

**San José State University**  
**Department of Computer Science**  
**CS-144, Advanced C++ Programming, Section 1, Spring 2018**

**Course and Contact Information**

<b>Instructor:</b>	Michael Loceff
<b>Office Location:</b>	Online in the Canvas Portal
<b>Telephone:</b>	(We'll use SJSU Zoom Meetings for Voice/Video)
<b>Email:</b>	Michael.loceff@sjsu.edu
<b>Office Hours:</b>	Daily in the course management system ("Canvas"), with typical response time < 4 hours, or by arrangement.
<b>Class Days/Time:</b>	Flexible 3-4 times/week at student convenience
<b>Classroom:</b>	Fully On-line using SJSU's Canvas portal
<b>Prerequisites:</b>	CS 46B and CS 49C (with a grade of C- or better in each), or equivalent knowledge of object-oriented programming and (C or Java or Python), or instructor consent.
<b>Course Format:</b>	Fully online using SJSU Canvas login. (No in-person meetings.) There will be assigned readings, programming labs and/or quizzes due at specific days each week, but you can read the course content at times the fit your schedule. All work, including programming assignments and exams, will be submitted and graded online in Canvas. Regular instructor interaction will be available in class forums and/or optional live Zoom video chat.

**Additional Online Course Requirements**

This course adopts a fully online delivery format using Canvas. You will need Internet connectivity and a laptop or desktop computer on which to install a C++ compiler and visit the lecture, assignment, exam and submission pages. A smart-phone or tablet is not adequate for doing programming and cannot be substituted for a personal computer. Full instructions for installing compilers and submitting all work will be found in your first week reading at the Canvas course site.

**Announcements and Messaging**

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on the [Canvas Learning Management System course login website](http://sjsu.instructure.com) at <http://sjsu.instructure.com>. You are responsible for regularly checking with the messaging system through [MySJSU](http://my.sjsu.edu) at <http://my.sjsu.edu> and the Canvas course, itself, to learn of any updates. Detailed instructions will be supplied in course

site starting the first day of class. Please try to use the Canvas site, not email, to contact me regarding questions once the course has begun

## Course Description

Advanced features of C++, including operator overloading, memory management, templates, exceptions, multiple inheritance, RTTI, namespaces, tools. Prerequisite: CS 46B and CS 49C (with a grade of C- or better in each), or equivalent knowledge of object-oriented programming and C, or instructor consent.

## Course Learning Outcomes (CLO)

Upon successful completion of this course, students will be able to:

1. Understand and apply advanced C++ constructs and concepts such as the STL (Standard Template Library), operator overloading, memory management, templates, exception handling, multiple inheritance, RTTI (Run-Time Type Information), namespaces, and various C++ related tools/libraries/modules.
2. Apply the advanced C++ constructs covered in this course to the design and implementation of moderately sophisticated C++ applications.

## Texts/Readings

### Textbook

There are no required books for this class. All the necessary material will be available on the course Canvas web page.

However, you should have a recommended reference available. Here are a few suggestions, but you can choose anything comparable:

- **Absolute C++ (any edition)** by Walter Savitch
- **Big C++ (any edition)** by Cay Horstman and Timothy Budd
- **The C++ Programming Language** (any edition) by Bjarne Stroustrup
- **Thinking in C++** (any edition) by Bruce Eckel

## Other Technology Requirements / Equipment /Material

You will need access to a desktop or laptop computer. You can use any C++ compiler or integrated development environment (IDE) that you wish, but I will be offering specific download and installation instructions only for **Microsoft Visual Studio/C++** for Windows users and **Xcode** for Mac users, **both free**. There are ample online resources if you choose other Windows or Mac compilers (or a Linux/Unix IDE).

## Course Requirements and Assignments

According to SJSU policy, it is expected that students spend a minimum of forty-five hours for each unit of credit (normally three hours per unit per week), including preparing for class, participating in course

activities, completing assignments, and so on. More details about student workload can be found in [University Policy S12-3](http://www.sjsu.edu/senate/docs/S12-3.pdf) at <http://www.sjsu.edu/senate/docs/S12-3.pdf>.

Programs, Quizzes, Midterms, Forum Participation, and a Final exam are all components of this class. Programs are due most weeks on Wednesdays at 2 PM in the Canvas portal. Tests and forums are also found in the same portal. Detailed instructions for accessing these areas -- as well as a calendar of dates -- will be available in the Canvas site starting the first day of class.

NOTE that [University policy F69-24](http://www.sjsu.edu/senate/docs/F69-24.pdf) at <http://www.sjsu.edu/senate/docs/F69-24.pdf> states that "Students should attend all meetings of their classes, not only because they are responsible for material discussed therein, but because active participation is frequently essential to insure maximum benefit for all members of the class. Attendance per se shall not be used as a criterion for grading."

### Assignments, Tests, Participation and Grading

- 13 Programming Assignments @20 points each = 260 points
- 2 Midterms @40 points each = 80 points
- 2 Quizzes @20 points each = 40 points
- Final Exam = 80 points
- Online Public Forum Participation (questions, comments, or answers at least once a week) = 60
- TOTAL POINTS POSSIBLE = 520

Note that "All students have the right, within a reasonable time, to know their academic scores, to review their grade-dependent work, and to be provided with explanations for the determination of their course grades." See [University Policy F13-1](http://www.sjsu.edu/senate/docs/F13-1.pdf) at <http://www.sjsu.edu/senate/docs/F13-1.pdf> for more details.

