

San José State University  
Department of Computer Engineering

CMPE 180A  
Data Structures and Algorithms in C++

Section 1  
Fall 2020

**Course and Contact Information**

Instructor: Ron Mak  
Office location: ENG 250 (but working from home)  
Email: [ron.mak@sjsu.edu](mailto:ron.mak@sjsu.edu)  
Website: <http://www.cs.sjsu.edu/~mak/>  
Office hours: TuTh 4:30 – 5:30 PM online via Zoom  
Class days/time: Th 6:00 – 8:45 PM online via Zoom  
Classroom: online  
Prerequisites: Admission into the Computer Engineering or the Software Engineering master's degree program.

**Course catalog description**

Individual work in computer engineering. Prerequisite: Upper division standing and instructor consent. Not available to Open University Students.

Object-oriented data organization and representation as strings, arrays, stacks, queues, dequeues, lists, sets, trees, tables, and graphs. Sorting and searching and algorithm design and performance analysis. Testing methods and data will be discussed.

**Course Format**

This course adopts a synchronous online classroom delivery format. To participate in classroom activities, submit assignments, and take tests/exams remotely, a student must have a computer with adequate internet connection and bandwidth for accessing Canvas and attending Zoom video meetings. A smartphone or tablet with a camera capable of running Zoom is also needed for video recording of your test environment during the tests/exams.

**Faculty Web Page and Canvas**

Course materials, syllabus, assignments, grading criteria, exams, and other information will be posted at my [faculty website](http://www.cs.sjsu.edu/~mak) at <http://www.cs.sjsu.edu/~mak> and on the [Canvas Learning Management System course login website](http://sjsu.instructure.com) at <http://sjsu.instructure.com>. You are responsible for regularly checking these websites to learn of any updates. You can find Canvas video tutorials and documentations at <http://ges.sjsu.edu/canvas-students>.

## Instructor's description

CMPE 180A will also examine some of the features of modern C++, such as lambda expressions, smart pointers, and move semantics. It will show how to write efficient C++ programs that adhere to good design principles while avoiding pitfalls of the language.

## Course learning outcomes (CLO)

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

- CLO 1: Apply **object-oriented features** of C++, including polymorphism and recursion.
- CLO 2: Apply **advanced features** of C++, including operator overloading, memory management, templates, the Standard Template Library (STL), exceptions, multiple inheritance, runtime type identification (RTTI), namespaces, etc.
- CLO 3: Apply **modern features** of C++, including lambda expressions, smart pointers, move semantics, etc.
- CLO 4: Use **standard abstract data types** and data structures, including stacks, queues, and linked lists, trees, in the design of software programs.
- CLO 5: Apply **standard algorithmic techniques** including recursions, hashing, searching, and sorting in the design of software programs.
- CLO 6: Use high-level **software development tools**, including advanced text editors, compilers, linkers, source-level debuggers to implement and debug software programs.

## Academic Integrity

You may study together and discuss the assignments, but what you turn in must be your individual work. Assignment submissions will be checked for plagiarism using Moss from the Computer Science Department at Stanford University (<http://theory.stanford.edu/~aiken/moss/>). See <http://www.cs.sjsu.edu/~mak/Moss/> for a report from a Moss run.

**Copying code from another student's program or sharing your program code are equal violations of academic integrity.** Moss is not fooled by renaming variables, reformatting code, or re-ordering functions.

**Violators of academic integrity will suffer severe sanctions, including academic probation.** Students who are on academic probation are not eligible for work as instructional assistants in the university or for internships at local companies.

“Major exams in this class may be video recorded to ensure academic integrity. The recordings will only be viewed if there is an issue to be addressed. Under no circumstances will the recordings be publicly released.”

### Required Texts

Title:	<b>Problem Solving with C++, 10th edition</b>
Author:	Walter Savitch
Publisher:	Pearson, 2017
ISBN:	978-0134710747
Title:	<b>Data Structures Using C++, 2nd edition</b>
Author:	D.S. Malik
Publisher:	Cengage Learning, 2010
ISBN:	978-0324782011

### Recommended Texts

Title:	<b>Big C++, 3rd edition</b>
Author:	Cay S. Horstmann, Timothy A. Budd
Publisher:	Wiley, 2013
ISBN:	978-1118674291
Title:	<b>C++ How to Program, 10th edition</b>
Author:	Paul Deitel, Harvey Deitel
Publisher:	Pearson, 2016
ISBN:	978-0134448237
Title:	<b>The C++ Programming Language, 4th edition</b>
Author:	Bjarne Stroustrup
Publisher:	Addison-Wesley Professional, 2013
ISBN:	978-0321563842
Title:	<b>Data Structures and Algorithms in C++, 4th edition</b>
Author:	Adam Drozdek
Publisher:	Cengage Learning, 2012
ISBN:	978-1133608424
Title:	<b>Data Structures and Algorithm Analysis in C++, 4th edition</b>
Author:	Mark A. Weiss
Publisher:	Pearson, 2013
ISBN:	978-0132847377

## Software to Install

This class will use the GNU C++ compiler. You should install and use an interactive development environment (IDE) such as Eclipse CDT.

