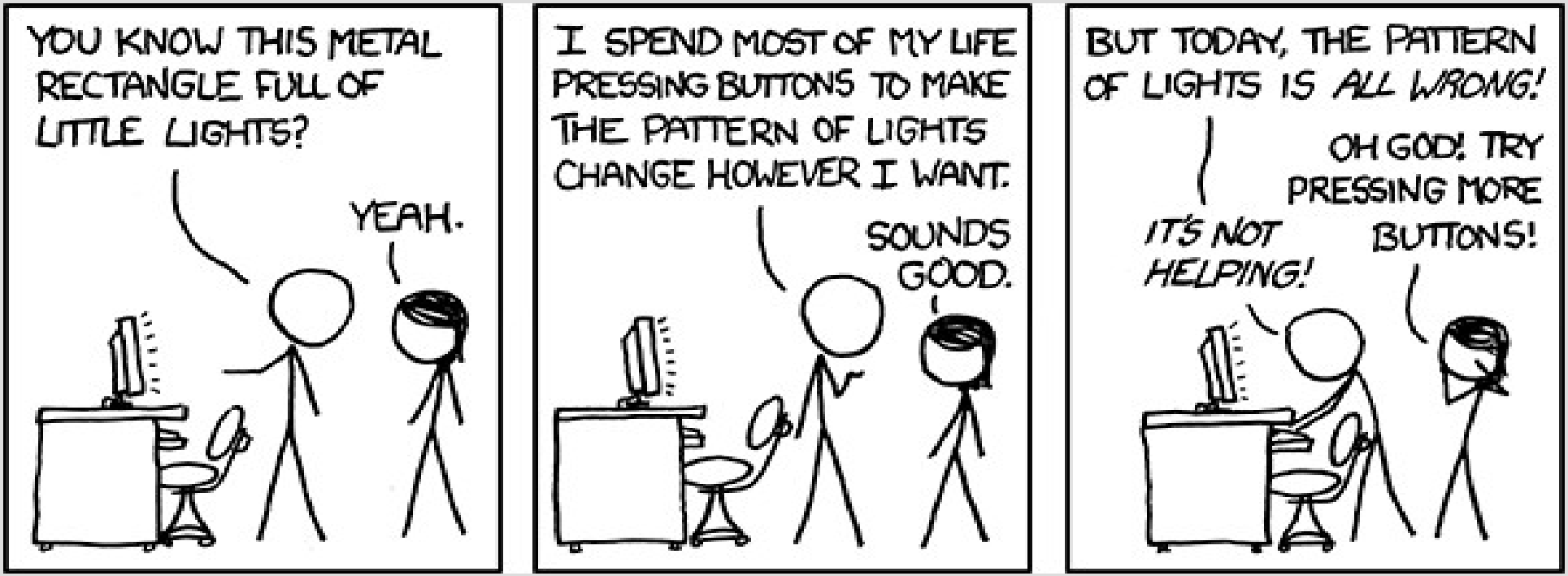
Welcome to CSci 1113

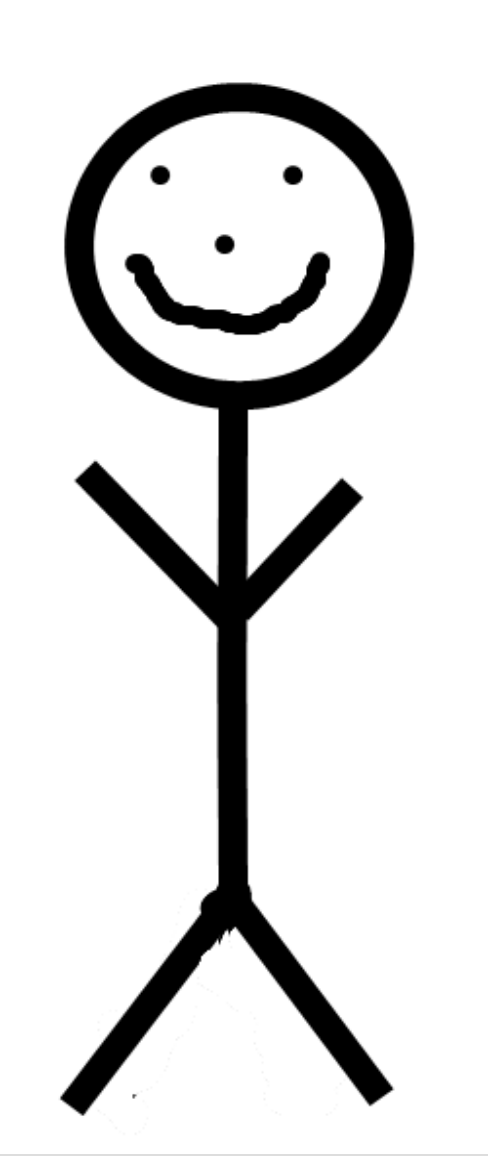
Introduction to C/C++ Programming for Scientists and Engineers



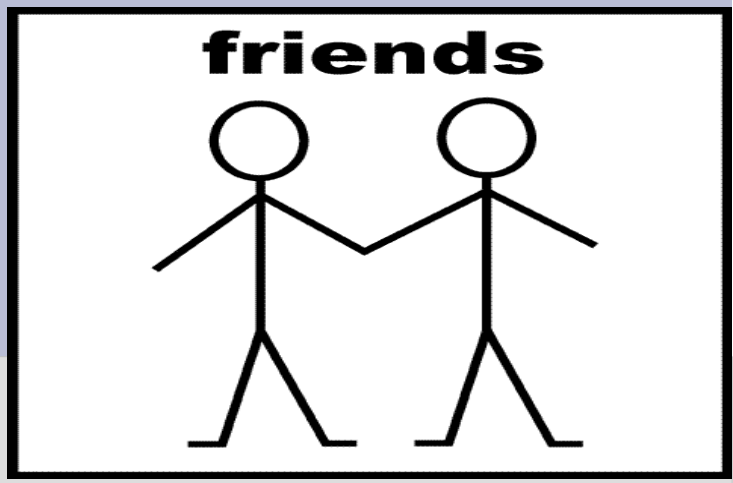
Instructor (me)

James Parker
Shepherd Laboratories 391

Primary contact:
jparker@cs.umn.edu



TAs



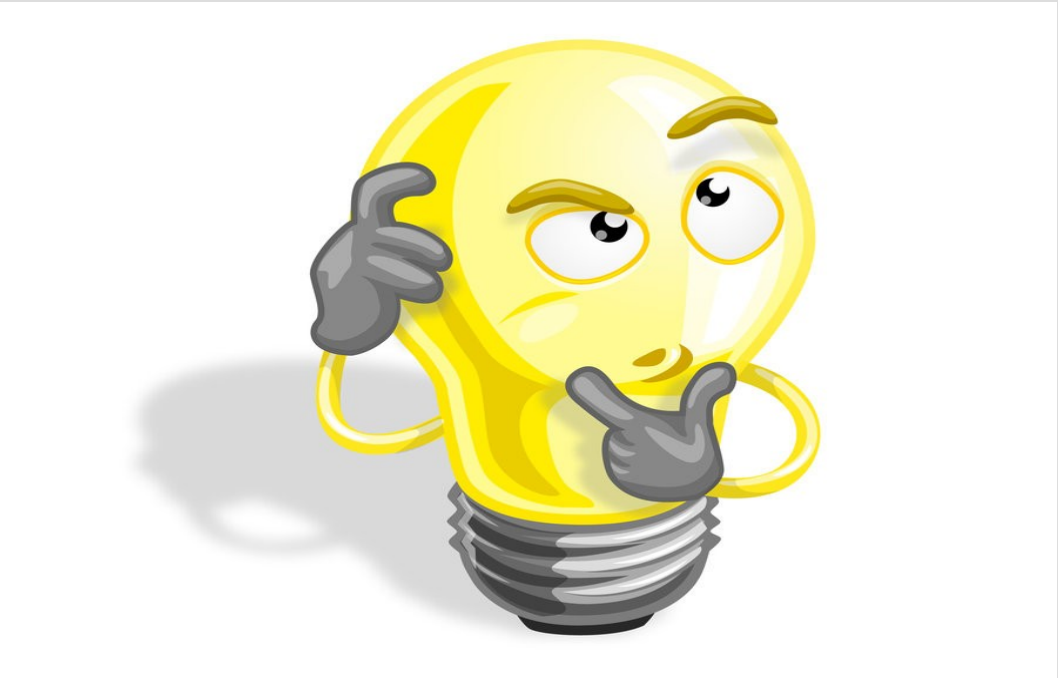
Karthik Unnikrishnan, Prashanth Venkatesh,
Jackson Benning, Yanjun Cui, Mitchell Dillon,
Skye Gagnon, Jacob Hammer, Samuel Highbargin,
Lin Huynh, Shane Jung, Jin Hong Kuan, Jan-Wei Lim,
Haoran Liu, Ying Lu, Sophia Manicor,
Andrew McCullough, Adam McCune, Kyle Meng,
Brandon Nee, Tanner Skluzacek, Antonio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

Questions?

