USC Viterbi
School of Engineering

Professional C++

ITP 435x (3 Units)
Fall 2014

Objective This course provides students with the advanced knowledge they will need to succeed as a professional C++ developer. By semester's end, students will:

1. Become familiar with advanced C++ language idioms.
2. Gain exposure to common libraries used professionally today.
3. Understand how to write efficient and high-quality C++ code.

Concepts Code Generation. Memory layout. Templates. STL. Optimization. Exceptions. RTTI. Design Patterns. Metaprogramming. Lambda Expressions. Boost. Custom Memory Allocators. C++11. Compilers.

Prerequisites CSCI 104 or ITP 365x

Instructor Sanjay Madhav

Contact Any questions related to the course and material should be posted on Piazza.
Email: madhav@usc.edu (Only for non-course questions or prospective students).

Office Hours Monday – Thursday 2:30PM – 4:30PM in OHE 530H

Lab Assistants Tianyu Lang and Matthew Pohlmann (contact via Piazza)

Lecture Monday 12 – 1:50PM in OHE 540

Lab Wednesday 12 – 1:50PM in OHE 540

Course Structure The topics covered during lecture will be applied to the five lab assignments spread out through the semester. All lab assignments must be done *individually*.

There are two exams that are comprehensive of all topics covered.

There will periodically be "pop" quizzes. These quizzes will generally be 15-20 minutes in length, and can occur during any class session. There are a total of six quizzes, with the lowest quiz grade being dropped.

Textbooks **Required**:
Effective C++ (Third Edition). Scott Meyers. ISBN-10: 0321334876.
Optional:
Modern C++ Design: Generic Programming and Design Patterns Applied. Andrei Alexandrescu. ISBN-10: 0201704315.
Reference:
The C++ Programming Language (4th Edition). Bjarne Stroustrup. ISBN-10: 0321563840.

Grading The course is graded with the following weights:

Lab Assignments (10% each)	50%
Quizzes (6 total, lowest grade dropped)	10%
Midterm Exam	20%
Final Exam	20%
TOTAL POSSIBLE	100%

Grading Scale

Letter grades will be assigned according to the following scale:

93%+	A
90-92%	A-
87-89%	B+
83-86%	B
80-82%	B-
77-79%	C+
73-76%	C
70-72%	C-
69	D+
67-68	D
66	D-
65 and below	F

Half percentage points will be rounded up to the next whole percentage. So for instance, 89.5% is an A-, but 89.4% is a B+.

There is no curving. Students will receive the grade they earn. Extra credit is generally not offered.

Policies

Make-up policy for exams: To make up for a missed exam, the student must provide a satisfactory reason (as determined by the instructor) along with proper documentation. Make-up exams are only allowed under extraordinary circumstances.

Late Assignments: Late assignments will be accepted one day late for a 15% penalty and two days late for a 30% penalty. An assignment later than this will be given a grade of 0, unless there is an extraordinary and documented reason as to why.

Students will be able to setup their own PC or Mac for use in the class, as the software is free for Viterbi students. Note that the software we will use works on PC only, so students with a Mac will have to either install Boot Camp or a VM.

Alternatively, there are machines in the classroom that can be used. Before logging off a computer, students must ensure that they have emailed or saved projects created during the class or lab session. Any work saved to the computer will be erased after restarting the computer. ITP is not responsible for any work lost.

ITP offers Open Lab use for all students enrolled in ITP classes. These open labs are held beginning the second week of classes through the last week of classes. Please contact your instructor for specific times and days for the current semester.

Academic Integrity	USC seeks to maintain an optimal learning environment. General principles of academic honesty include the concept of respect for the intellectual property of others, the expectation that individual work will be submitted unless otherwise allowed by an instructor, and the obligations both to protect one's own academic work from misuse by others as well as to avoid using another's work as one's own. All students are expected to understand and abide by these principles. SCampus, the Student Guidebook, (www.usc.edu/scampus or http://scampus.usc.edu) contains the University Student Conduct Code (see University Governance, Section 11.00), while the recommended sanctions are located in Appendix A.
Plagiarism	In this class, all homework submissions will be compared with current, previous, and future students' submissions using MOSS, which is a code plagiarism identification program. If your code significantly matches another student's submission, you will be reported to SJACS with the recommended penalty of an F in the course. It is okay to discuss solutions to specific problems with other students, but it is not okay to look through another student's code. It does not matter if this code is online or from a student you know, it is cheating. Do not share your code with anyone else in this or a future section of the course, as allowing someone else to copy your code carries the same penalty as you copying the code yourself.
Students with Disabilities	Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m.–5:00 p.m., Monday through Friday. Website and contact information for DSP: http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html, (213) 740-0776 (Phone), (213) 740-6948 (TDD only), (213) 740-8216 (FAX) ability@usc.edu.
Emergency Preparedness	In case of a declared emergency if travel to campus is not feasible, USC executive leadership will announce an electronic way for instructors to teach students in their residence halls or homes using a combination of Blackboard, teleconferencing, and other technologies. Please activate your course in Blackboard with access to the course syllabus. Whether or not you use Blackboard regularly, these preparations will be crucial in an emergency. USC's Blackboard learning management system and support information is available at blackboard.usc.edu.

