will be absolutely NO exceptions to these requirements.

Purpose:

Introduces the principles of object-oriented programming with emphasis on objects, classes, inheritance, and polymorphism. A programming language such as C++ is used to apply these principles in several application domains. Also stressed are tools and techniques for testing, debugging, and organizing and managing larger programs.
Texts:

Required:


Recommended:


Other Useful References:


Evaluation and Grading:

Point Distribution

Final grades will be based on the number of points achieved over the following:

<table>
<thead>
<tr>
<th>Item</th>
<th>Weight</th>
<th>Tentative Dates</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Testing and SE</td>
<td>50%</td>
<td>Check Course Web site.</td>
</tr>
<tr>
<td>Homework/Quizzes</td>
<td>5%</td>
<td>TBA</td>
</tr>
<tr>
<td>Test 1</td>
<td>8%</td>
<td>September 26</td>
</tr>
<tr>
<td>Test 2</td>
<td>12%</td>
<td>November 8</td>
</tr>
<tr>
<td>Final Exam</td>
<td>25%</td>
<td>Tuesday, Dec. 12, 1:05–3:05 CTE</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.

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 and homework. The decision to utilize a curve rests entirely with the course instructor. If a curve is applied, the Krider Curve will be used; see the course Website for details.
**Krider Curve**

- The Krider Curve was developed by Dr. Daniel W. Krider, Prof. of Mathematics, Concord College, Athens, WV. The curve coerces grades into forming a distribution which more closely resembles a Normal Distribution. This is a partial solution to the problem of class grades tending to be skewed toward the lower values. The Krider Curve forces symmetrical clustering about the mean (property of central tendency) and smaller variance within the scores (less dispersion).

\[ f(x) = x + \alpha(100 - x) \]
Application

- Variables:
  - Given the following:

  \( x \leftrightarrow \text{student’s grade} \)
  \( y \leftrightarrow \text{student's curved grade} \)
  \( \alpha \leftrightarrow \text{curve percentage} \)

- The Krider formula, \( f(x) = x + \alpha(100 - x) \), is applied to each individual student grade.

- For example, if \( \alpha = 1/3 \) then

  \[
  y = x + \frac{1}{3} (100 - x)
  \]

- This is equivalent to averaging two X values with one grade of a 100. The lower scores receive a larger curve than the upper scores. An appropriate action when one considers that poorer students require more help. The curve ensures that a student will never surpass another student who scores higher.
Alpha Derivation

- Alpha can be set to achieve a desired class mean.

Given the following

\[ \bar{x} \iff \text{class mean} \]
\[ \mu \iff \text{desired mean} \]
\[ n \iff \text{class size} \]

The formula for deriving alpha’s value for a desired mean:

\[
\mu = \bar{x} + \alpha \left( 100 - \frac{n}{n} \right) - \frac{\sum_{i=1}^{n} x_i}{n}
\]

\[
\mu = \bar{x} + \alpha \left( 100 - \frac{\sum x_i}{n} \right)
\]

\[
\mu = \bar{x} + \alpha \left( 100 - \bar{x} \right)
\]

\[
\alpha = \frac{\mu - \bar{x}}{100 - \bar{x}}
\]
Sources for Help/Questions etc.
- CS 2704 Classmates:
  CS 2704 Listserv & USENET news group
- CS 2704 GTAs
- CS 2704 Instructor

Microsoft C++ Help

**CS 2704 USENET Newsgroup**
† Connect to: vatech.class.cs2704
† CS 2704 TAs & others will respond to questions.
† Other Newsgroups: csugrad.pc-clone, csugrad.programming.c, comp.lang.c, comp.lang.visual
† A panel of "experts" will respond to questions.

**CS 2704 ListServ**
† Connect directions will posted online
† CS 2704 classmates, TAs & others will respond to questions.

Lecture Instruction

Lectures will consist of presentations, applications, problems and solutions interspersed with classroom discussion.
Test Environments

- All programming assignments submitted are required to compile under either Microsoft Visual C++, version 6.0 or the GNU g++ compiler installed on the FreeBSD machines in McB 124.
- Programs will be tested under either Windows NT or FreeBSD.
- It is the student’s responsibility to ensure that his/her programs execute correctly in the appropriate environment.
- Points will be deducted for programs not meeting this requirement.
- GTAs do not have time to go to dorms, etc. to grade programs on specific systems.

Program Demonstrations

- Demonstrations will take place in the McBryde CS Dept. Computer Lab for some assignments.
- Students may not bring their systems to the labs to execute their programs upon.
Backups

- **Students are responsible for making backup copies of all their work in this course.** 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 floppies).

- 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. Systems fail. Plan for it. It is inevitable!
Company Trademarks

Microsoft, Windows, MS, MS-DOS are registered trademarks™ of Microsoft Corporation.

Microsoft Visual C++ is a registered trademark™ of Microsoft Corporation.

IBM, OS/2 are trademarks™ of International Business Machines Corporation.

Adobe, Acrobat and Postscript are trademarks™ of Adobe Systems Inc.

UNIX is a trademark™ of UNIX System Laboratories.

Internet Explorer is a trademark™ of Microsoft Corporation.

Netscape Navigator is a trademark™ of Netscape Communications Corporation.