CS 16 – Solving Problems with Computers I
Syllabus – Spring 2017

<table>
<thead>
<tr>
<th>Class Time:</th>
<th>Tu &amp; Th 12:30pm – 1:45pm</th>
<th>Location:</th>
<th>Buchanan 1930</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instructor:</td>
<td>Ziad Matni</td>
<td>Email:</td>
<td><a href="mailto:zmatni@cs.ucsb.edu">zmatni@cs.ucsb.edu</a></td>
</tr>
<tr>
<td>Office Hours:</td>
<td>Tu 10:00 am – 12:00 am in SMSS 4409</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lab Times:</td>
<td>W 8am, 9am, 10am in PHELP 3525</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class Main Website:</td>
<td><a href="https://ucsb-cs16-s17.github.io/">https://ucsb-cs16-s17.github.io/</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class Piazza Site:</td>
<td><a href="https://piazza.com/ucsb/spring2017/cs16">https://piazza.com/ucsb/spring2017/cs16</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Catalog Description
https://www.cs.ucsb.edu/education/courses/cmpsc-16

Class Overview
This is an intermediate class in computer science that will cover the basic building blocks for solving problems using computers. Specifically, you will learn how to use computational abstractions to solve problems and how to translate these computational abstractions into C++ programs. It’s assumed that you’ve taken both MATH 3A and CS 8/ENG 3 already and that you are familiar enough with introductory topics on computers and programming. By the end of the course, students will be able to:

- **identify** correct C++ syntax and semantics and will be familiar with the common mechanisms of the C++ language
- **apply** Linux tools to create, submit, compile and run C++ programs
- **select** appropriate computational abstractions based on their knowledge of underlying computer systems
- **decompose** complex problems into more manageable parts
- **create** C++ programs that solve application-specific problems

Learning how to program requires time, perseverance, and consistent practice: exactly like practicing a musical instrument, a field sport, or cooking a gourmet meal. There’s a science behind programming, but it is also about technique – and that requires you to “get your hands dirty” and practice, practice, and then practice some more! You are bound to make mistakes – and that’s ok because you will learn from them. Making mistakes means you are learning! Do not be afraid of trying something that you initially have no clue about! Remember that practicing early and often will make you a better programmer in the end. This means that you should not procrastinate and wait until the last moment to do your assignments and homework… but you knew that already, didn’t you!? ;)

What you need to know BEFORE you take this course
This course will present C++ from the beginning; no prior knowledge of C++ is assumed. However, it IS assumed that you already have successfully completed CMPSC 8, or have an equivalent background in programming. You should be comfortable with all of the following:
• Problem solving
  o breaking down a problem into a sequence of steps
  o abstracting specific problems into general ones and finding general solutions
• Memory concepts
  o variables, primitive vs. reference variables, name, type, value
  o assignment statements
  o scope of variables
• Control structures
  o for loops, if/else, while loops
• Arrays (or a similar data structure, e.g. Lists in Python)
  o index vs. value, finding sum, min, max, average, count of elements matching some condition, making a new list of elements containing only those that match some condition
• Functions
  o function call vs. function definition
  o parameters (arguments)
• Testing
  o How to test your code
• Input/output concepts
  o Writing to the terminal
  o Reading from the keyboard
  o Reading and writing to files
  o Neatly formatting output
• Program style
  o How to write code that other people can read and understand

What you will learn by the end of this course to be ready for the next programming class (CS 24)

So, what is it that you need to know by the end of this course? Here’s the list of just a few of the things you’ll need to know to be ready for CS 24 (the next programming course). You’ll have the opportunity to learn all of these things (though not necessarily in this order).

• Basic data types in C++ (int, double, char, bool, string, const)
• The basic control structures in C++ (if/else, while, for etc.)
• Defining functions in C++, and passing parameters to functions in 3 different ways (by value, by pointer, and by reference)
• The basic principles of recursion, and some idea of when a recursive solution is appropriate.
• Scope and lifetime of variables in C++
• Using arrays in C++, and C-strings (null-terminated character arrays)
• Using arrays in functions in C++
• Defining and working with structures (struct) in C++
• Using structures to create singly linked lists
• The difference between the stack and the heap
• Converting from binary to decimal, octal, and hex, and back again—and how this relates to how C++ programs store various kinds of data in memory.
**Required Textbook**

