Welcome to CSci 1113

Introduction to C/C++ Programming for Scientists and Engineers
Instructor (me)

James Parker
Shepherd Laboratories 391

Primary contact:
jparker@cs.umn.edu
Raunak Manekar, Jiawei Mo, Jackson Benning, Violet Chang, Aaron Duebner, Nicholas Freiter, Nickhil Gupta, Adam Hodapp, Kyle Houser, Aaron Koenigsberg, Michael Markiewicz, AJ Sakher, Guangyu Yan, Tiannan Zhou, Laura Ziegelski
Questions?

Direct questions to:
Moodle forum discussion
jparker@cs.umn.edu
Problem Solving With C++, Walter Savitch, 10th edition
CSELabs account

You need a CSELabs account to participate in labs in this course

Lab attendance is mandatory (please make an account!)
https://cseit.umn.edu/
Forms

- Classroom Access Form
- CSE Account Authorization Form
- CSE Labs Classroom Reservation Form
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.

For further information send email to operator@cselabs.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.

Submit
CSE Labs account

CSE Labs 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

Moodle page will have grades and homework submissions
Class website

Moodle 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</th>
<th>Lecture Materials</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>(001) slides</td>
<td>(010) Ch. 1</td>
<td>Unix tutorial</td>
<td>(no lab this week)</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></td>
<td>Ch. 2, Section 4.2</td>
<td>Lab 1: Basic C++ programs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Sept. 17</td>
<td>Selection, boolean expressions, if-else, multeway-if, switch</td>
<td>Sections 3.1, 3.2</td>
<td></td>
<td>Lab 2: Sequence and selection</td>
<td>HW 0, Wednesday Sept. 19 at 11:00 P.M.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Sept. 24</td>
<td>Iteration, while loops, for loops, loop paradigms</td>
<td>Sections 3.3, 3.4</td>
<td></td>
<td>Lab 3: Iteration</td>
<td>HW 1, Wednesday Sept. 26 at 11:00 P.M.</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>10/2--Quiz</td>
<td>Ch. 4, 5</td>
<td>Quiz Covers Ch 1-3.2 (up to week 3: if-)</td>
<td>Lab 4: User defined functions</td>
<td>HW 2, Wednesday Oct. 3 at 11:00 P.M.</td>
</tr>
</tbody>
</table>
Syllabus

15% Labs
30% Homework (due Wednesdays)
  5% Quiz (Oct. 3)
10% Midterm 1 (Oct. 17)
15% Midterm 2 (Nov. 21)
25% Final (Monday Dec. 17, 8:00am to 10:00am in this room)
Syllabus

Each week there will be either a homework due or a test.

Homework is due Wednesdays at 11:00 P.M. on Moodle.

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

Homework must be coded individually

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

Simply move the blue ball with your mouse and avoid the red balls. Easy? Definitely not!
How do you get change for $18.26 with the least amount of bills and coins?
Repetitive tasks

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