Overview
This course introduces students to the field of computer science and engineering. An overview of the disciplines within computer science such as networks, AI, robotics, graphics, and computer architecture will be integrated throughout the course. Starting from first principles of computer organization, students will receive a foundation in programming focusing on C++. Fundamental programming concepts along with current issues such as image processing and embedded systems will be covered through relevant programming projects. The course will culminate in a comprehensive programming assignment and/or a team-based robotics project that integrates the concepts taught in the course. A lecture/lab course format will be employed to provide hands-on experience and active learning techniques.

Learning Objectives
Upon completion of this course students will be able to:
1. Define and discuss the disciplines of computer science with the intention of choosing future courses that are applicable to the student’s personal learning goals.
2. Understand the key hardware components in a modern computer system and how software is mapped to the HW.
3. Use a computer to solve problems by developing simple algorithms and then implement them using a specific programming language
4. Implement key algorithms within the field
5. Understand and determine the computational complexity of simple algorithms
6. Write computer programs using conditional and iterative structures, functional decomposition
7. Select an appropriate basic data structure (e.g. arrays) and access methods (e.g. pointers)
8. Understand basic object-oriented principles.
9. Design non-trivial embedded software to control a robot to navigate and interact within a controlled environment.

Prerequisite
None, but a desire to learn.

Lecture
Monday/Wednesday 12:00-1:50 pm in SAL 126
Tuesday/Thursday 12:00-1:50 pm in SAL 126
Tuesday/Thursday 3:30-5:20 pm in WPH 207

Lab
Friday 1:00-3:50pm in MHP 101
Study Sessions
GFS 109
Tuesday 6-7pm, 7-8pm, 8-9pm
Wednesday 8-9pm
Thursday 5:30-6:30pm, 7-8pm, 8-9pm

2 Textbooks
2. Learning Computing With Robots In C++, Deepak Kumar

Professor
Dr. Sheila Tejada
Office: SAL 316
E-mail: stejada@usc.edu
Website: http://bcf.usc.edu/~stejada
Office Hours:
Monday/Wednesday: 3-5pm
Tuesday/Thursday: 2-3pm
and by appointment

TAs
Kaveh Shahabi (kshahabi@usc.edu) (Lead TA)
Jeffery Tanedo (jtanedo@usc.edu)

Undergraduate Teaching Aides (UAs)
Eric Kapitanski (kapitans@usc.edu)
Michael Borke (mborke@usc.edu)
Francesca Nannizzi (nannizzi@usc.edu)
Janice Leung (janicejl@usc.edu)
Luke Shiffer (shiffer@usc.edu)

Graders:
Yi Zhang (zhan954@usc.edu)
Rattan Priya (prattan@usc.edu)

Grading:
The following point structure will be used in determining the grade for the course.

<table>
<thead>
<tr>
<th>Component</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Participation &amp; Activities</td>
<td>20%</td>
</tr>
<tr>
<td>Programming Assignments</td>
<td>40%</td>
</tr>
<tr>
<td>Midterm Exams</td>
<td>20%</td>
</tr>
<tr>
<td>Final Exam</td>
<td>20%</td>
</tr>
</tbody>
</table>

Final letter grades for the course will follow this scale:

<table>
<thead>
<tr>
<th>Percentage</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;95%</td>
<td>A</td>
</tr>
<tr>
<td>95-90%</td>
<td>A-</td>
</tr>
<tr>
<td>89-87%</td>
<td>B+</td>
</tr>
<tr>
<td>86-83%</td>
<td>B</td>
</tr>
<tr>
<td>82-80%</td>
<td>B-</td>
</tr>
<tr>
<td>79-77%</td>
<td>C+</td>
</tr>
<tr>
<td>76-73%</td>
<td>C</td>
</tr>
<tr>
<td>72-70%</td>
<td>C-</td>
</tr>
<tr>
<td>69-67%</td>
<td>D+</td>
</tr>
<tr>
<td>66-63%</td>
<td>D</td>
</tr>
<tr>
<td>62-60%</td>
<td>D-</td>
</tr>
<tr>
<td>&lt; 60%</td>
<td>F</td>
</tr>
</tbody>
</table>

Participation & Activities
Unlike some traditional classroom settings where the instructor talks and students listen, we will attempt to create a classroom environment where the instructor facilitates active student participation in their own
learning process. Students are expected to set their own learning goals (i.e. be curious) and then actively pursue those goals **both in and out of the classroom** through personal study, programming, and in-class activities. Simply showing up to class is not enough; come to class ready to think, ask questions, and work with your fellow students. Small in-class and out-of-class activities (both individual and group-based) will be provided to help facilitate achievement of learning goals. Students may miss 2 activities due to sickness or absence without penalty. Participation & Activities are graded based on completion, not on correctness.

Types of Participation & Activities:
1. Problem Sets: Assigned problems from the textbooks (Malik or LRCCpp) or handouts to be submitted at study sessions.
2. Lab Assignment: Small programming assignment to familiarize students with a programming concept designed to be completed during Friday Lab time.
3. On-line reviews: On-line forms with questions and problems.
4. In-class exercises: Individual and group activities and problems
5. In-class presentations: brief presentations to the class on a CS topic or problem

**Programming Assignments**

Programming assignments are larger more comprehensive assignments that should challenge students to integrate several programming concepts. **They are to be completed individually unless otherwise noted.** (A few group assignments may be scattered throughout the semester). A separate document will be provided listing the expectation for the submission, style, and documentation of programming assignments as well as rubrics for how assignments will be graded. Copying (and then modification) of any portion of code from Internet sources or fellow students is prohibited unless cleared with the instructor. See the Statement on Academic Integrity.