### Grading Scale

Semester grade will be computed as a percentage of the total scores as follows:

Percentage (minimum for each grade)	Grade
90	A
88	A-
85	B+
80	B
78	B-
75	C+
70	C

68	C-
65	D+
60	D
58	D-
Below 58	F

## Classroom Protocol

Programming assignments and exams are meant to be done individually. However, you can ask me for help in the Canvas public forums (recommended), CS Study Lab (MH226) or private messaging. Also, you are free to ask for general assistance in the public forums where other students or I may give you pointers or clarification without posting exact homework code. Any code found to be largely copied from the web or from other students' work will be reported to the dean of students (and will receive an automatic 0 with disqualification for future bonus points). Both the recipient and the unauthorized source (if it was a student who gave you the answer) will be reported.

## University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' [Syllabus Information web page](http://www.sjsu.edu/gup/syllabusinfo/) at <http://www.sjsu.edu/gup/syllabusinfo/>

## CS-144 / Advanced C++ Programming, Spring 2018, Course Schedule

The material will be organized into (usually) *two online modules per week*, but you can study each of them at any time within its assigned the week that fits your schedule. I and other students will be available in forums to answer questions daily (I'll be available 7 days/week).

**Course Schedule (Subject to minor changes up until the first day of class at which time the final schedule will be posted in the Canvas portal).**

**NOTE:** There will be **13 Programming Assignments** due almost every **Wednesday**, beginning Wed Feb 7 at 2 PM. These are not listed in the schedule below, but their due dates can be found at the course site. They may take anywhere from 3 to 6 hours each week for you to complete. There will be **2 quizzes**, **2 midterm exams** and **1 final exam**, and their dates appear, below.

Week	Date	Topics
1R, 1QS	1/24	"Quick Start" to C++, compiler/IDE download instructions.

Week	Date	Topics
1A	1/29	Data types, type compatibility, loop preview
1B	2/1	User input, strings and numbers, type conversions, output formatting
2A	2/5	Selection (if, else/if, switch), relational expressions, logical operators
2B	2/8	Repetition (for and while loop comparison and variations), more string mechanics
3A	2/12	Functions, methods, namespaces, locals, globals, formal parameters, default arguments, function/method overloading
3B	2/15	OOP, introduction to classes in C++
4A	2/19	Instance/static methods and data, data privacy, constructors, mutators, accessors
4B	2/22	Early pointers, dynamic allocation, the “this” pointer, functional return of objects, summary of OOP design strategy for C++
Quiz 1	2/23	Online. About 20 minutes in one sitting, available to be taken any time this day.
5A	2/26	Reference parameters, fixed-size 1-D arrays, memory management, examples using sorting
5B	3/1	Anonymous objects, arrays inside classes, multi-class interaction
6A	3/5	Multi-file projects, recursion in C++, binary search algorithms
6B	3/8	Bit operations, binary and hex in C++, address vs. value passing, applications to cellular automata, “life” and M.Mitchell rules
Midterm Exam 1	3/9	Online. 1 hour in one sitting, available to be taken any time this day.
7A	3/12	Enums, C strings vs. C++ string class
7B	3/15	Standard template library (STL) containers: vectors, lists, stacks, iterators, the subset sum problem
8A	3/19	Inheritance, method and constructor chaining, building stack ADTs from scratch with subclasses.
8B	3/22	Inheritance applied to linked-lists, security considerations in accessors, non-virtual method techniques.
	3/26	(Spring Break)
	3/29	(Spring Break)
9A	4/2	Dynamic allocation and destruction of array data, dynamic vs. fixed-size arrays, multi-dimensional arrays, stack vs. heap, deep copies
9B	4/4	Destructors, copy constructors, application of inheritance to circuit complexity, mathematical Boolean functions and multi-segment-display logic

<b>Week</b>	<b>Date</b>	<b>Topics</b>
10A	4/9	Operator overloading, friends, insertion and assignment operators in inheritance
10B	4/12	Guarded code, built-in and user-defined exceptions,
Quiz 2	4/13	Online. About 20 minutes in one sitting, available to be taken any time this day.
11A	4/16	Polymorphism, virtual functions, RTTI, dynamic casts, typeid(), abstract classes, pure virtual functions
11B	4/19	Multiple Inheritance, method resolution order
Midterm Exam 2	4/20	Online. 1 hour in one sitting, available to be taken any time this day.
12A	4/23	Specialization and designing custom class and function templates
12B	4/25	Template instantiation/specialization details and static/instance issues, advanced use of "const," built-in STL algorithms, sparse matrices in C++
13A	4/30	Building trees in C++, OOP applied to lazy tree deletion, functors, tree traversals, more about deep copies and copy constructors
13B	5/3	File I/O and streams
14A	5/7	Advanced STL Classes and Examples: sets, priority_queues, maps and Flexible Ordering Techniques.
14B	5/10	Bonus assignment details and catch-up
15A	5/14	Final exam details and catch-up
Final Exam	5/18 (From student's home or college computer.)	Online. 2 hours in one sitting, available to be taken any time this day. This will be physically taken in any quiet and secure site of the student's choosing. (Detailed guidelines about taking on-line final exams posted at the course site.)