Problem Solving with C++ (9th Edition), 9th Edition  
by Walter Savitch  
ISBN-10: 0133591743

**Class Format**

This is a large lecture class that meets twice a week and is accompanied by a lab. Attending lectures and your lab is **mandatory**. Attendance will be taken in labs and missing too many will result in the instructor taking off up to a half grade from your final grade (e.g. a B- becomes a C+).

| This course has multiple readings, 15-16 homework assignments, 8-9 lab assignments, 2 midterms and one final exam. You will submit homework as a hardcopy in class, submit lab assignments online, and do all the exams in the same classroom. It is really important to do the class readings ahead of time. Also, class participation is vital and highly encouraged (and recognized too!) |

Just as in a math class, everything we do in this class (and almost all classes in CS) builds on all the work that came before. So, everything is cumulative—meaning that you can’t afford to miss any classes unless absolutely necessary. Miss two lectures in a 10-week two-lecture per week course, and you’ve already skipped 10% of the course—it wouldn't be surprising if your performance (i.e. final grade) in the course dipped by a similar amount!

**You may find the workload heavy.** It may even feel unreasonable compared to your other courses. However, I assure you that it is not unreasonable, given the goal of making you a skilled beginning programmer. Programming is a skill, and the only way to get good at it is lots and lots of practice, which takes lots and lots of time. The usual "folklore" rule of thumb is 8–12 hours per week for a normal college class. That means you should expect, at a minimum to put in 5–9 hours per week on this course, on top of the 3 hours 20 minutes you spend in lecture and lab each week.

**Lectures**

The purpose of the lectures in this course is to guide you through the readings, homework, and labs:

- To provide an overview of how everything fits together.
- To provide hands-on demonstrations of things you'll do on your own later.
- To provide additional information that is not in the textbook (and to sometimes clarify the textbook).
- To provide an opportunity to ask questions, and hear answers to questions asked by others.

This course moves quickly. So attendance is very important. We'll sometimes cover two or even three chapters in a given week. As a result, **there will be a homework assignment that you have to turn in at almost every class.**
Homework

In almost every class, you'll be given a homework assignment that is due in the following class – the exception is the class period immediately before an exam, where no homework will be given.

These are typically pencil/paper type problems, though sometimes you'll need access to a computer to solve them. If you don't have reliable access to a computer at home (or in your dorm), please plan your schedule so that you can spend time in the CSIL computer lab between classes.

Homework assignments are completed on paper—they may NOT be submitted electronically—and may ONLY be submitted in person, in the class in which they are due.

You may NOT turn in a homework assignment "on behalf of" an absent classmate, or have someone else turn in your homework for you—doing so in this course is a form of academic dishonesty. You can work with a “homework-buddy”, but you each have to turn in your own work and you have to disclose who you worked with (there’s a place to do that on the homework form).

Again, please do NOT:

- Turn in homework on a day other than when it is due. No late submissions accepted.
- Have someone else turn in your homework for you (that will be considered academic dishonesty).
- Leave homework in a mailbox or slide it under a door.
- Email your homework or upload it anywhere online.
- Copy answers directly from other students or (heaven forbid!) website. Do your own work!
- Forget to cite (i.e. give credit to) your sources, if you consult your textbook, a website, or person.

Labs

The labs meet at PHELP 3525 and are run by the TAs. Attendance is taken and mandatory.

Please do not switch your registered lab sections before clearing it with all TAs involved (space is tight in these labs). You will be given lab assignments every week. You typically will start these in the lab and finish them up on your own time afterwards. The lab assignments have to be turned in by Tuesday at noon, by uploading them using the submit.cs service. You can ONLY turn in your lab assignments on submit.cs.

In some labs, you will be asked to pair up and work with one other partner in the lab. This “pair programming” concept is explained further in another section in this syllabus.

Again, please do NOT:

- Use anything other than submit.cs to submit your lab.
- Turn in labs late. They are due by NOON on Tuesdays (except when indicated otherwise).
- Copy answers directly from other students or (heaven forbid!) website. Do your own work!
- Forget to cite (i.e. give credit to) your sources, if you consult your textbook, a website, or person.
Exams

Both the midterm exams and the final exam are closed book. You may bring one 8.5” x 11” sheet of paper for notes (single sided for midterms; double sided for final), but you have to surrender your note page with your exam. The final exam is cumulative.