Direct questions to:

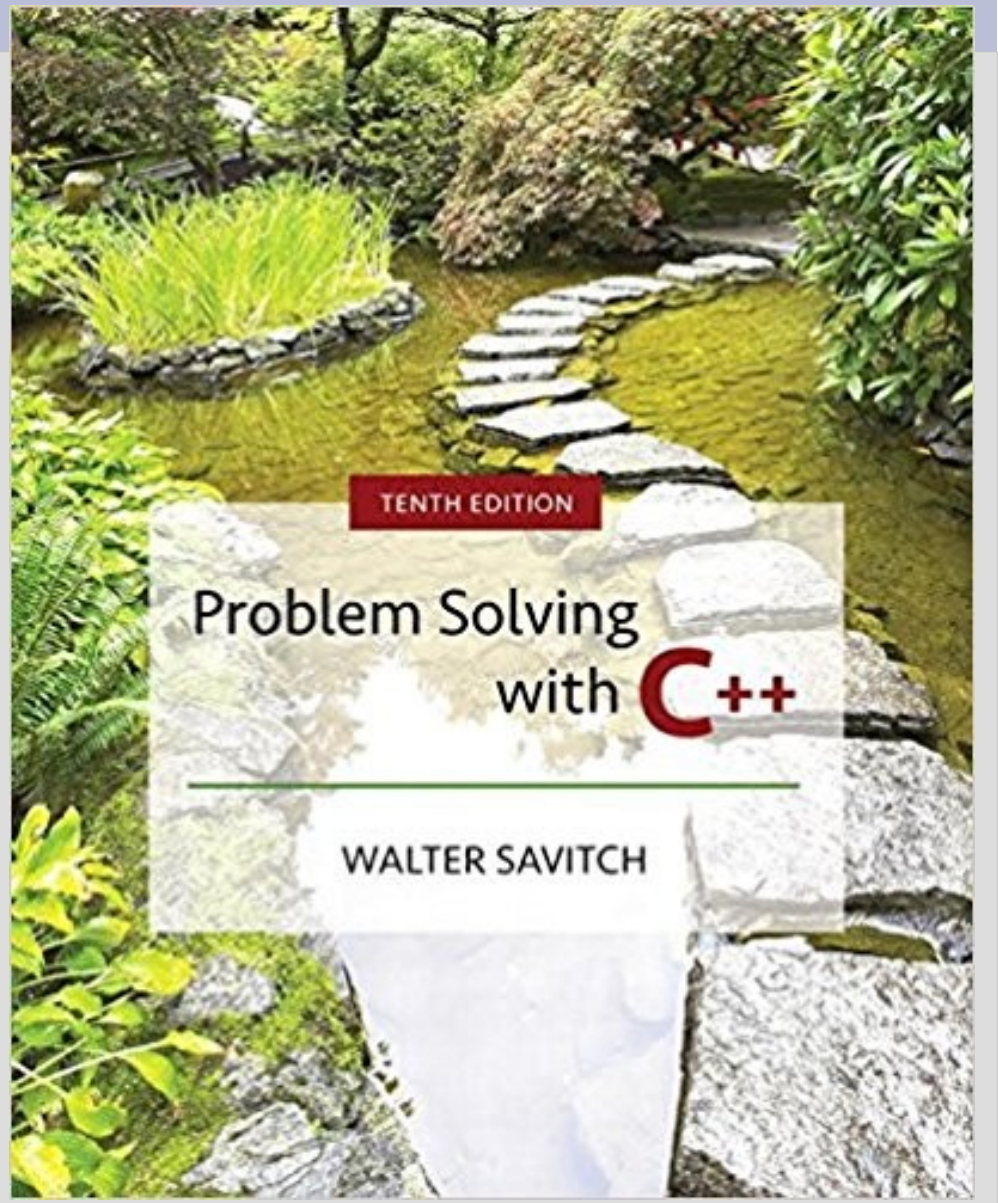
Canvas forum discussion

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)

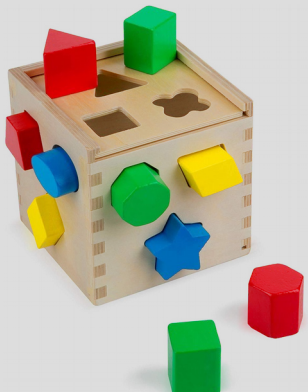

Daniel


Me

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)

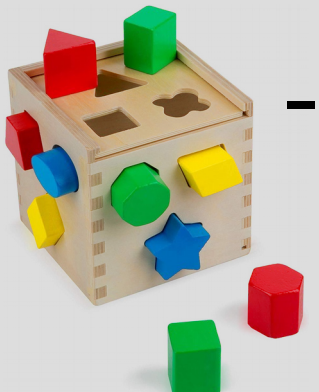


-group problem solving

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)



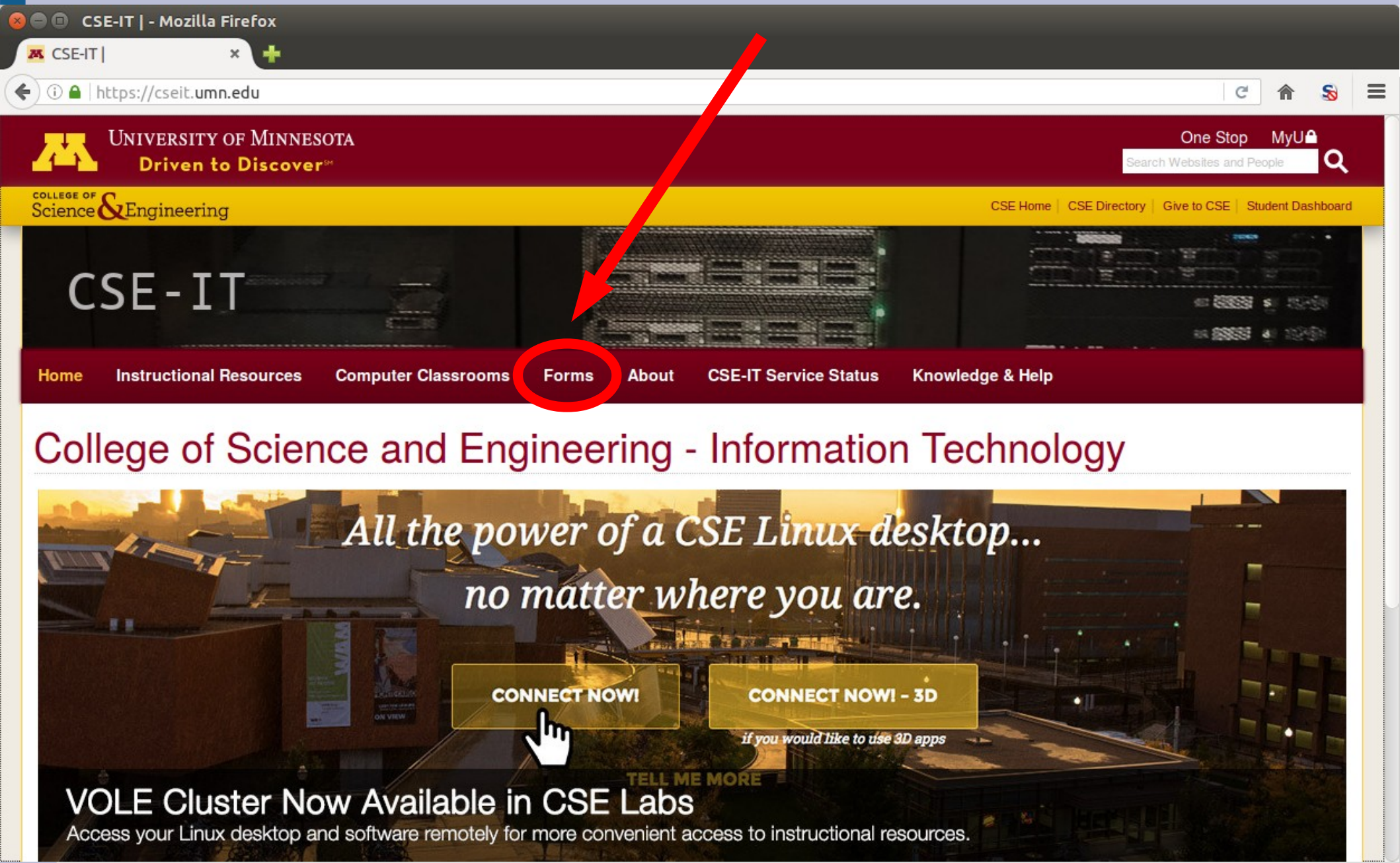
-group problem solving
-free food!

