

<b>Objective</b>	This course provides students with the advanced knowledge they will need to succeed as a professional C++ developer. By semester's end, students will: <ol style="list-style-type: none"> <li>1. Become familiar with advanced C++ language idioms.</li> <li>2. Gain exposure to common libraries used professionally today.</li> <li>3. Understand how to write efficient and high-quality C++ code.</li> </ol>
<b>Concepts</b>	Code Generation. Memory layout. Templates. STL. Optimization. Exceptions. RTTI. Design Patterns. Metaprogramming. Lambda Expressions. Boost. Custom Memory Allocators. C++11. Compilers.
<b>Prerequisites</b>	CSCI 104 or ITP 365x
<b>Instructor</b>	Sanjay Madhav
<b>Contact</b>	Students in the course should post their questions on Piazza. <i>Email:</i> madhav@usc.edu (Only for non-course questions or prospective students).
<b>Office Hours</b>	Tuesday/Thursday 4:30-6:30PM in OHE 530H
<b>Lecture</b>	Tuesday and Thursday, 2:00 – 3:20PM in KAP 160
<b>Course Structure</b>	The topics covered during class meetings will be applied to the seven programming assignments spread out through the semester. All programming assignments must be completed <i>individually</i> .

Here's an example of what the assignments were in spring 2017. They may vary in spring 2018.

1. RLE Compression/Decompression
2. Password Cracker
3. Paint/Smart Art Program
4. DNA Amino Acid Histogram + Sequence Alignment
5. Zombie Apocalypse Simulator
6. Zombie-C Compiler
7. Travelling Trojan

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

<b>Textbooks</b>	<b>Required:</b> <i>Effective C++ (Third Edition)</i> . Scott Meyers. ISBN-10: 0321334876. <b>Recommended:</b> <i>Effective Modern C++</i> . Scott Meyers. ISBN-10: 1491903996.										
<b>Grading</b>	The course is graded with the following weights: <table border="0"> <tr> <td>Programming Assignments (7% each)</td> <td>49%</td> </tr> <tr> <td>Midterm Exam</td> <td>21%</td> </tr> <tr> <td>Final Exam</td> <td>25%</td> </tr> <tr> <td>Class Participation</td> <td>5%</td> </tr> <tr> <td><b>TOTAL POSSIBLE</b></td> <td><b>100%</b></td> </tr> </table>	Programming Assignments (7% each)	49%	Midterm Exam	21%	Final Exam	25%	Class Participation	5%	<b>TOTAL POSSIBLE</b>	<b>100%</b>
Programming Assignments (7% each)	49%										
Midterm Exam	21%										
Final Exam	25%										
Class Participation	5%										
<b>TOTAL POSSIBLE</b>	<b>100%</b>										

---

**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+.

Depending on the overall class average at the end of the semester, the above grading scale may be relaxed. 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, two days late for a 30% penalty, and three days late for a 45% penalty. An assignment submitted later than this will be given a grade of 0, unless there is an extraordinary and documented reason as to why it was late.

Students will be able to setup their own PC or Mac for use in the class, as all software is free either in general or specifically for students enrolled in Viterbi courses. All projects natively build on both PC and Mac, assuming the appropriate software is installed. Linux should work as well, but no technical support will be provided for students who wish to use Linux.

Alternatively, 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.