**Reading Assignments**

Readings from the book contain theoretical concepts, examples and usable code that will be very helpful for all the work in this course.

**Course Exams**

Students will be expected to know the material from the assigned readings in the book, the in-class activities, and the homework. The exams are a student’s chance to demonstrate that they fully understand the course material. Exams are closed book. They will cover the lectures, readings, activities, and homework. For absences due to illness, a doctor’s note is **required** as proof of illness or emergency. There are no make-up exams, but providing the instructor with a doctor’s note will add the weight of the missed exam to the next exam.

**Extra Credit**

There will be opportunities during the semester to earn extra credit points. Students who earn 100 extra credit points by the end of the semester and whose final grade is greater than 60% are eligible to have
their final grade increased by at most %3 to the next letter grade.

**Policies**

**Statement for 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. The phone number for DSP is (213) 740-0776.

**Statement on 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* contains the Student Conduct Code in Section 11.00, while the recommended sanctions are located in Appendix A: [http://www.usc.edu/dept/publications/SCAMPUS/gov/](http://www.usc.edu/dept/publications/SCAMPUS/gov/). Students will be referred to the Office of Student Judicial Affairs and Community Standards for further review, should there be any suspicion of academic dishonesty. The Review process can be found at: [http://www.usc.edu/student-affairs/SJACS/](http://www.usc.edu/student-affairs/SJACS/).

**Homework Submission**

Please also note that you must submit your code using the [USC Blackboard System](http://www.usc.edu/), since the Blackboard System timestamps your submission. You should also verify what you have submitted is what you intended to submit.

Please note that **it is your responsibility** to ensure that **you have submitted valid submissions**.

**Late Policy**

All homework must be turned in on time. Late submissions will receive severe penalties. If you submit within 24 hours after the grace period, you will receive 80% of your grade. If you submit within 48 hours after the grace period, you will receive 50% of your grade. If you are unable to complete a homework assignment due to illness or family emergency, please see the instructor as soon as possible to get an extension. A doctor's note is **required** as proof of illness or emergency. In general, when you get sick, it's best to see a doctor and get a note just in case you may need it later.
Week 1 – Introduction to Computer Science, Basic Computer Organization, and Simple I/O
Reading: Chapter 1 Overview p2-22, Chapter 2 C++ Basics p27-72 (Malik)
Lab: LAB 1 Introduction to Aludra, Piazza, Blackboard, HTML, USC email, “Hello, World”
Programming Assignment: Homework 1 CSI: Computer Science Investigator

Week 2 – Algorithmic Thinking and Control Structures
Reading: Chapter 4 Control Structures (Selection) p183-229, Chapter 5 Control Structures II (Repetition) p259-313 (Malik)
Programming Assignment: Homework 2 Choose Your Own Adventure

Week 3 – Basic Program Design and Abstractions; Program Decomposition and Functions
Reading: Chapter 6 User-defined functions p336-390 (Malik)
Lab: Lab 2 ASCII Art

Week 4 – Arrays, Multi-dimensional Arrays and File I/O
Reading: Chapter 8 Arrays and Strings p506-535,542-571, Chapter 3 Input/Output p121-131,142,153-170 (Malik)
Lab: Lab 3 Sudoku
Programming Assignment: Homework 3 Encryption

Week 5 – Structs/ Review for Midterm 1
Reading: Chapter 9 Records (struct) p592-607 (Malik)
** MIDTERM 1**

Week 6 – Data Storage/Representation, Pointers
Reading: Appendix E: Additional C++ Topics p1525-1530, Chapter 12 Pointers p782-806 (Malik)
Programming Assignment: Homework 4 Social Networks

Week 7 – Pointers and References
Reading: Chapter 12 Pointers p782-806 (Malik)
Lab: Lab 4 Swapit!

Week 8 – Dynamic Memory Allocation, Abstract Data Types, and Classes
Reading: Chapter 10 Classes p630-672, Chapter 7 Strings p476-494 (Malik)
Lab: Lab 5 ChromaKey
Programming Assignment: Homework 5 Big Data

Week 9 – Vectors/ Review for Midterm 2
Reading: Chapter 21 STL p1401-1412,1417-1421 (Malik)
** MIDTERM 2**

Week 10 – Robots in Education/ Ethics
Reading: Learning Computing With Robots In C++, Chapter 1-5
Lab: Lab 6 Ethics
Programming Assignment: Homework 6 What is Computer Science?

Week 11 – HCI: Human Computer Interaction
Reading: Learning Computing With Robots In C++
Lab: Lab 7 Robots Got Talent!
Week 12 – Sorting, Big-O notation and complexity
  Reading: Chapter 18 Searching and Sorting Algorithms p1224-1227,1234-1256 (Malik)
  Lab: What is Computer Science? Robot Demonstrations

Week 13 – Image Processing
  Reading: Learning Computing With Robots In C++, Chapter 7-12
  Lab: Lab 8 USAR: Urban Search and Rescue
  Programming Assignment: Homework 7 Computer Vision

Week 14 – Recursion
  Reading: Chapter 15 Recursion p1001-1028 (Malik)
  Lab: Lab 9 Mars Rover

Week 15 – STL/Review for Final
  Reading: Chapter 21 STL p1401-1412,1417-1421 (Malik)
  Lab: Review for Final

** Final Exam – Tuesday, May 14 at 4:30-6:30pm **

Syllabus is subject to change.