CSELabs account

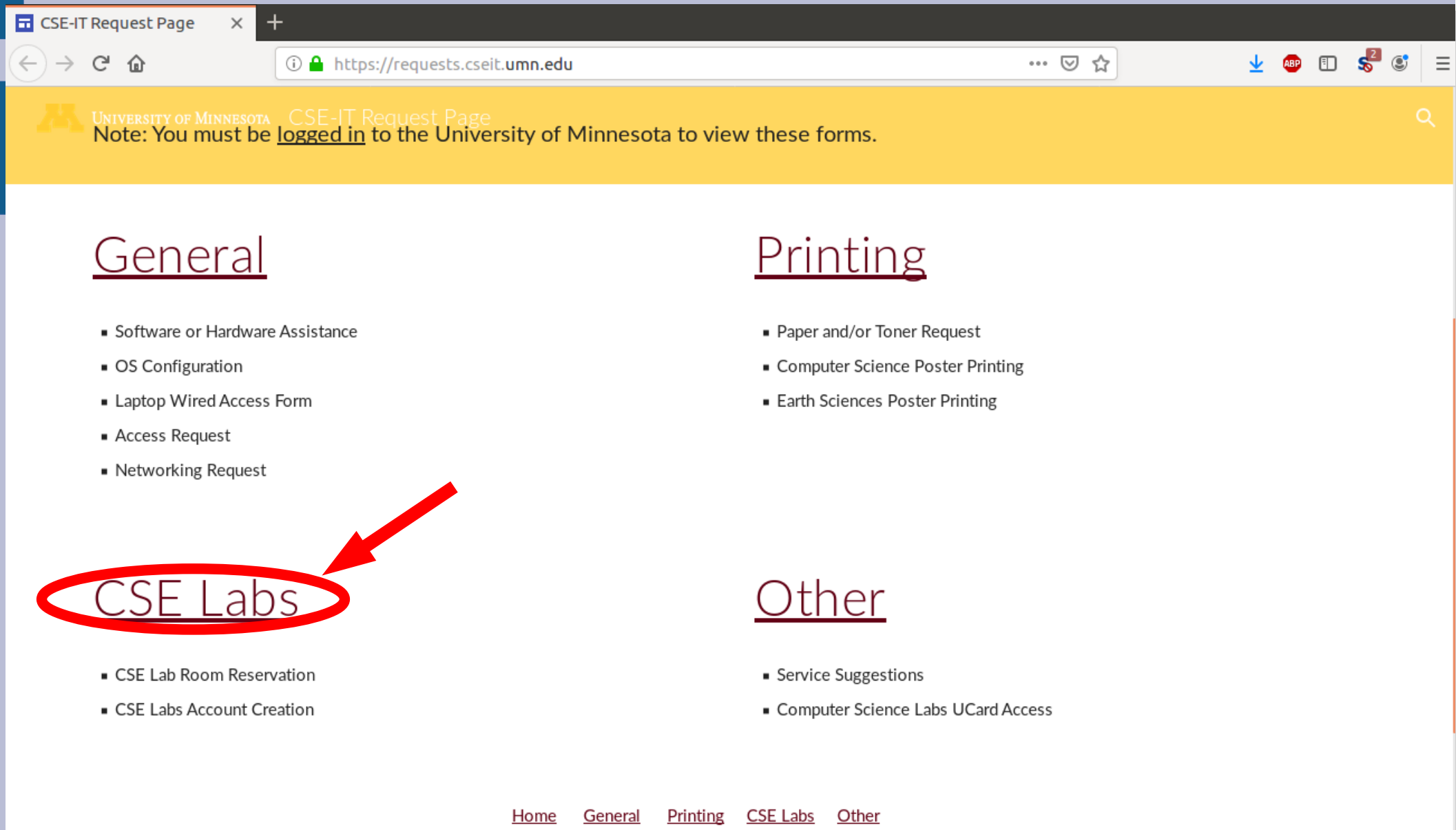
You need a CSELabs account to participate in labs in this course

Lab attendance is mandatory (please make an account!)

