

Course Introduction

Larry Caretto
Computer Science 106
**Computing in Engineering
and Science**

January 31, 2006

Outline

- Discuss course outline
 - See full outline on course web site
<http://www.csun.edu/~lcaretto/comp106>
- Assessment quiz (not graded)
- Computers and computer programming
- Today's laboratory will introduce the C++ programming environment you will use for the course

Power Point is Boring

- Some students have this opinion
 - It is true if you do not participate
- Handouts of slides, downloaded in advance, allow you to listen and think without having to write notes
- Some slides are have questions with time for everyone to answer
 - Thinking about answers, even if you do not get correct answer will help you learn

Course and Instructor

- Course lecture and lab in JD 1600
 - Lecture section (class number 11672) TTh 12:30 – 1:20
 - Laboratory section (class number 11673) TTh 1:30 – 2:45
- Larry Caretto (lcaretto@csun.edu)
 - Office (JD 3333, 818.677.6448)
 - Office hours: MW 3:15 to 3:45 pm and Wed 6:14 to 6:45 pm; other times by email, drop-in, telephone, or appointment

Email

- Campus policy requires students to monitor their CSUN email addresses
 - These addresses will be used class email list comp106-c@csun.edu
- Setup account and, if desired, forward email to another address
 - <https://www.csun.edu/helpdesk/stuact.html>
 - <https://www.csun.edu/account>

Textbook and Cheat Sheet

- Tony Gaddis, *Starting Out with C++*, (fourth edition) Scott/Jones, 2004
 - Contains CDs with compilers for home use
 - “Update” edition with “backpack” is same book, but more expensive
- A C++ study guide such as the one produced by BarCharts, Inc.® is allowed for closed book exams

Course Objectives

- Learn basic programming concepts in a higher level language
- Learn how to write computer programs in C++ using Visual Studio .Net
- You will be focused on the second objective during the course, but you should be learning concepts that can be applied to other languages

California State University Northridge 7

Class Schedule

- See online course outline
 - Schedule of topics and assigned reading, midterm and final
 - Schedule of laboratory assignments, quizzes, and due dates for assignments
 - Late assignments are penalized 10% of maximum grade for every week or part of a week they are late
 - No late assignments accepted after May 26 (Friday of final exam week)

California State University Northridge 8

Course Grading

- You will receive a single grade for this integrated lecture--laboratory course
 - The CSUN grading system requires a separate grade for laboratory and lecture, but both grades will be the same
 - There is no absolute grading scale
 - Grading is based on assessment of student learning
 - Plus/minus grading will be used

California State University Northridge 9

Grading Criteria

A: Knows almost all of the course material and is able to apply it to new tasks.

B: Satisfies one, but not both, of the conditions for an A grade.

C: Knows fundamentals of the course and is able to apply this knowledge to routine algorithms and programs.

D: Has learned some course material but is not able to apply all the course fundamentals

F: Has failed to demonstrate knowledge of the course material beyond a minimal level.

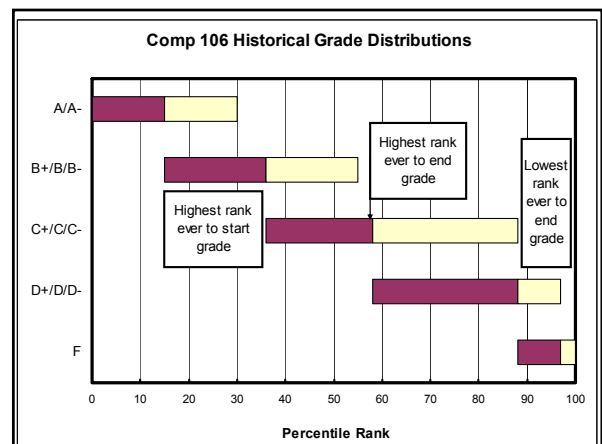
California State University Northridge 10

Historical Grade Distributions

Percentile	0-12 %	12-30 %	30-36 %	36-55 %	55-58 %	58-88 %	88-97 %	97-100 %
Grade Range	A/ A-	A to B	B+/ B/ B-	B to C	C+/ C/ C-	C to D	D to F	F

- Overall rankings will be prepared and distributed during course
- Distribution for this class may not be in ranges shown