GNU C++ is usually pre-installed on the Mac and Linux platforms. If you are on a Mac, you should not use Apple's Xcode development environment for this class, because it may cause you to write programs that will not port to other platforms. The Windows platform often has significant compatibility challenges. Therefore, if you're on Windows, you should [install the Ubuntu distribution](https://tutorials.ubuntu.com/tutorial/tutorial-ubuntu-on-windows#0): <https://tutorials.ubuntu.com/tutorial/tutorial-ubuntu-on-windows#0>.

Some useful tutorials:

- “Install Ubuntu on Windows 10 and on VirtualBox”  
<http://www.cs.sjsu.edu/~mak/tutorials/InstallUbuntu.pdf>
- “Configure Ubuntu for Software Development”  
<http://www.cs.sjsu.edu/~mak/tutorials/ConfigureUbuntu.pdf>
- “Install Eclipse for Java and C++ Development”  
<http://www.cs.sjsu.edu/~mak/tutorials/InstallEclipse.pdf>
- “Install MPIR on Ubuntu and MacOS X”  
<http://www.cs.sjsu.edu/~mak/tutorials/InstallMPIR.pdf>

## The C++ 2011 standard

Use the 2011 standard of C++. You must set this explicitly for your project in Eclipse:

- **Eclipse:** Right-click on your project in the project list at the left side of the window. Select “Properties” from the drop-down context menu. In the left side of the properties window, select “C/C++ Build” → “Settings”. In the Settings dialog, select “GCC C++ Compiler” → “Dialect”. For “Language standard” select “ISO C++ 11”. Click the “Apply” button, answer “Yes”, and then click the “OK” button.

To compile on the command line, include the `-std=c++11` option:

```
g++ -std=c++11 -o foo foo.cpp
```

Make sure you use hyphens and not dashes.

## Course requirements and Assignments

*This class will progress rapidly, and you must work hard to keep up. Do not fall behind in the reading. There will be multiple programming assignments each week.*

Each assignment will be worth a specified maximum number of points, depending on difficulty, and it will be due before the start of the next class. No assignment will be accepted after its solution is presented in class (it will get a 0 score).

***This is a challenging course that will demand much of your time and effort throughout the semester.***

The university's syllabus policies:

- [University Syllabus Policy S16-9](http://www.sjsu.edu/senate/docs/S16-9.pdf) at <http://www.sjsu.edu/senate/docs/S16-9.pdf>.
- Office of Graduate and Undergraduate Program's [Syllabus Information web page](http://www.sjsu.edu/gup/syllabusinfo/) at <http://www.sjsu.edu/gup/syllabusinfo/>

“Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally 3 hours per unit per week with 1 of the hours used for lecture) for instruction or preparation/studying or course related activities including but not limited to internships, labs, clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.”

## Exams

The exams will test understanding (not memorization) of the material taught during the semester and now well each of you participated in your team assignments and project. Instant messaging, e-mails, texting, tweeting, file sharing, or any other forms of communication with anyone else during the exams will be strictly forbidden.

There can be no make-up quizzes and midterm examination unless there is a documented medical emergency. Make-up final examinations are available only under conditions dictated by University regulations.

## Grading Information

This class is graded credit/no credit (CR/NC). Your individual final class grade will be weighted as follows:

65%	Assignments
15%	Midterm exam
20%	Final exam

Each assignment and exam will be scored (given points) but not assigned a letter grade. The average score will be posted after each assignment and exam.

During the semester, you can keep track of your progress in Canvas. At the end of the semester, all the students will be ranked in order of their weighted class scores. Students who score above a threshold will receive CR, and the rest will receive NC. We expect at least 75% of students to receive CR.

## Zoom Classroom Etiquette

- **Mute your microphone.** To help keep background noise to a minimum, make sure you mute your microphone when you are not speaking.
- **Be mindful of background noise and distractions.** Find a quiet place to “attend” class, to the greatest extent possible.
  - Avoid video setups where people may be walking behind you, people talking, making noise, etc.
  - Avoid activities that could create additional noise, such as shuffling papers, listening to music in the background, etc.
- **Position your camera properly.** Be sure your webcam is in a stable position and focused at eye level.
- **Limit your distractions and avoid multitasking.** You can make it easier to focus on the meeting by turning off notifications, closing or minimizing running apps, and putting your smartphone away (unless you are using it to access Zoom).
- **Use appropriate virtual backgrounds.** If using a virtual background, it should be appropriate and professional and should not suggest or include content that is objectively offensive and demeaning.