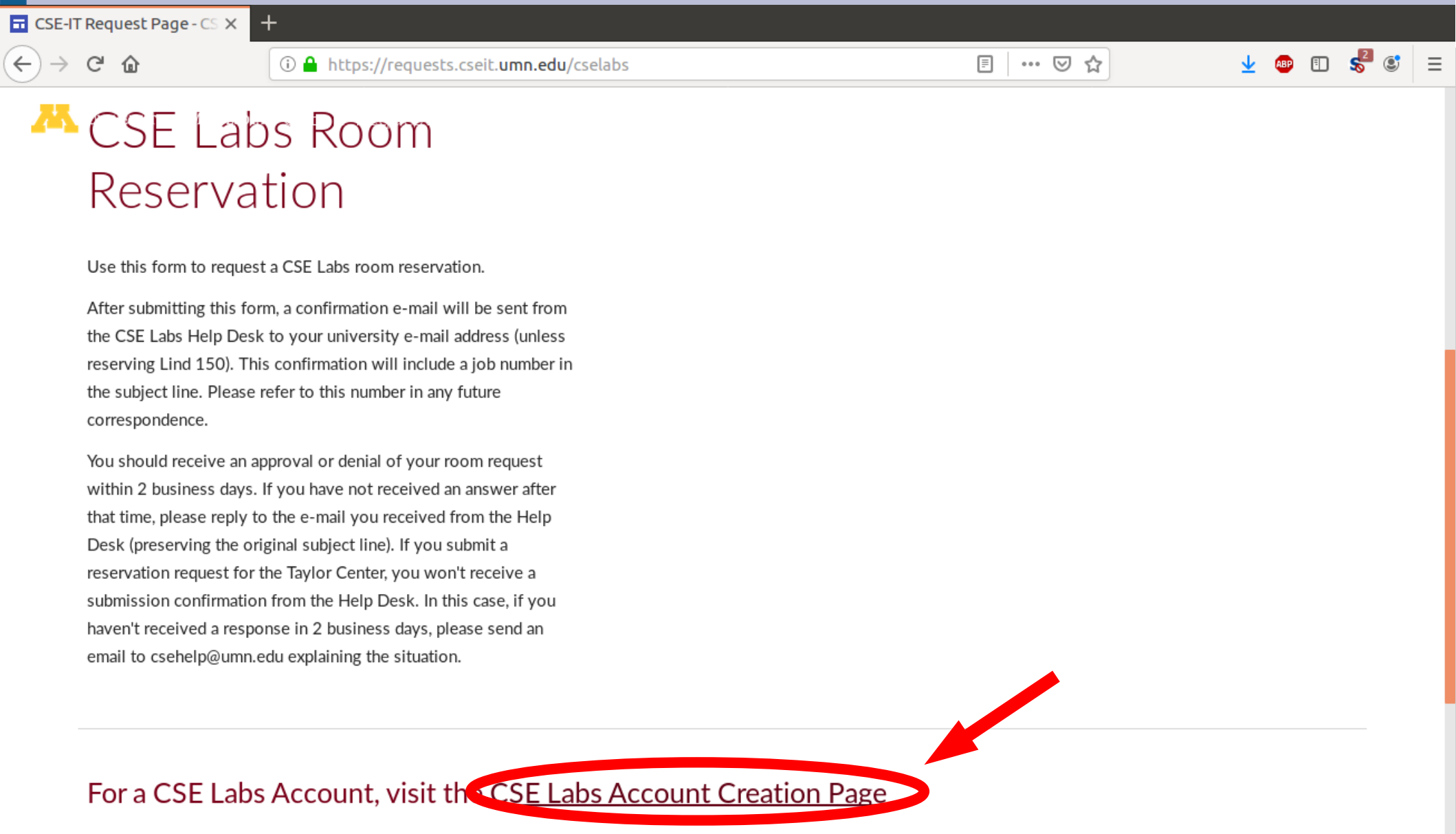
https://cseit.umn.edu/



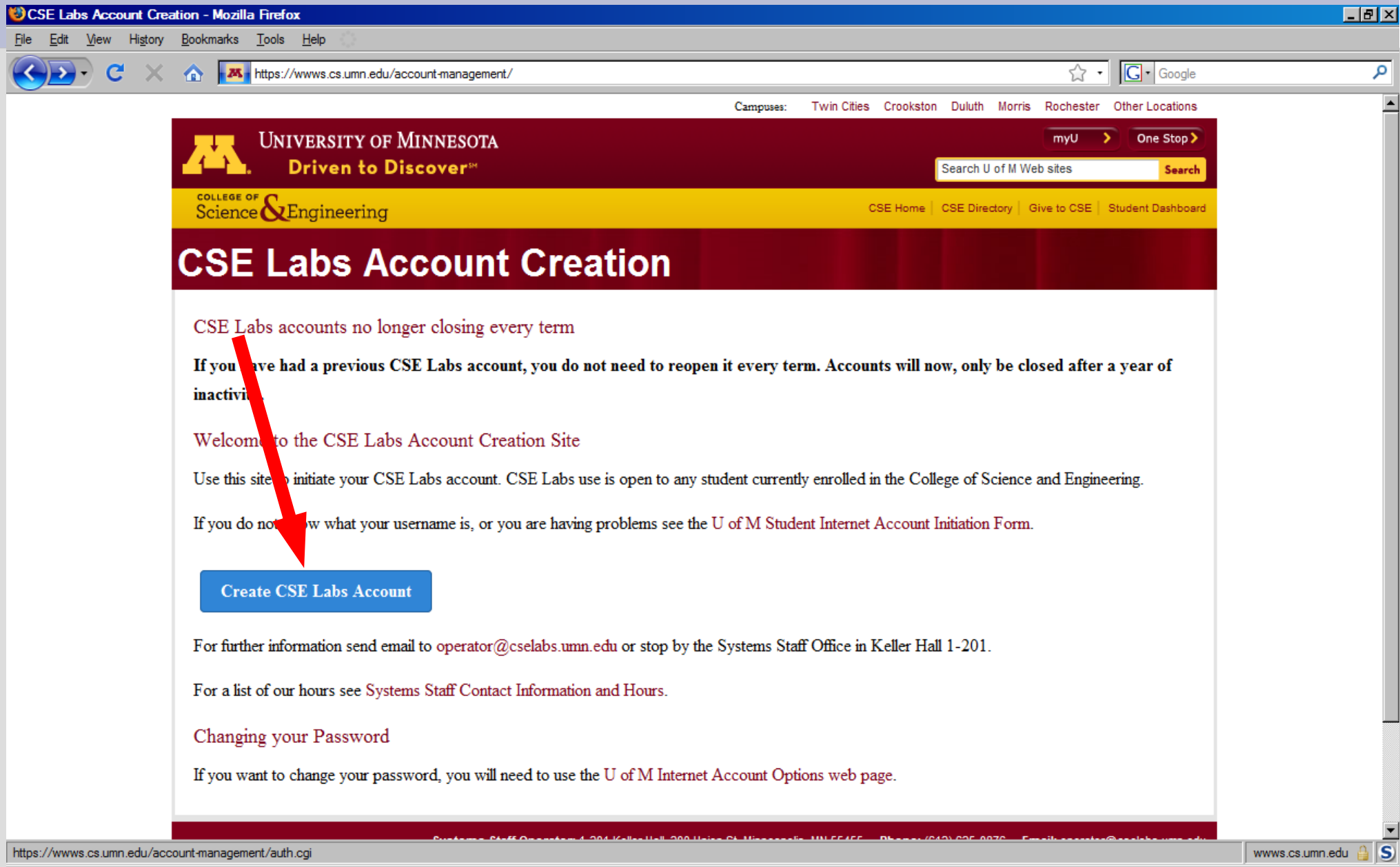
https://cseit.umn.edu/



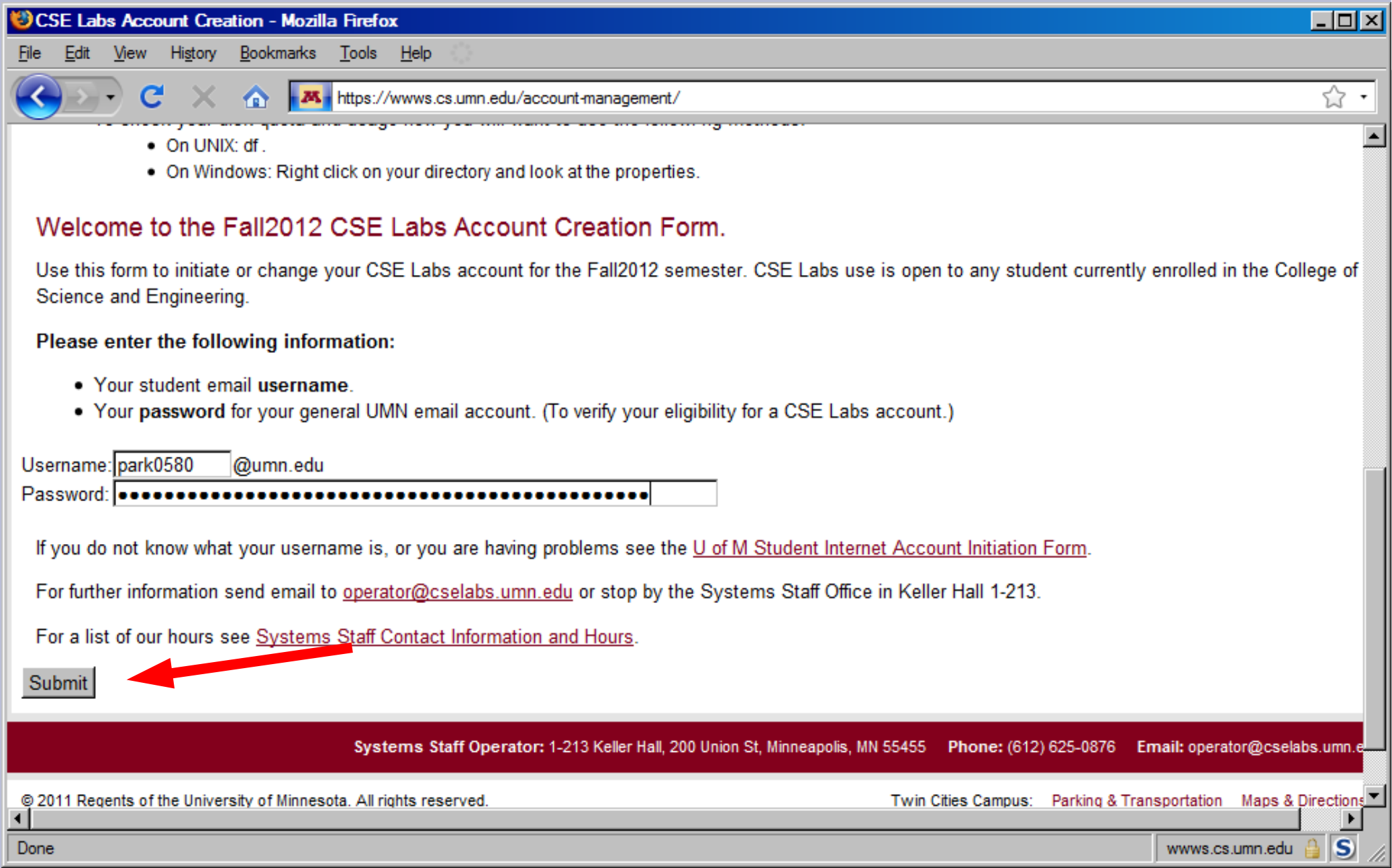
https://cseit.umn.edu/



https://cseit.umn.edu/



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

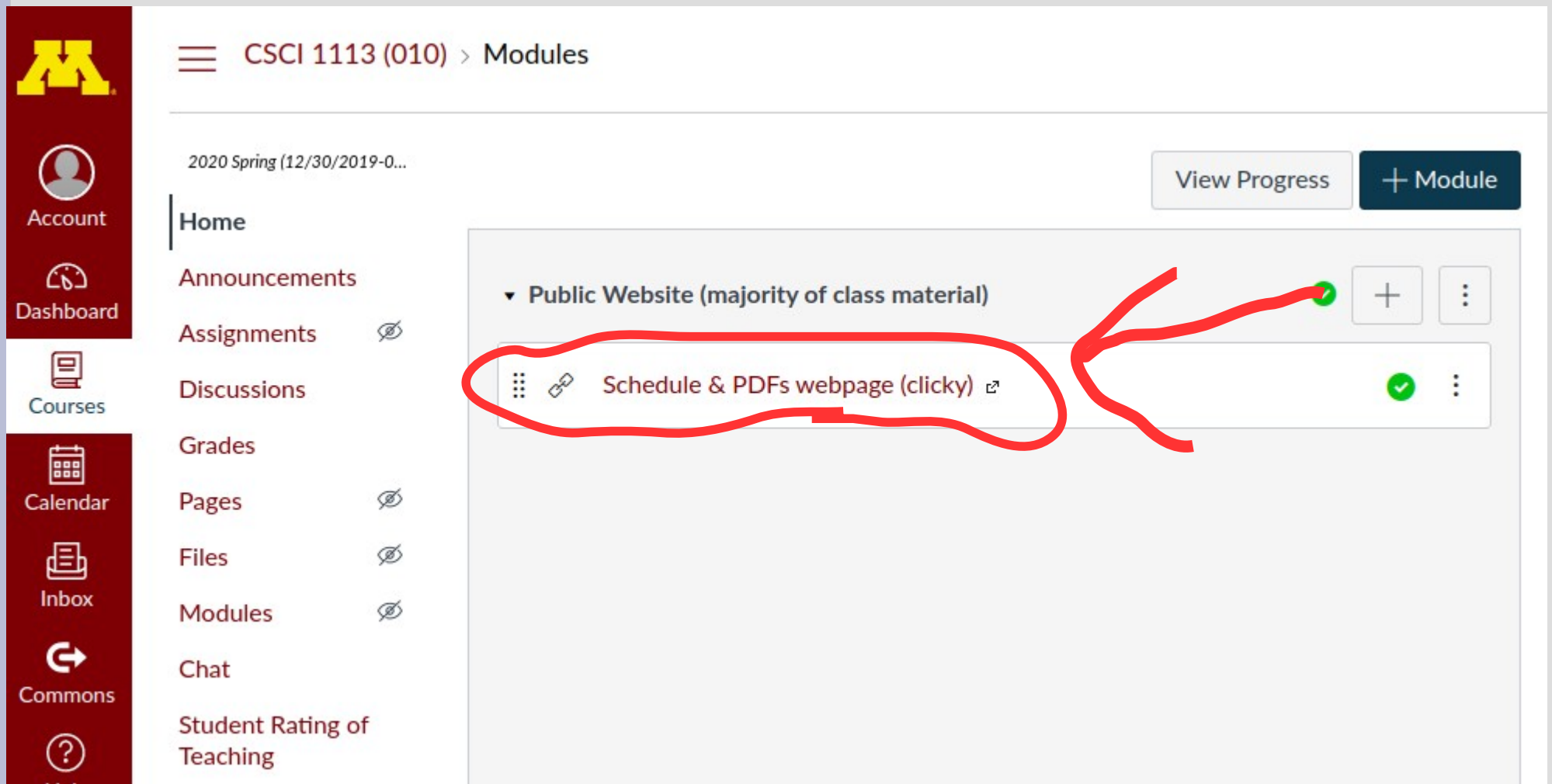
www.cs.umn.edu/academics/classes
Or google “umn.edu csci class”