The dates are set for the 3 exams and will not change.
If you miss any of them, you will get an F in the class.

Make Up Policy

If you miss a class, you miss the opportunity for the points on that in-class assignment, or homework that was due. Period. Generally speaking, I do not allow for makeups in this class, with few exceptions.

There is no makeup for homework or lab assignments, except for excused absences arranged and agreed to by the instructor in advance. If you don’t turn in an assignment by the due date and time, you will get a zero grade for that assignment. Nonetheless, I will drop the 2 lowest homework grades at the end of the quarter.

There is no makeup for exams. This is a stricter policy than with assignments. The midterms and the final exam dates are announced in this syllabus and are fixed. If you believe you cannot attend any of these dates, especially the final exam, please consider dropping the class.

In rare cases, if there is a documented family emergency, documented extended illness, documented required court appearance, or other situation beyond the students' control (with documentation) the instructor may grant additional make up days entirely at the instructor's discretion—but this is not a guarantee or a right. Asking for accommodation because “I already bought my plane ticket” or “I have out of town guests that week” is a futile exercise that will get you nowhere…

Late Submission Policy

Late submission means 48 hours after deadline. Anything submitted beyond that is graded with a zero. Late submissions of homework will result in a 20% penalty. Recall: homework is due at the start of class and you can only submit homework in class.

Late submissions of lab assignments are NOT GUARANTEED TO BE GRADED! At the very least, they will also result in a 20% penalty.

The primary purpose of the deadlines is to allow the TA to manage his/her workload. So, in regards to the lab assignments, all labs must be graded in one sitting, or the TA will not be able to keep up with the workload. Therefore, if you want your work to be graded without penalty, turn it in on time.
If you turn in your work late, you RISK GETTING A ZERO!
The Use of Laptops and Smartphones in Class

In lecture
Using laptops/tablets in class is a controversial topic these days. As with all technology, they amplify the virtues and vices already present in society. I will allow students to use laptops/tablets in class to take notes, to participate in online activities, and to submit short writing assignments. However, I am extremely aware of the distractions afforded by laptops/tablets.

If I notice your laptop/tablet activities are completely off-task and distracting to students around you, you will be asked to leave class and the class will be counted as an unexcused absence. In general I expect that you will all behave like the adults that you are, recognize that you are paying for the course you are taking, and treat that time with the respect it deserves.

I do not allow the use of cell phones in class. Please turn them off or put them on vibrate mode before you enter the classroom. If your phone causes a distraction in class, or if I (or one of the TAs) notice you using your phone in class, you will be asked to leave class and the class will be counted as an unexcused absence. Additionally, I reserve the right to ban laptops from lecture at any given time if I sense that they detract from the learning outcomes described for the class.

In lab
All labs require you to use the Linux environment. Knowledge of Linux is required for future CS courses (and really for most CS careers you wish to have). By having the entire class using the same programming environment, we reduce the overhead of sorting out operating system-specific complications and we can easily assign pairs for paired-programming assignments. So, please refrain from bringing your laptop to the lab (but feel free to use it to do your work at home).

About Pair Programming
Some (but not all) of the programming work in this course will be done using a style of programming known as “pair programming”. This is where two people work together at the same terminal, as “lab partners”, to solve a programming problem.

For the assignments where pair programming is used, it is required, not optional. Here's why:

- Pair programming is a real-world skill that is highly valued by employers.
  - Many companies use pair programming extensively, including several local area employers of UCSB CS graduates.
- Companies that employ UCSB CS and CE grads tell us that our graduates have good technical skills but need better skills and working in pairs and groups to solve problems.
  - Incorporating pair programming into our curriculum is part of our response to this “real-world” feedback.
- Most students find it helpful and enjoyable—UCSB CS students that were surveyed about their pair programming experiences overwhelmingly reported positive results.
- There is also evidence in the scientific literature that it improves student learning, and helps you get better grades.

To learn more about pair programming, watch the following video (it takes less than 10 minutes). http://bit.ly/pair-programming-video.
We also realize that working in groups has another, potentially less positive, side to it: namely the problem of “freeloaders”. So, please:

- Do NOT “just copy” homework or code from others and claim it as your own work. That is called plagiarism and is subject to harsh consequences from the instructor, the department, and the university.
- Do NOT work together on assignments unless you’ve been specifically told that it is allowed.