---

<p><b>Statement on Academic Conduct and Support Systems</b></p>	<p><b>Academic Conduct</b>  Plagiarism – presenting someone else’s ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in <i>SCampus</i> in Section 11, <i>Behavior Violating University Standards</i> <a href="https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions/">https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions/</a>. Other forms of academic dishonesty are equally unacceptable. See additional information in <i>SCampus</i> and university policies on scientific misconduct, <a href="http://policy.usc.edu/scientific-misconduct/">http://policy.usc.edu/scientific-misconduct/</a>.</p> <p>Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the <i>Office of Equity and Diversity</i> <a href="http://equity.usc.edu/">http://equity.usc.edu/</a> or to the <i>Department of Public Safety</i> <a href="http://capsnet.usc.edu/department/department-public-safety/online-forms/contact-us">http://capsnet.usc.edu/department/department-public-safety/online-forms/contact-us</a>. This is important for the safety whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. <i>The Center for Women and Men</i> <a href="http://www.usc.edu/student-affairs/cwm/">http://www.usc.edu/student-affairs/cwm/</a> provides 24/7 confidential support, and the sexual assault resource center webpage <a href="http://sarc.usc.edu">sarc.usc.edu</a> describes reporting options and other resources.</p>
<p><b>A Further Note on Plagiarism</b></p>	<p><b>Support Systems</b>  A number of USC’s schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the <i>American Language Institute</i> <a href="http://dornsife.usc.edu/ali">http://dornsife.usc.edu/ali</a>, which sponsors courses and workshops specifically for international graduate students. <i>The Office of Disability Services and Programs</i> <a href="http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html">http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.html</a> provides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, <i>USC Emergency Information</i> <a href="http://emergency.usc.edu/">http://emergency.usc.edu/</a> will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.</p> <p>In this class, all programming assignment 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.</p> <p>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.</p>

Course Outline			
W	Date	Topic(s)	Reading/PA
1	1/9	Introduction; Tools and Testing	
	1/11	Modernizing C++	<i>Effective</i> : Intro & #1-4; 7; 9-12; 20, 27
2	1/16	<b>No class (synchronization with M/W)</b>	
	1/18	Lambdas/Functional Programming	<i>Modern</i> : #2, 5, 6
3	1/23	Genetic Algorithms	
	1/25	Sizeof; Virtual tables	<i>Effective</i> : #5, 6, 26, 30; <b>PA1 Due 1/24 @ 11:59PM</b>
4	1/30	Is-a vs. Has-a; Preprocessor	<i>Effective</i> : #32-40;
	2/1	Bioinformatics and Dynamic Programming	
5	2/6	Writing Optimized and Secure Code	<i>Effective</i> : #30-31;
	2/8	Design Patterns	<b>PA2 Due 2/7 @ 11:59PM</b>
6	2/13	Smart Pointers	<i>Modern</i> : #18-21
	2/15	Basic Parallel Programming; Intel TBB	
7	2/20	<b>No class (synchronization with M/W)</b>	
	2/22	Guest Lecture	<b>PA3 Due 2/21 @ 11:59PM</b>
8	2/27	Midterm Review	
	3/1	<b>Midterm exam</b>	
9	3/6	Move Semantics	<i>Modern</i> : #23-26;
	3/8	Exceptions and RTTI	<b>PA4 Due 3/7 @ 11:59PM</b>
	3/13	<b>Spring Break</b>	
	3/15		
10	3/20	Custom Memory Allocators	<i>Effective</i> : #49-52
	3/22	Template Metaprogramming	
11	3/27	Uniform Initializers; Initializer Lists	<i>Modern</i> : #7
	3/29	Secure Design, Development, and Test	<b>PA5 Due 3/28 @ 11:59PM</b>
12	4/3	Intro. to Compilers – Basics; Lexical Analysis	
	4/5	Intro. to Compilers – Syntax Analysis	
13	4/10	Intro. to Compilers – Code Generation	
	4/12	Boost Library; C++11 Concurrency; Testing	<i>Modern</i> : #25; <i>Effective</i> : #55; <b>PA6 Due 4/13 @ 11:59PM</b>
14	4/17	C++14, 17, and 20	
	4/19	Deep C	
15	4/24	Selected Talks from Cppcon2017	
	4/26	Final Review	<b>PA7 Due 4/27 @ 11:59PM</b>
		<b>Final Exam – Thursday, May 3 @ 2-4PM</b>	

Note that “Effective” refers to *Effective C++* while “Modern” refers to *Effective Modern C++*.