Syllabus, schedule, other goodies

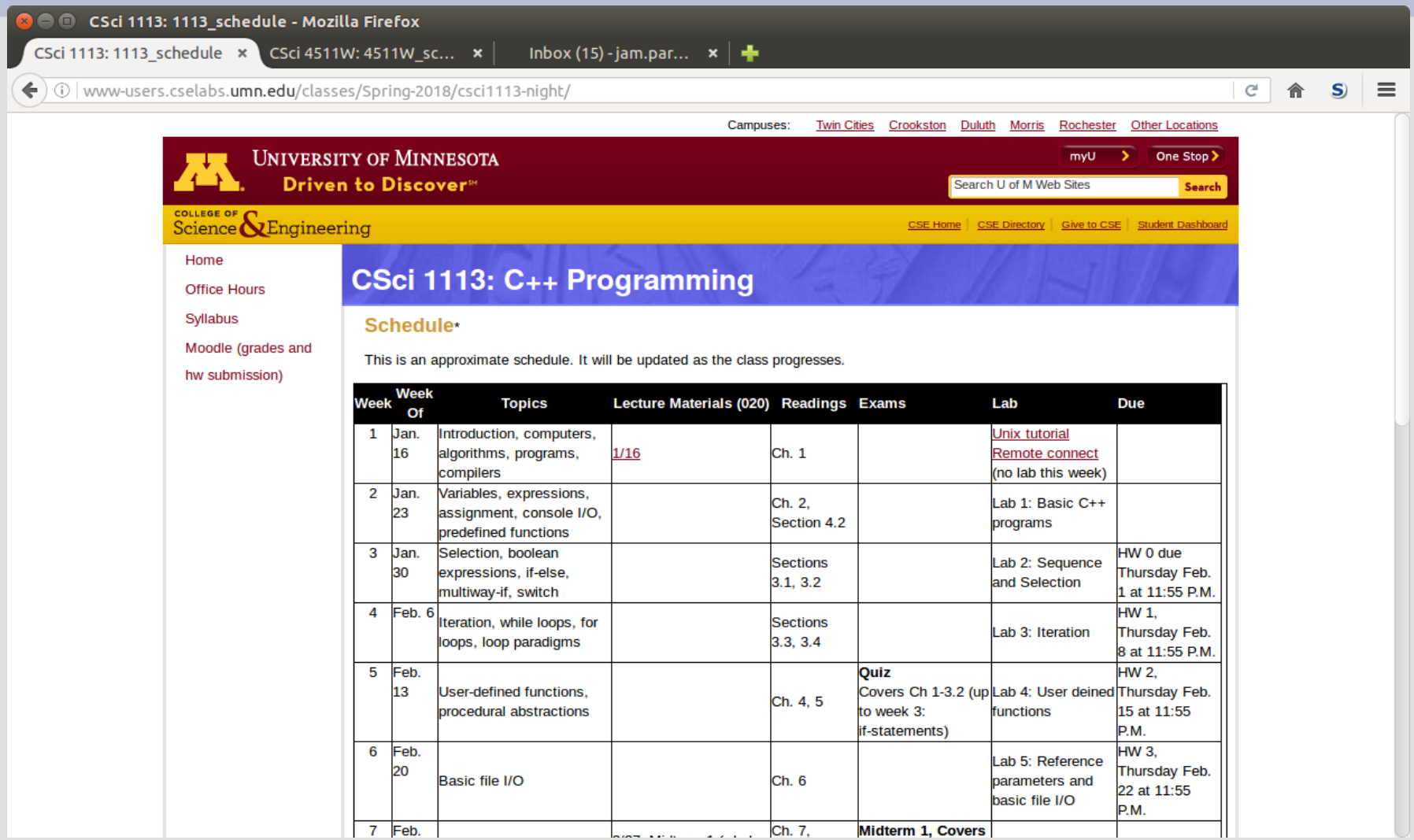
Canvas page will have grades and (maybe) homework submissions

Class website

Canvas also has a link to the website:



www.cs.umn.edu



CSci 1113: 1113_schedule - Mozilla Firefox

CSci 1113: 1113_schedule | CSci 4511W: 4511W_sc... | Inbox (15) - jam.par...

www-users.cselabs.umn.edu/classes/Spring-2018/csci1113-night/

Campuses: Twin Cities Crookston Duluth Morris Rochester Other Locations

UNIVERSITY OF MINNESOTA
Driven to Discover℠

myU One Stop

Search U of M Web Sites Search

COLLEGE OF Science & Engineering

CSE Home | CSE Directory | Give to CSE | Student Dashboard

- Home
- Office Hours
- Syllabus
- Moodle (grades and hw submission)

CSci 1113: C++ Programming

Schedule*

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

Week	Week Of	Topics	Lecture Materials (020)	Readings	Exams	Lab	Due
1	Jan. 16	Introduction, computers, algorithms, programs, compilers	1/16	Ch. 1		Unix tutorial Remote connect (no lab this week)	
2	Jan. 23	Variables, expressions, assignment, console I/O, predefined functions		Ch. 2, Section 4.2		Lab 1: Basic C++ programs	
3	Jan. 30	Selection, boolean expressions, if-else, multiway-if, switch		Sections 3.1, 3.2		Lab 2: Sequence and Selection	HW 0 due Thursday Feb. 1 at 11:55 P.M.
4	Feb. 6	Iteration, while loops, for loops, loop paradigms		Sections 3.3, 3.4		Lab 3: Iteration	HW 1, Thursday Feb. 8 at 11:55 P.M.
5	Feb. 13	User-defined functions, procedural abstractions		Ch. 4, 5	**Quiz** Covers Ch 1-3.2 (up to week 3: if-statements)	Lab 4: User deined functions	HW 2, Thursday Feb. 15 at 11:55 P.M.
6	Feb. 20	Basic file I/O		Ch. 6		Lab 5: Reference parameters and basic file I/O	HW 3, Thursday Feb. 22 at 11:55 P.M.
7	Feb.			Ch. 7,	**Midterm 1, Covers**		

Syllabus

15% Labs
30% Homework (due Fridays)
  5% Quiz (Feb. 18)
10% Midterm 1 (March 3)
15% Midterm 2 (April 14)
25% Final (May 12, 6:30-8:30pm)

Syllabus

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

Homework is due Fridays at 11:55 P.M. (more details 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 questions must be 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)

Homew

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 & Inheritence

Syllabus

Any questions?

What can I program?

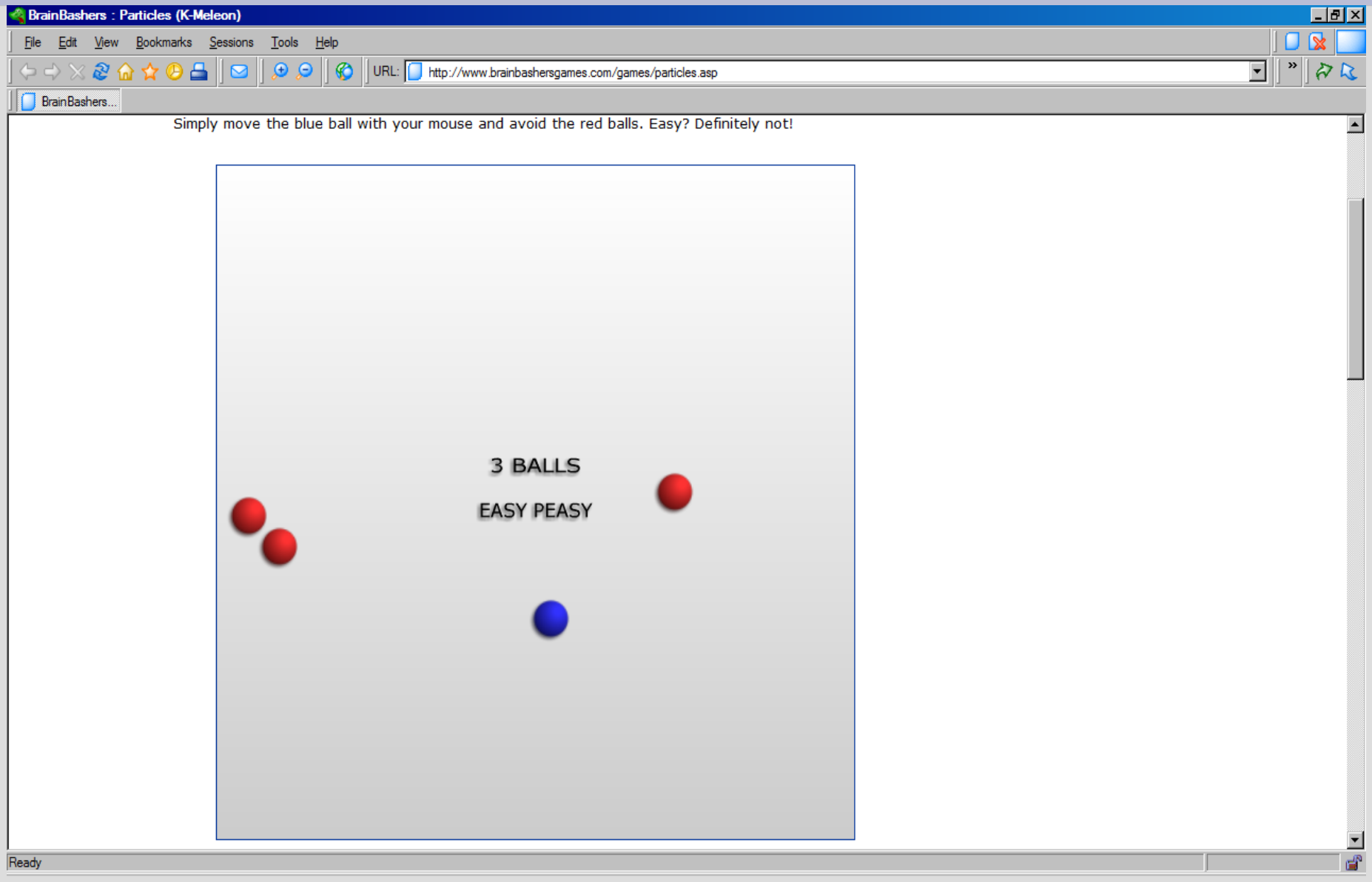
If you can think of an explicit process (of simple steps) to solve your problem, then it can be programed.

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.
7. Bake for 1 hour. Cool on a rack.