California State University Northridge 11



Basis of Grading

- Homework (with posted solutions) will be assigned but not collected or graded
 - Exam and quiz questions are like these homework problems
- Laboratory assignments
 - First four programming exercises 5%
 - Second four programming exercises 10%
 - Three programming projects 11%

Grading Basis Continued

- Three laboratory quizzes (30, 60, and 90 minutes) (4%, 8%, 12% -- total 24%)
- One midterm exam 20%
- Final exam 30%
- Students found cheating will receive an F in the course and the incident will be reported to the dean of students for possible disciplinary action

Working Together

- Teamwork for mutual learning is good
- Do not just do what someone tells you to do without understanding
- Submitting someone else's work as your own is a violation of the student conduct code
- You can form programming teams of 2 to 3 students to hand in a team program
 - Notify instructor in advance
 - One submission and one grade for all students on the team

Class Conduct

- Show respect for your fellow students and the instructor
 - Come to class on time and do not leave early
 - Turn off cell phones and pagers
 - Do not disturb others during lecture
 - Ask instructor if you have a question
 - Participate in class discussions and try to answer questions posed to class

Laboratory Sessions

- Start with lecture – do not use computers during this time
- Lab assignment instructions, supplemented by lecture presentations, both available on line
- No lab assistant – sometimes there is a long queue of students who need help
 - Get help from other students or come to ask me for simple questions

Laboratory Assignments

- Work during laboratory periods, open lab hours, or at home
 - Assignments may be turned in early
 - You can start assignments ahead of the lab schedule in the course outline
- Submit printed copy or email by 11:59:59 pm on due date
 - Submit hard copies in class or in JD 3333
 - Place under door if instructor not in office (assignments found next day assumed OK)

Initial Assignments

- Reading assignments
 - Pages 1-22 for today
 - Pages 29-45 and 83-92 for Thursday
 - Pages 45-61 and 100-106 for February 7
- This week's homework (February 8)
 - Pages 25–26, problems 33, 34, 35, and 36
- First programming exercise (on learning Visual C++) due Thursday, February 2

Initial Assessment

- Information on student background and ungraded quiz on simple concepts in mathematics and computing
- Will take 5 to 10 minutes
 - Turn in page when finished
 - One minute warning before collecting all
- Will then have brief lectures on computing and first laboratory

Computer Hardware

- Central processing unit (CPU)
 - Program controller interprets and executes instructions in stored program
 - Arithmetic logical unit does necessary calculations and logical comparisons
- Central memory (random access memory or RAM)
 - “Random” access means that we can access any memory location

Computer Hardware II

- Long-term memory
 - Hard drive
 - Portable memory devices – disks, portable drives, solid-state drives
- Input-output devices
 - Keyboards, mice, touch screens
 - Infrared, wireless and network connections
 - Terminal screens, printers and plotters
 - Computer control system signals

Computer Operations

- The operating system (Mac OS X, Windows XP, Unix, Linux, etc.) allow a user to perform basic instructions on a computer
 - Manipulate information stored as files in the computer storage
 - Start execution of one or more programs
- Assume Comp 106 students familiar with graphical user interface (GUI)

Programming

- A program is a set of instructions that direct the actions of the computer
 - Stored in memory and executed by program controller
 - Actual program executed by computer is a series of machine language instructions
 - Programmers write code (called source code) in more understandable languages, like C++, called higher-level languages

Programming Operations

- An editor (simplified word processor) is used to write and edit the (C++) program
- A compiler translates the (C++) program into object code
- A linker creates the final executable program
 - Links object code placed in more than one source code file and program libraries

Integrated Development

- An integrated development environment (IDE), like Visual C++, groups the editor, compiler, linker, and debugging tools into a single program
- Compilers can check for syntax errors (like a spelling or grammar checker in a word processor)
- Other errors may still exist and have to be found before the program is correct

Development Cycle

- Program requirements specification
- Program design
- Design of individual units
- Coding of individual units
- Testing of individual units
- Unit integration and testing
- Final product test against specifications and requirements

Program Testing

- A key step in programming
- Debugging a program done by checking code, intermediate output and debugger
- Submit correct programs
 - Incorrect programs are returned to student with a very low grade
 - It is better to submit a correct program late than to submit an incorrect one on time