The bottom line:
- The instructor will try to be very specific about what kinds of collaboration are permitted, and what kinds of collaboration are not permitted, and are considered a form of academic dishonesty.
- If you are not sure about whether some kind of collaboration is permitted or not, or if dishonesty is taking place, it is your responsibility to ask questions.

**Grading**

Grades will be placed on a webpage with anonymized student perm ID numbers. The webpage URL will be given out in class. Make sure you check with the instructor or the TAs to get your most up-to-date grade. Your participation in class is always noted – those who make interesting contributions in class can expect to receive some extra credit.

<table>
<thead>
<tr>
<th>Item</th>
<th>Grade %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework</td>
<td>20%</td>
</tr>
<tr>
<td>Labs</td>
<td>30%</td>
</tr>
<tr>
<td>Midterms</td>
<td>25%</td>
</tr>
<tr>
<td>Final</td>
<td>25%</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>100 %</strong></td>
</tr>
</tbody>
</table>

**Class Grade Distributions**

These are calculated to 2 decimal places and strictly assigned.

<table>
<thead>
<tr>
<th>Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>[93 – 100]</td>
<td>A</td>
</tr>
<tr>
<td>[90 – 93]</td>
<td>A-</td>
</tr>
<tr>
<td>[87 – 90]</td>
<td>B+</td>
</tr>
<tr>
<td>[83 – 87]</td>
<td>B</td>
</tr>
<tr>
<td>[80 – 83]</td>
<td>B-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Range</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>[77 – 80]</td>
<td>C+</td>
</tr>
<tr>
<td>[73 – 77]</td>
<td>C</td>
</tr>
<tr>
<td>[70 – 73]</td>
<td>C-</td>
</tr>
<tr>
<td>[60 – 70]</td>
<td>D</td>
</tr>
<tr>
<td>&lt; 60</td>
<td>F</td>
</tr>
</tbody>
</table>

[X – Y] means “X to Y inclusive of X (but not Y)”

**A+ grades:** These may be awarded to the very best performing students in the class—but the cutoff for A+ grades will be determined at the end of the course at the discretion of the instructor (there is no predetermined cutoff---an average of 97 or more doesn't guarantee you an A+ grade.)
CS16 (Spring 2017) TEACHING ASSISTANTS

The teaching assistants (TAs) aid the instructor in multiple ways and are responsible to lead the labs, do the grading, proctor classes and exams, and help out students through their office hours.

The TA office hours will be held in Trailer 936.

<table>
<thead>
<tr>
<th>TA NAME</th>
<th>LAB SECTION</th>
<th>OFFICE HOURS</th>
<th>EMAIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sujaya Maiyya</td>
<td>Wed. 8 am</td>
<td>Mon. 3 – 5 pm</td>
<td><a href="mailto:sujaya_maiyya@umail.ucsb.edu">sujaya_maiyya@umail.ucsb.edu</a></td>
</tr>
<tr>
<td>Jinjin Shao</td>
<td>Wed. 9 am</td>
<td>Thu. 3 – 5 pm</td>
<td><a href="mailto:jinjin_shao@umail.ucsb.edu">jinjin_shao@umail.ucsb.edu</a></td>
</tr>
<tr>
<td>Nataly Moreno</td>
<td>Wed. 10 am</td>
<td>Wed. 2 – 4 pm</td>
<td><a href="mailto:nataly_moreno@umail.ucsb.edu">nataly_moreno@umail.ucsb.edu</a></td>
</tr>
<tr>
<td>Bay-Yuan Hsu (Grader)</td>
<td>-</td>
<td>-</td>
<td><a href="mailto:bhsu@umail.ucsb.edu">bhsu@umail.ucsb.edu</a></td>
</tr>
</tbody>
</table>

UCSB Policies on Academic Integrity and Honesty

I adhere strictly to the University’s academic integrity policy. Please cite other people’s work if you are going to refer to it in any of your work.