Repetitive tasks



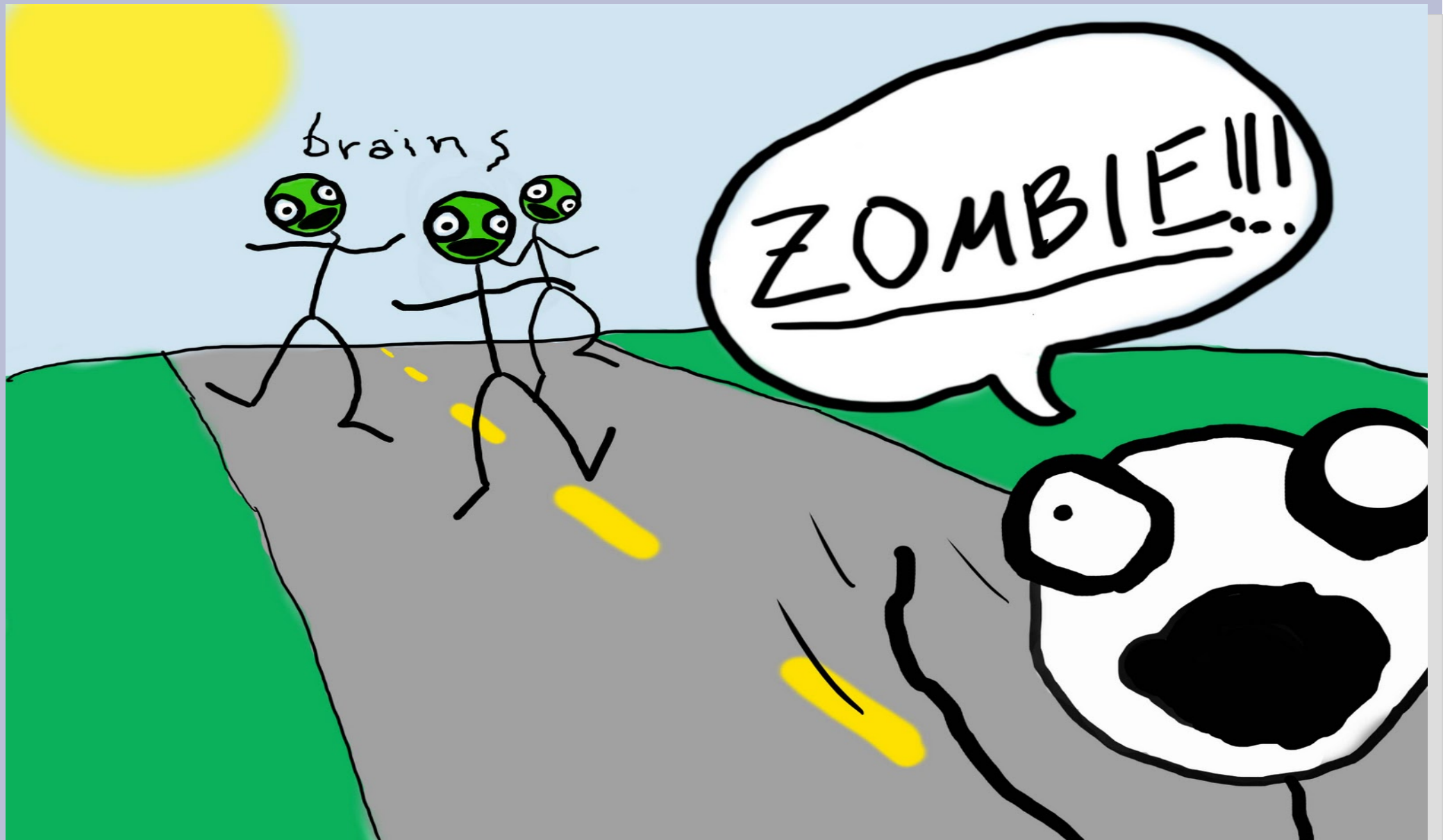
ATMs

How do you get change for $18.26 with the least amount of bills and coins?

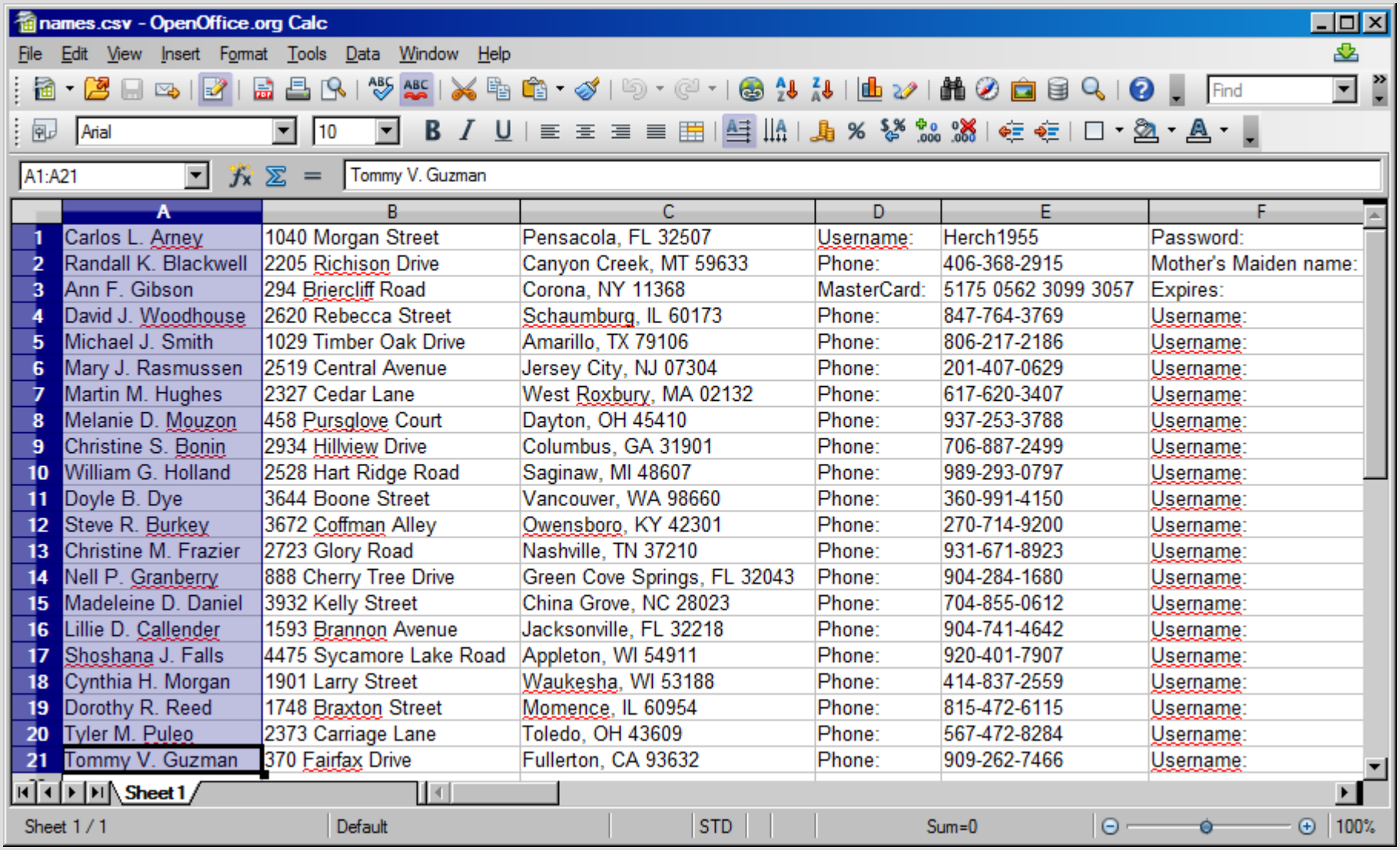
Repetitive tasks

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

Repetitive tasks



Repetitive tasks



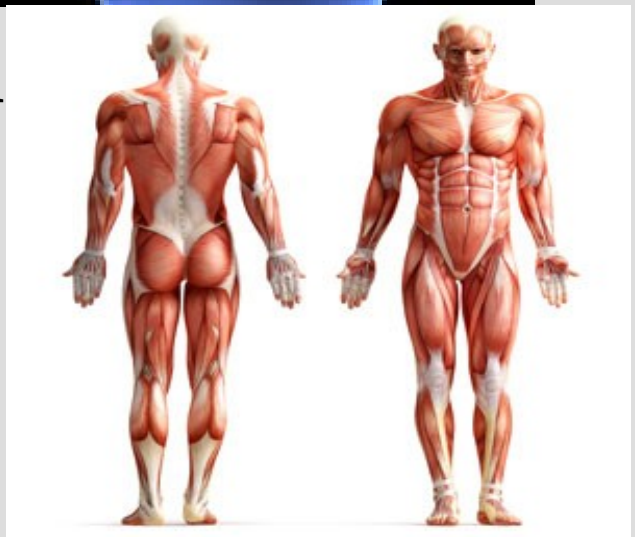
	A	B	C	D	E	F
1	Carlos L. Arney	1040 Morgan Street	Pensacola, FL 32507	Username:	Herch1955	Password:
2	Randall K. Blackwell	2205 Richison Drive	Canyon Creek, MT 59633	Phone:	406-368-2915	Mother's Maiden name:
3	Ann F. Gibson	294 Briercliff Road	Corona, NY 11368	MasterCard:	5175 0562 3099 3057	Expires:
4	David J. Woodhouse	2620 Rebecca Street	Schaumburg, IL 60173	Phone:	847-764-3769	Username:
5	Michael J. Smith	1029 Timber Oak Drive	Amarillo, TX 79106	Phone:	806-217-2186	Username:
6	Mary J. Rasmussen	2519 Central Avenue	Jersey City, NJ 07304	Phone:	201-407-0629	Username:
7	Martin M. Hughes	2327 Cedar Lane	West Roxbury, MA 02132	Phone:	617-620-3407	Username:
8	Melanie D. Mouzon	458 Pursglove Court	Dayton, OH 45410	Phone:	937-253-3788	Username:
9	Christine S. Bonin	2934 Hillview Drive	Columbus, GA 31901	Phone:	706-887-2499	Username:
10	William G. Holland	2528 Hart Ridge Road	Saginaw, MI 48607	Phone:	989-293-0797	Username:
11	Doyle B. Dye	3644 Boone Street	Vancouver, WA 98660	Phone:	360-991-4150	Username:
12	Steve R. Burkey	3672 Coffman Alley	Owensboro, KY 42301	Phone:	270-714-9200	Username:
13	Christine M. Frazier	2723 Glory Road	Nashville, TN 37210	Phone:	931-671-8923	Username:
14	Nell P. Granberry	888 Cherry Tree Drive	Green Cove Springs, FL 32043	Phone:	904-284-1680	Username:
15	Madeleine D. Daniel	3932 Kelly Street	China Grove, NC 28023	Phone:	704-855-0612	Username:
16	Lillie D. Callender	1593 Brannon Avenue	Jacksonville, FL 32218	Phone:	904-741-4642	Username:
17	Shoshana J. Falls	4475 Sycamore Lake Road	Appleton, WI 54911	Phone:	920-401-7907	Username:
18	Cynthia H. Morgan	1901 Larry Street	Waukesha, WI 53188	Phone:	414-837-2559	Username:
19	Dorothy R. Reed	1748 Braxton Street	Momence, IL 60954	Phone:	815-472-6115	Username:
20	Tyler M. Puleo	2373 Carriage Lane	Toledo, OH 43609	Phone:	567-472-8284	Username:
21	Tommy V. Guzman	370 Fairfax Drive	Fullerton, CA 93632	Phone:	909-262-7466	Username:

Auto leveling?

