

\$Id: syllabus-cmps109.mm,v 1.2 2016-06-10 17:01:15-07 - - \$

PWD: /afs/cats.ucsc.edu/courses/cmps109-wm/Syllabus

URL: http://www2.ucsc.edu/courses/cmps109-wm/:/Syllabus/

1. General Information

The generic part of the syllabus contains detailed information about prohibiting cheating, due dates and times, submitting assignments, and verification of the submit. Read it carefully, as you will be held responsible for it.

Directory: The directory `/afs/cats.ucsc.edu/courses/cmps109-wm/` and its subdirectories contain all assignments, handouts, examples, old exams, etc.

Piazza: <https://piazza.com/> is for questions and discussions that are appropriate in the classroom or lab section.

Assignments: All assignments must be submitted electronically and must work on the Unix servers (`unix.ucsc.edu`).

Due Dates: Due dates are announced in the `README` files in the course directory. You must frequently check the `README`.

Cheating: ***Cheating will not be tolerated. See the section on cheating in the generic part of the syllabus.***

Grades: To pass the course, both the programming component and the testing component will be taken into consideration. Failing either component may be cause to fail the course. Your final grade will be computed as follows:

Programming assignments: $5 \times 10\% = 50\%$

Midterm and final tests in class: $2 \times 25\% = 50\%$

Summer quarter does not have an exam week.

2. Course Description from Catalog

CMPS-109. Advanced Programming. An introduction to object-oriented techniques of software development including data abstraction, inheritance, polymorphism, and object-oriented design. Extensive practice using a computer to solve problems, including construction of graphical user interfaces and a multithreaded client/server application. **Prerequisites:** CMPS-012B/M or CMPS-013H.

3. Textbooks and References

Only the first textbook (by Stroustrup) is listed as “required”. The others are optional references.

- (1) Bjarne Stroustrup: *Programming Principles and Practice Using C++*, second edition. Addison-Wesley, 2014. ISBN 0-321-99278-4. This is an elementary textbook for a first course in C++. The previous edition will also work, or any other C++ textbook you may already have. It is strongly recommended that you read a book advertising the use of C++11. Any C++ book not discussing C++11 should be considered obsolete.

- (2) The C++ Resources Network: Use this site to find specific information about various classes in the standard library.
<http://www.cplusplus.com/>
- (3) C++11: The New ISO C++ Standard FAQ:
<http://www.stroustrup.com/C++11FAQ.html>
- (4) Bjarne Stroustrup: *The C++ Programming Language, 4th edition*. Addison-Wesley, 2013. This is the revised definitive description of C++11.
- (5) Stanley B. Lippman, Josée Lajoie, Barbara E. Moo: *C++ Primer, 5th edition*. Addison-Wesley, 2013. A good primer discussing C++11.
- (6) Nicolai M. Josuttis: *The C++ Standard Library, 2nd edition: A Tutorial and Reference*. Addison-Wesley, 2012. A specific tutorial on the library, with C++11.
- (7) JTC1/SC22/WG21 — The C++ Standards Committee: Latest publicly available draft: *N3797 Working Draft, Standard for Programming Language C++*. 2013-10-13.
<http://www.open-std.org/jtc1/sc22/wg21/>
<http://www.open-std.org/jtc1/sc22/wg21//docs/papers/2013/n3797.pdf>
- (8) Bjarne Stroustrup: *The Design and Evolution of C++*. Addison-Wesley, 1994. This is a historical document where Stroustrup discusses his design philosophy and how it derived from C with Classes.
- (9) P.J. Plauger, Alexander Stepanov, Meng Lee, David Musser: *The C++ Standard Template Library*. Prentice-Hall, 2001. Detailed description of the implementation of the STL, showing detailed code examples.
- (10) Scott Meyers, <http://www.aristeia.com/books.html>

4. Detailed Syllabus.

This course is about programming in C++, including C++11. Prior programming knowledge of ANSI C is assumed.

- (1) C vs C++. Some differences: Input/output, strings, vectors instead of standard I/O, character and other arrays. Fundamental data types. Using the Standard Template Library (STL).
- (2) Functions: pass by value, reference, and const reference. Namespaces.
- (3) Classes: interface and implementaton. Header files and file guards. Member functions. Operator and function overloading.
- (4) Input and output streams. User-defined I/O operators. Formatting I/O.
- (5) Vectors and free store. Memory management Constructors and destructors. Copying and assignment of objects. Pointers and references. Shared_ptr and Unique_ptr.
- (6) Vectors and arrays. Copy and move constructors, and copy and move assignments. Destructors. Explicit constructors.

- (7) Inheritance polymorphism and object-oriented programming. Abstract classes. Virtual functions. Overriding functions and operators. Multiple inheritance as interfaces.
- (8) Template polymorphism and generic programming. Containers and inheritance. Range checking and resource management. Exceptions: defining, throwing, and catching.
- (9) Containers and iterators in the standard library. Vectors, lists, and strings. Algorithms and maps.
- (10) Graphical user interfaces (GUI). Threads. Sockets. Client/server applications.
- (11) Miscellaneous other topics: TBA.

5. Students with Disabilities

If you qualify for classroom accommodations because of a disability, please submit your Accommodation Authorization Letter from the Disability Resource Center (DRC) to me as soon as possible, preferably within the first week of the quarter. Contact DRC by phone at 831-459-2089 or by email at drc@ucsc.edu for more information.

6. Pair Programming

You may do pair programming if you choose. You are responsible for choosing a partner with whom you can work. Read the guidelines in the directory [pair-programming/](#).