Welcome to CSci 1113

Introduction to C/C++ Programming for Scientists and Engineers

YOU KNOW THIS METAL RECTANGLE FULL OF LITTLE LIGHTS?

YEAH.

I SPEND MOST OF MY LIFE PRESSING BUTTONS TO MAKE THE PATTERN OF LIGHTS CHANGE HOWEVER I WANT.

SOUNDS GOOD.

BUT TODAY, THE PATTERN OF LIGHTS IS ALL WRONG!

OH GOD! TRY PRESSING MORE BUTTONS!

IT'S NOT HELPING!
Instructor (me)

James Parker
Shepherd Laboratories 391

Primary contact:
jparker@cs.umn.edu
Karthik Unnikrishnan, Prashanth Venkatesh, Jackson Benning, Yanjun Cui, Mitchell Dillon, Skye Gagnon, Jacob Hammer, Samuel Highbargin, Lin Huynh, Shane Jung, Jin Hong Kuan, Jan-Wei Lim, Haoran Liu, Ying Lu, Sophia Manicor, Andrew McCullough, Adam McCune, Kyle Meng, Brandon Nee, Tanner Skluzacek, Antonio Turley, Ruobing Wang, Kaiwei Wu, Yuyang Xiao, Songyu Yan, Lei Zhang, Xintong Zhang
Questions?

Direct questions to:
Canvas forum discussion
jparker@cs.umn.edu
Textbook

Problem Solving With C++, Walter Savitch, 10th edition
Sister course: CSci 1115

This course is an “introduction” (from start), but many find it difficult.

We started to run a supplementary course to provide additional help: CSci 1115(Th 6pm)
Sister course: CSci 1115

This course is an “introduction” (from start), but many find it difficult.

We started to run a supplementary course to provide additional help: CSci 1115(Th 6pm)
Sister course: CSci 1115

This course is an “introduction” (from start), but many find it difficult

We started to run a supplementary course to provide additional help: CSci 1115(Th 6pm)

Daniel

Me
This course is an “introduction” (from start), but many find it difficult.

We started to run a supplementary course to provide additional help: CSci 1115(Th 6pm)

-group problem solving
Sister course: CSci 1115

This course is an “introduction” (from start), but many find it difficult

We started to run a supplementary course to provide additional help: CSci 1115(Th 6pm)

- group problem solving
- free food!
You need a CSELabs account to participate in labs in this course.

Lab attendance is mandatory (please make an account!)

CSELabs account
https://cseit.umn.edu/
https://cseit.umn.edu/
Welcome to the Fall2012 CSE Labs Account Creation Form.

Use this form to initiate or change your CSE Labs account for the Fall2012 semester. CSE Labs use is open to any student currently enrolled in the College of Science and Engineering.

Please enter the following information:

- Your student email **username**.
- Your **password** for your general UMN email account. (To verify your eligibility for a CSE Labs account.)

Username: park0580@umn.edu
Password: **************

If you do not know what your username is, or you are having problems see the [U of M Student Internet Account Initiation Form](https://www.cs.umn.edu/). For further information send email to operator@cse.labs.umn.edu or stop by the Systems Staff Office in Keller Hall 1-213.

For a list of our hours see [Systems Staff Contact Information and Hours](https://www.cs.umn.edu/).

[Submit](https://cseit.umn.edu/)
CSELabs account

CSELabs account used in lab (first lab ensures account working)

Register ASAP

Problems?
Bug operator@cselabs.umn.edu
Class website

www.cs.umn.edu/academics/classes
Or google “umn.edu csci class”

Syllabus, schedule, other goodies

Canvas page will have grades and (maybe) homework submissions
Class website

Canvas also has a link to the website:
# CSci 1113: C++ Programming

## Schedule

This is an approximate schedule. It will be updated as the class progresses.