Software vs Hardware

Software - the more intangible code on a computer

Hardware - the physical Parts of the computer



Hardware interaction

CPU

Input

Output

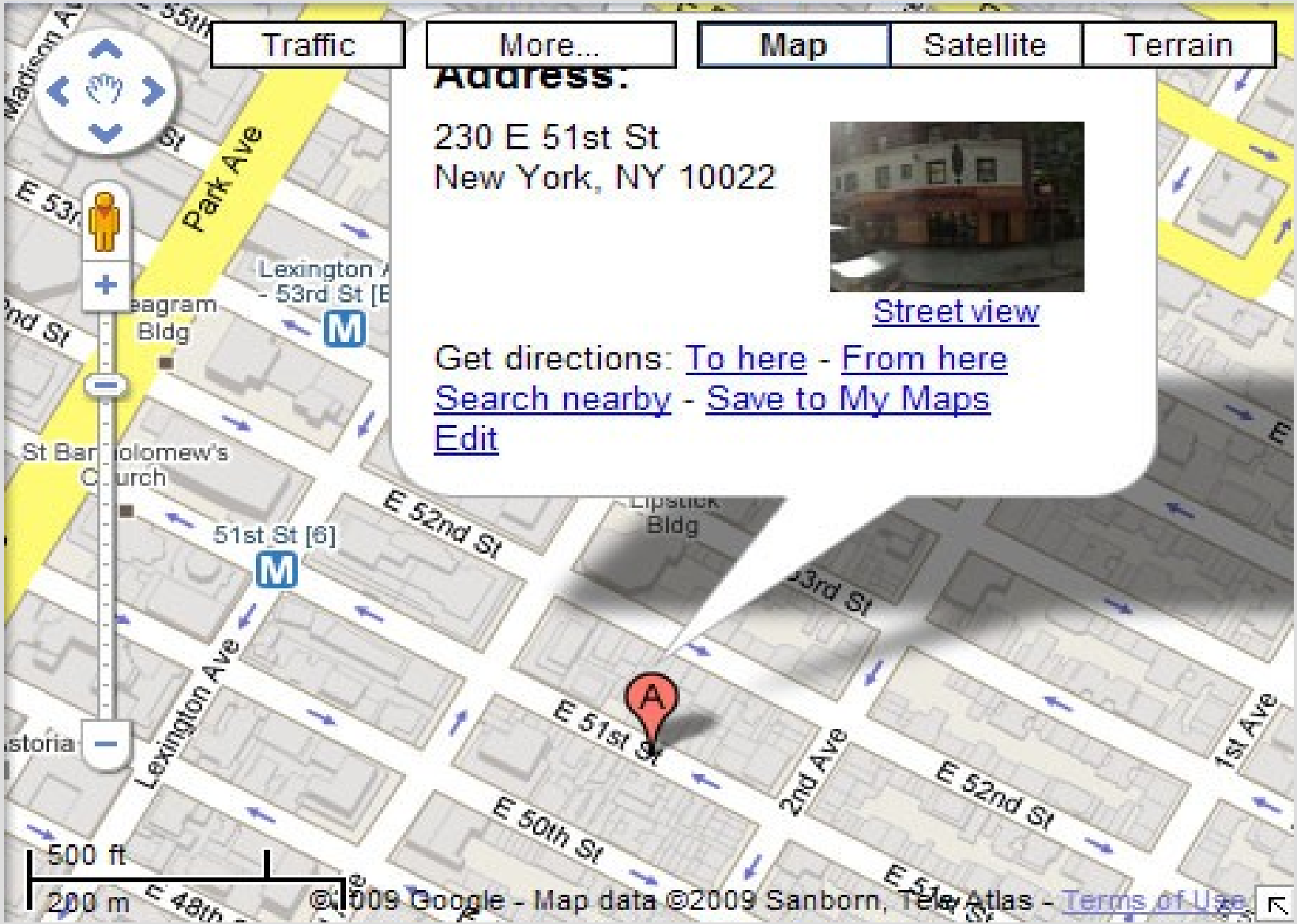
Memory

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...

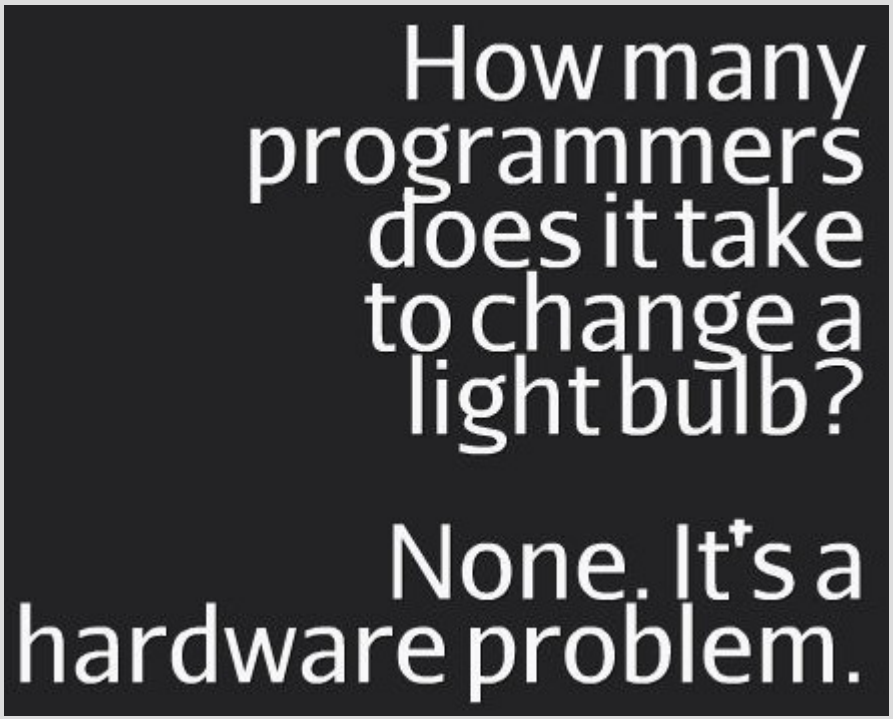
Banana nut bread? Ingredients

ATM? Dollars & coins

Ball game? ~~Balls & mouse~~

Lots of pixels (tiny color dots)

Break time!


How many programmers does it take to change a light bulb?
None. It's a hardware problem.

Object Oriented

Main focus is on objects and how they interact (represented by me as boxes)

Reusable groups of actions (verbs) between objects are called functions (squiggly boxes)

These actions can take additional information called arguments,
(an analogy is ordering at a restaurant; the ordering format is the same, different food)

Object Oriented

One format is:
object.function(argument, argument...);

Example:
James.teaches(CSci 1113);
teach(James, CSci 1113);

The dot (period) shows that “teaching” is an action done by “James”

Banana Nut Bread

Ingredients

* 3 or 4 ripe bananas, smashed
* 1/3 cup melted butter
* 1 cup sugar
* 1 egg, beaten
* 1 teaspoon vanilla
* 1 teaspoon baking soda
* Pinch of salt
* 1 1/2 cups of all-purpose flour
* 1 cup of nuts

Data (Objects)

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.
7. Bake for 1 hour. Cool on a rack.

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.
7. Bake for 1 hour. Cool on a rack.

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.
7. Bake for 1 hour. Cool on a rack.