[Link to policy]

It is expected that students attending the University of California understand and subscribe to the ideal of academic integrity, and are willing to bear individual responsibility for their work. Any work (written or otherwise) submitted to fulfill an academic requirement must represent a student’s original work. Any act of academic dishonesty, such as cheating or plagiarism, will subject a person to University disciplinary action. Using or attempting to use materials, information, study aids, or commercial “research” services not authorized by the instructor of the course constitutes cheating. Representing the words, ideas, or concepts of another person without appropriate attribution is plagiarism. Whenever another person’s written work is utilized, whether it is a single phrase or longer, quotation marks must be used and sources cited. Paraphrasing another’s work, i.e., borrowing the ideas or concepts and putting them into one’s “own” words, must also be acknowledged. Although a person’s state of mind and intention will be considered in determining the University response to an act of academic dishonesty, this in no way lessens the responsibility of the student.

(Section A.2 from: [Link to policy], Student Conduct, General Standards of Conduct)

Disabled Students Program (DSP)

UCSB provides academic accommodations to students with disabilities. Students with disabilities are responsible for ensuring that the Disabled Students Program (DSP) is aware of their disabilities and for providing DSP with appropriate documentation. DSP is located at 2120 Student Resource Building and serves as the campus liaison regarding issues and regulations related to students with disabilities. The DSP staff works in an advisory capacity with a variety of campus departments to ensure that equal access is provided to all disabled students.

If you have a disability that requires accommodation in this class, please go see the DSP very early on in the quarter. I will only honor these types of requests for accommodation via the DSP. More information about the DSP is found here: [Link to DSP]

[Section from policy]
**Class Schedule**

*The lecture topics are subject to change or re-arrangement.*

<table>
<thead>
<tr>
<th>W #</th>
<th>L #</th>
<th>Date</th>
<th>Topics</th>
<th>Textbook Readings</th>
<th>Homework Due (Start of Class)</th>
<th>Lab Due (on Tuesday)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4/4</td>
<td>Intro to the class</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4/6</td>
<td>4/6</td>
<td>C++ Basics</td>
<td>Ch. 1 and 2</td>
<td>HW 1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4/11</td>
<td>Data Representation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Flow of Control &amp; Style</td>
<td>Ch. 2 and 3</td>
<td>HW 2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>4/13</td>
<td>More Flow of Control</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Passing parameters to C++</td>
<td>Ch. 3</td>
<td>HW 3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4/18</td>
<td>Functions in C++</td>
<td>Ch. 4</td>
<td>HW 4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>4/20</td>
<td>Functions in C++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Testing and debugging</td>
<td>Ch. 5</td>
<td>HW 5</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>4/25</td>
<td>MIDTERM 1 (lessons 1 thru 6)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>4/27</td>
<td>I/O Streams in C++</td>
<td>Ch. 6</td>
<td>HW 6</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>5/2</td>
<td>Arrays in C++</td>
<td>Ch. 7</td>
<td>HW 7</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>5/4</td>
<td>Arrays in C++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5/9</td>
<td>5/8</td>
<td>Strings and C-Strings</td>
<td>Ch. 8</td>
<td>HW 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5/11</td>
<td>5/10</td>
<td>Vectors in C++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>5/16</td>
<td>Pointers and Dynamic Arrays</td>
<td>Ch. 9</td>
<td>HW 11</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MIDTERM 2 (lessons 7 thru 12)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>5/23</td>
<td>Separate compilations and <em>makefiles</em></td>
<td>Ch. 12</td>
<td>HW 12</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>5/25</td>
<td>Structures and Classes in C++</td>
<td>Ch. 10</td>
<td>HW 13</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>5/30</td>
<td>Linked Lists</td>
<td>Ch. 13</td>
<td>HW 14</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td>6/1</td>
<td>Recursion in C++</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>17</td>
<td>6/6</td>
<td><em>tbd</em></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>6/8</td>
<td>Review for Final Exam</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Monday, June 12th, 12:00 – 3:00 PM**

**FINAL EXAM (cumulative)**

* Lab 3 is due on Monday (5/1) by noon.
** Lab 6 is due on Monday (5/22) by noon.

Please note the following:

- Students must submit all homework in printed form by start of class per the schedule.
- Students must submit lab assignments on submit.cs.ucsb.edu by that week’s Tuesday by noon (except for Labs 3 and 6).
- The midterms will be taken in class, at the start of class (so don’t be late!), and will take up the entire class period.
- The final exam is set by the university. Instructors have no control over setting final exams dates and times.