<table>
<thead>
<tr>
<th>Week</th>
<th>Week Of</th>
<th>Topics</th>
<th>Lecture Materials (001)</th>
<th>Lecture Materials (010)</th>
<th>Readings</th>
<th>Exams</th>
<th>Lab</th>
<th>Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sept. 4</td>
<td>Introduction, computers, algorithms, programs, compilers</td>
<td>slides</td>
<td>Ch. 1</td>
<td>Unix tutorial (no lab this week)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sept. 10</td>
<td>Variables, expressions, assignment, console I/O, predefined functions</td>
<td></td>
<td>Ch. 2, Section 4.2</td>
<td>Lab 1: Basic C++ programs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Sept. 17</td>
<td>Selection, boolean expressions, if-else, multway-if, switch</td>
<td></td>
<td>Sections 3.1, 3.2</td>
<td>Lab 2: Sequence and selection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Sept. 24</td>
<td>Iteration, while loops, for loops, loop paradigms</td>
<td></td>
<td>Sections 3.3, 3.4</td>
<td>Lab 3: Iteration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Oct. 1</td>
<td>User-defined functions, procedural abstractions</td>
<td>10/3--Quiz</td>
<td>Ch. 4, 5</td>
<td>Quiz Covers Ch 1-3.2 (up to week 3: if-else)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>10/2--Quiz</td>
<td></td>
<td>Lab 4: User defined functions</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Moodle (grades and hw submission)
Syllabus

15% Labs
30% Homework (due Fridays)
  5% Quiz (Feb. 19)
10% Midterm 1 (March 4)
15% Midterm 2 (April 15)
25% Final (Monday May 11, 10:30am to 12:30am in this room)
Each week there will be either a homework due or a test.

Homework is due Fridays at 11:55 P.M. (more info to come)

Late homework is not accepted, but we will drop the lowest one.
Syllabus

Labs can be checked off up until a week after the lab ("warm up" Qs must be checked off in your lab)

Homework must be done by yourself

Don't cheat

Really... don't cheat
Homework

Homework will be both a creative and problem solving endeavor:

Lego example
Build a castle with:
- 4 walls enclosing
- Door
- At least one tower (higher than wall)
Homework
Exams

All exams will be open book/notes
Electronic notes okay
(no memorization)

You **cannot:**
1. Use the internet (no typing)
2. Compile/run programs
3. Talk to or copy from others
Syllabus

Grading scale:

93% A
90% A-
87% B+
83% B
80% B-

77% C+
73% C
70% C-
67% D+
60% D
Below F
Schedule

Ch. 1: Introduction, Programs, Compilers
Ch. 2: Input/Output, Data, Expressions
Ch. 3: Control Flow (if and loops)
Ch. 4, 5: Functions (return values)
Ch. 6: File I/O
Ch. 7, 8: Arrays and Strings
Ch. 9: Pointers and Dynamic Arrays
Ch. 10&11: Classes and Operator Overloading
Ch. 14&15: Recursion & Inheritance
Syllabus

Any questions?
What can I program?

If you can think of an explicit process (of simple steps) to solve your problem, then it can be programed.
Banana Nut Bread

Directions
1. Preheat the oven to 350°F (175°C).
2. Mix butter into the mashed bananas in a large mixing bowl.
3. Mix in the sugar, egg, and vanilla.
4. Sprinkle the baking soda and salt over the mixture and mix in.
5. Add the flour and nuts last, mix.
6. Pour mixture into a buttered 4x8 inch loaf pan.
Repetitive tasks

If you feel like a mindless zombie when you do it a lot, you can probably program it.
Repetitive tasks
Repetitive tasks
Repetitive tasks

Simply move the blue ball with your mouse and avoid the red balls. Easy? Definitely not!
ATMs

How do you get change for $18.26 with the least amount of bills and coins?
Auto leveling?
Software vs Hardware

Software - the more intangible code on a computer

Hardware - the physical Parts of the computer
Hardware interaction

Input → CPU → Memory → Output
Memory addressing

Data is stored in “addresses” inside the memory.

Later in this class, we will use these addresses to manipulate and share data.
Memory addressing
Object oriented programming

OOP - focus on data and how they interact

To make algorithms for OOP, it is often useful to identify the data you are working with and their relationships before programming.
Object oriented programming

Data for...

Banana nut bread?
ATM?
Ball game?
Object oriented programming

Data for...

Banana nut bread? Ingredients
ATM?
Ball game?
Object oriented programming

Data for...

Banana nut bread? Ingredients
ATM? Dollars & coins
Ball game?
Object oriented programming

Data for...

Banana nut bread? Ingredients
ATM? Dollars & coins
Ball game? Balls & mouse
Object oriented programming

Data for...

Banana nut bread? Ingredients
ATM? Dollars & coins
Ball game? Balls & mouse

Lots of pixels (tiny color dots)