Banana Nut Bread

Pseudo code directions

1. oven.preheat(350);
2. bowl.mix(butter, bananas);
3. bowl.mix(sugar, egg, vanilla);
4. bowl.sprinkle(baking soda, salt);
5. bowl.mix(flour, nuts);
6. bowl.pour(pan);
7. pan.bake(60);
8. pan.cool();

Banana Nut Bread

Pseudo code directions #2

1. oven.preheat(350);
2. bowl.add(butter, bananas);
3. bowl.mix();
4. bowl.add(sugar, egg, vanilla);
5. bowl.mix();
6. bowl.sprinkle(baking soda, salt);
7. bowl.add(flour, nuts);
8. bowl.mix();
9. pan.pour(bowl);
10. pan.bake(60);
11. pan.cool();

Banana Nut Bread

```
mashedBananas = bananas.mashed();
bowl.add(butter, mashedBananas);
```

same as:

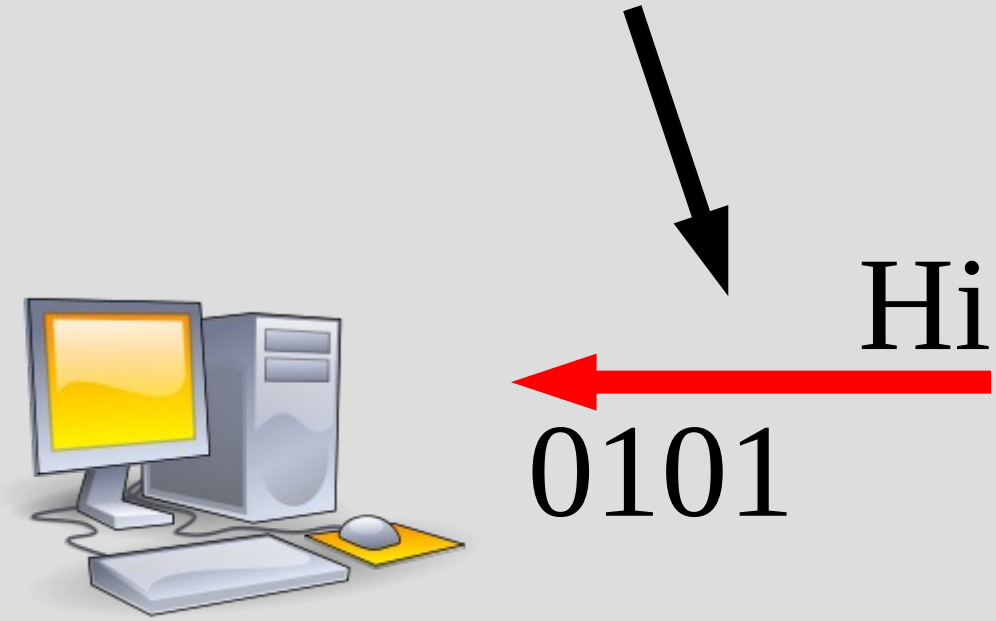
```
bowl.add(butter, bananas.mashed());
```

```
Kitchen.bowl.add(butter, bananas.mashed());
```

```
hand.mix(butter, mashedBananas);
bowl.add(hand.mix(butter, mashedBananas));
```

Compiling

Converting code to binary is called compiling



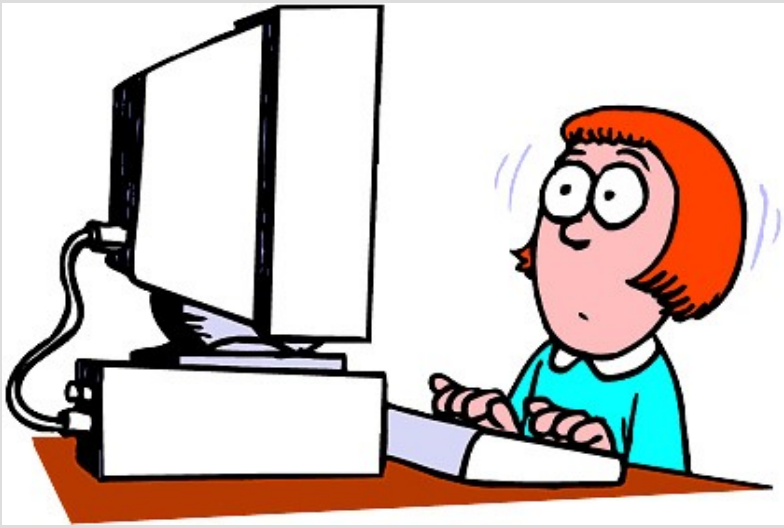
Compiling

Often this compiled code Will not work on other computers

0101

Hi

0101



Compiling


Microsoft
Mac OS
Sie sind nicht berechtigt, unrechtmäßige Kopien dieses Datenträgers zu erstellen.
Sie müssen ein lizenzierter Benutzer von Microsoft Office sein, um die Update-Version installieren zu dürfen.
Microsoft
Office 98
Macintosh Edition
Upgrade
Für Ergebnisse, die so brillant sind wie Ihre Ideen.

Compiling

C++ is a high level language (human readable)

Compiling changes a high level language into a low level language that is easier for the computer (computer cannot run high level)

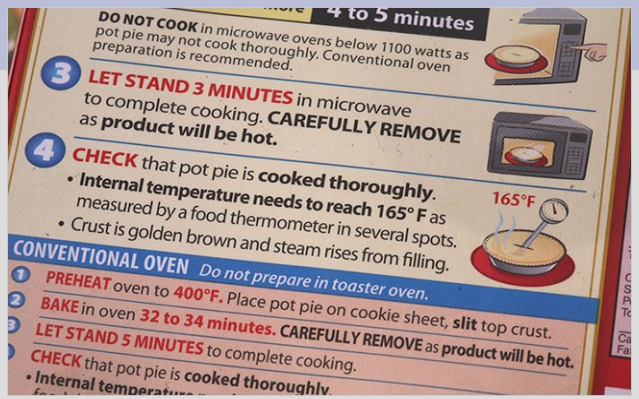
Compiling

Your source code is the original language you wrote your program in (the C++ code for us)

You must recompile the source code every time you save a change before running the program again

Compiling tl;dr

directions



cook

meal

eat

satiated

code

compile

1's and 0's (program)

run

pretty colors

Compiling

In labs, the computers will come with a program called “geany” (which I will use too)

This program is where you can write code and easily compile simple programs

To run it either click the terminal icon () on the left bar or press Ctrl+Alt+T

Then type: geany (enter)

High level (C++)

```
#include <iostream>
using namespace std;

int main ()
{
  cout << "Hello World! ";
  return 0;
}
```

(See: helloWorld.cpp)

Low level (Assembly)

```
MODEL SMALL
IDEAL
STACK 100H

DATASEG
MSG DB 'Hello, World!', 13, '$'

CODESEG
Start:
MOV AX, @data
MOV DS, AX
MOV DX, OFFSET MSG
MOV AH, 09H ; output ascii string
INT 21H
MOV AX, 4C00H
INT 21H
END Start
```

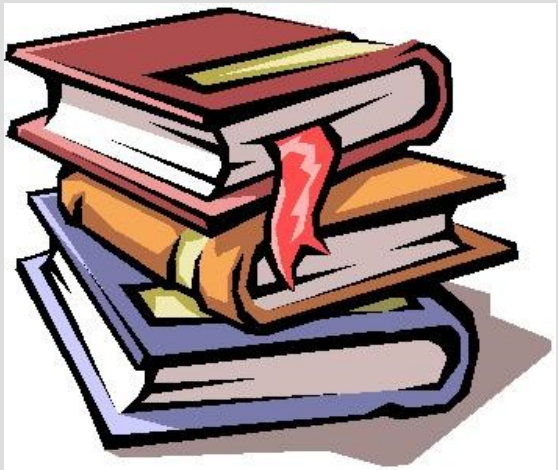
Ease of use

Why C++?

Speed

Control

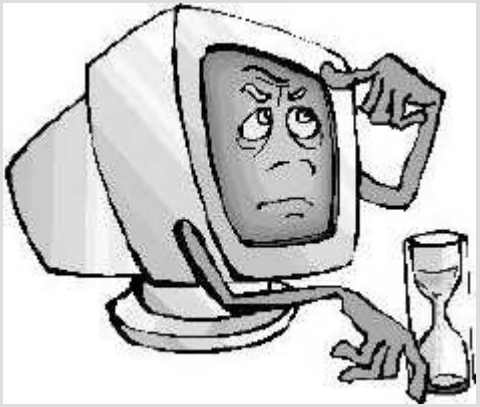
Libraries



Speed

Not all programming languages need to compile code as C++ (Java, Python)

Compiling can greatly increase speed of a program



Control

C++ allows you great control over your data (and its interpretation)

This comes with a burden of responsibility to properly manage your data

If you mismanage your data, you are likely to cause an error in your program

Libraries

C++ is an old language (older than me) and this comes with pros and cons...

Some aspects are quirky to enable backwards compatibility (and are honestly out of date)

Since it has been around for a long time, there are lots of supporting libraries
(and the language continues to develop...)

Java vs C++

Java

Goes anywhere
Comfy

C++

Fast
Fine tuned

Magic 8 ball

Magic 8 ball

What a rip off!

Magic 8 ball

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      cout << "Maybe.";
7
8      return 0;
9  }
```

Keyboard input

cout << “word”

- prints “word” to the screen

cin >> x

- store what is typed into “x”
  (x is some object or data)

Can also do arithmetic using +, -, / and *
(See: inputOutput.cpp)

Types of errors

Syntax error - code will not compile
e.g. cout(“hi”);

Runtime error - code crashes after starting
(see: runtimeError.cpp)

Logic error - code runs but doesn't return the correct answer
(see: logicError.cpp)

Syntax

Syntax is a fancy word for the “grammar” of programming languages

The basic English syntax is:
(subject) (verb) (noun)
“I eat bananas” not “Bananas I eat”

The computer is VERY picky (and stubborn) about grammar, and will not understand you unless you are absolutely correct!

Avoid errors

To remove your program of bugs, you should try to test your program on a wide range of inputs

Typically it is useful to start with a small piece of code that works and build up rather than trying to program everything and then debug for hours

Comments

Comments are ignored pieces of code (computer will pretend they do not exist)

// denotes a single line that is commented
// (everything before hitting enter)

/* denotes the beginning of a comment and the end of a comment is denoted by */

Additional facts

Braces denote a block of code { }
(belonging to a method, class, etc.)

“White space” is ignored, just as the your brain will ignore the bottom third of this slide (this is why we need a semi-colon)