Course Outline

Week 1 – Introduction and Assorted Topics (8/25 and 8/27)

- Course intro
- Pointers; const and mutable; C++ style casts
- Default members and explicit

Reading: *Meyers*: Introduction; Items 1-4, 7, 9-12, 20, 27, 32-40
Lab: Begin work on Lab 1: RLE Compressor.

Week 2 – Sizeof, Inheritance, and Code Generation (9/3)

- Class data layout detailed
- private inheritance, multiple inheritance, virtual destructors
- Preprocessor, macros and constexpr

Reading: *Meyers*: Items 5, 6, 26, 30, 31, 41-45, 53
Lab: None due to Labor Day (9/1)

Week 3 – Templates, Basic STL, and Lambda Expressions (9/8 and 9/10)

- Template generation
- STL container types (including C++11 additions)
- Function Objects and Lambda Expressions

Reading: None
<u>Lab 1 DUE Tuesday, 9/16 @ 11:59PM</u>

Week 4 – STL Algorithms, Exceptions, and RTTI (9/15 and 9/17)

- <algorithm>: merge, sort, etc.
- Using exceptions properly and their drawbacks
- Run-Time Type Identification (RTTI) and implementing your own RTTI

Reading: None
Lab: Begin work on Lab 2: Password Cracker.

Week 5 – Design Patterns and Smart Pointers (9/22 and 9/24)

- Singletons
- Factory, Proxy, Memento
- Smart pointers

Reading: None
Lab: Continue Lab 2.

Week 6 – Advanced Templates (9/29 and 10/1)

- Metaprogramming
- Applications: Static assert, compile-time calculations, conditional checks
- Variadic Templates

Reading: *Meyers*: Items 46-49; *Alexandrescu*: Chapters 2 and 3
<u>Lab 2 DUE Tuesday, 10/7 @ 11:59PM</u>

Week 7 – Writing Optimized and Secure Code (10/6 and 10/8)

- Optimization techniques
- Writing efficient code naturally
- Security pitfalls in C++

Reading: *Meyers*: Items 30 and 31.
Lab: Begin work on Lab 3: Paint Program in WTL.

Week 8 – **<u>Midterm Exam on Monday, 10/13</u>**

Lab: Continue Lab 3 (10/15).

Week 9 – Custom Memory Allocators (10/20 and 10/22)

- Overloading new and delete
- Memory allocator approaches (pool, stack, boundary allocators)
- Placement new

Reading: *Meyers*: Items 49-52

Lab 3 DUE Tuesday, 10/28 @ 11:59PM

Week 10 – Introduction to Compilers, Part 1 (10/27 and 10/29)

- Stages of compilation
- Regular expressions
- Lexical analysis and flex

Reading: *Meyers*: Items 49-52

Lab: Begin work on Lab 4: Zombie Simulation Machine.

Week 11 – Introduction to Compilers, Part 2 (11/3 and 11/5)

- BNF grammars
- Bison
- Abstract syntax trees

Lab: Continue Lab 4.

Week 12 – Introduction to Compilers, Part 3 (11/10 and 11/12)

- Code generation
- Register allocation
- x86 compilation examples

Lab 4 DUE Tuesday, 11/18 @ 11:59PM

Week 13 – Boost Library and Multithreading (11/17 and 11/19)

- lexical_cast**,** program_options, signal, bimap, etc
- C++11 threading library
- Threading applications

Reading: *Meyers*: Item 55, boost library documentation, std::thread ISO whitepaper

Lab: Begin work on Lab 5: ZombieC Compiler.

Week 14 – Assorted Topics (11/24)

- Overflow/additional topics

Reading: On Blackboard

Lab: No lab due to Thanksgiving

Week 15 – Testing and Code Verification (12/1 and 12/3)

- Unit testing
- Regression testing
- Fuzz testing

Lab 5 DUE Friday, 12/5 @ 11:59PM

Final Exam Friday, 12/12 @ 11AM