## Recording Zoom Classes

This course or portions of this course (i.e., lectures, discussions, student presentations) will be recorded for instructional or educational purposes. The recordings will be posted to the class webpage. The recordings will be deleted at the end of the semester. **If you prefer to remain anonymous** during these recordings, then please communicate with the instructor about possible accommodations (e.g., temporarily turning off identifying information from the Zoom session, including student name and picture, prior to recording).

## Students are Not Allowed to Record

Students are prohibited from recording class activities (including class lectures, office hours, advising sessions, etc.), distributing class recordings, or posting class recordings. Materials created by the instructor for the course (syllabi, lectures and lecture notes,

presentations, etc.) are copyrighted by the instructor. This university policy ([S12-7](#)) is in place to protect the privacy of students in the course, as well as to maintain academic integrity through reducing the instances of cheating. Students who record, distribute, or post these materials will be referred to the Student Conduct and Ethical Development office. Unauthorized recording may violate university and state law. It is the responsibility of students that require special accommodations or assistive technology due to a disability to notify the instructor.

### **Proctoring Software and Exams**

Exams will be proctored in this course through Respondus Monitor, LockDown Browser, and Zoom video meeting. Please note it is the instructor's discretion to determine the method of proctoring. If cheating is suspected the proctored videos may be used for further inspection and may become part of the student's disciplinary record. Note that the proctoring software does not determine whether academic misconduct occurred but does determine whether something irregular occurred that may require further investigation. Students are encouraged to contact the instructor if unexpected interruptions (from a parent or roommate, for example) occur during an exam. Please refer to the online exam instructions for details of the setup and requirements.

### **Technical Difficulties**

- **Internet connection issues:** Canvas autosaves responses a few times per minute as long as there is an internet connection. If your internet connection is lost, Canvas will warn you but allow you to continue working on your exam. A brief loss of internet connection is unlikely to cause you to lose your work. However, a longer loss of connectivity or weak/unstable connection may jeopardize your exam.
- **Other technical difficulties:** Immediately notify the instructor and explain the problem you are facing. Your instructor may not be able to respond immediately or provide technical support. However, the current state of your exam and communication will provide a record of the situation.

Contact the SJSU technical support for Canvas:

Technical Support for Canvas  
[Email: ecampus@sjsu.edu](mailto:ecampus@sjsu.edu)  
Phone: (408) 924-2337  
<https://www.sjsu.edu/ecampus/support/>

### **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 Program's [Syllabus Information web page](#) at <http://www.sjsu.edu/gup/syllabusinfo/>.

# CMPE 180A

## Data Structures and Algorithms in C++

Section 1  
Fall 2020

### Course schedule

Subject to change with fair notice.

Week	Dates	Topics and activities	Savitch	Malik
1	Aug 25	Introduction C++ basics Flow of control Simple input and output (I/O)	1, 2, 3	
2	Sep 1	Procedural abstraction Functions Scope Overloaded function names Call-by-value vs. call-by-reference	4, 5	
3	Sep 8	I/O streams Introduction to classes and objects Arrays, strings, and vectors	6, 7, 8	
4	Sep 15	Pointers New and delete operators Dynamic arrays	9	
5	Sep 22	Structures and classes Object-oriented programming Public and private members Constructors and destructors Friend functions Operator overloading Abstract data types (ADT)	10, 11	1, 2, 3
6	Sep 29	Arrays and vectors of objects Copy constructor How a vector grows Namespaces Linear vs. binary search	12	
7	Oct 6	A “safe” array type The assignment operator and [] operators The “Big Three” Linked lists Stacks and queues	13	

Week	Dates	Topics and activities	Savitch	Malik
8	Oct 13	<i>Midterm exam Tuesday, October 13</i> Inheritance Subclasses Polymorphism Virtual destructors Templates	15	
9	Oct 20	chrono, auto, and decltype Exception handling Standard Template Library (STL) STL vectors and linked lists STL iterators STL sorting	16, 17, 18.1, 18.2	4, 5
10	Oct 27	Constructor and destructor calls Why did my program crash? Recursion Think recursively Iterative vs. recursive binary search Towers of Hanoi	14	6
11	Nov 3	Introduction to algorithm analysis Recurrence relations Proof by induction Big-O notation Rates of growth and scalability Hashing	18.3	9
12	Nov 10	Selection sort Insertion sort Shellsort Mergesort for linked lists Analysis of mergesort Quicksort		10
13	Nov 17	Trees Tree traversals Binary search tree A balanced binary tree		11
14	Nov 24	String algorithms Graph algorithms Multiple inheritance Lambda expressions	18.4	12
15	Dec 1	Function objects Runtime type casting and identification Pointers vs. references Unique and shared smart pointers Move semantics The "Big Five"		

<i>Final Exam</i>	<i>Tuesday Dec 15</i>	Time: 5:15 – 7:30 PM online		
-------------------	-----------------------	--------------------------------	--	--