Welcome to

"Solving Problems with Computers I"

CS 16: Solving Problems with Computers I Lecture #1

Ziad Matni Dept. of Computer Science, UCSB

Image from agorolabs on slideshare.com

Terr Grow Doctor

A Word About Registration for CS16

FOR THOSE OF YOU NOT YET REGISTERED:

This class is currently FULL and the waitlist is CLOSED

Your Instructor

Your instructor: Ziad Matni

(zee-ahd mat-knee)

Email: *zmatni@cs.ucsb.edu*

(please put **CS16** at the start of the subject header – I teach 2 other classes!!!)

My office hours: Mondays 11:00 AM – 12:00 PM, at SSMS 4409

(or by appointment)

Your TAs

All labs will take place in PHELPS 3525

All TA office hours will take place in "Open Lab" Time in **PHELPS 3525**

LAB TAs

Shiyu Ji

Muqsit Nawaz

Xiyou Zhou

GRADER

Ashwini Patil

4/3/18



4/3/18

Matni, CS16, Sp18

You!

With a show of hands, tell me... how many of you...

- A. Are Freshmen? Sophomores? Juniors? Seniors?
- B. Are CS majors? Other?
- C. Have programmed before? What language?
- D. Have programmed before "just for fun"?
- E. Have programmed before "for work or school"?
- F. Have used a Linux or UNIX system before?

This Class

- An intermediate (not a beginner's) class in computer science
 - You WILL need to have taken a beginner's class somewhere
- Covers the basic building blocks for solving problems using computers, in general, and using C++ programming specifically
- Enables you to go on to take other exciting classes in programming!!!!!
 OMG!!!

Why Are We Using C++ in this Course?

- C++ is one of the most widely used and in-demand computer programming languages
 - For a list of commercial applications written in C++, see http://www.stroustrup.com/applications.html
- If you can learn C++, you can more easily learn (or even teach yourself) other popular P.L.s
 - Like Python, Java, PHP, Ruby, etc...
- It looks great on your resume!
 - Actually, it's a must-have on any "decent" CS major's resume...

How Is This Class Taught?

- Every class has a lecture based on the readings: ٠ YOU MUST DO THE READINGS BEFORE CLASS!!!
- You will be in a lab on Mondays: YOU MUST READ YOUR LAB ASSIGNMENT BEFORE YOU GO TO LAB!!!
- You have to do a lot of homework and lab assignments



BECAUSE PRACTICE MAKES PERFECT!!! (and also, it's actually fun)



- 15-16 (short) Homeworks 1x or 2x a week, all solo
- 2 Midterm Exams
- 1 Final Exam ullet

8-9 Lab Assignments
 1x a week, some solo/some paired

... and a partridge in a pear tree...

Why so much work??

Because Programming is a Skill...

Online Resources

Class MAIN webpage – ALL THE INFO! ALL THE TIME!

https://ucsb-cs16-s18-matni.github.io

Piazza discussions/Q&A AND IMPORTANT ANNOUNCEMENTS!:

https://piazza.com/ucsb/spring2018/cs16

GauchoSpace – ONLY FOR HOMEWORK ASSIGNMENTS!

https://gauchospace.ucsb.edu/courses/course/view.php?id=24038

If You Have Questions... Ask! But First, Check...



If You Have Questions... Ask! But First, Check...



IT'S IN THE SYLLABUS

This message brought to you by every instructor that ever lived. WWW.PHDCOMICS.COM

So... let's take a look at that syllabus... Full electronic version found at: <u>http://cs.ucsb.edu/~zmatni/syllabi/CS16S18_syllabus.pdf</u> <u>PLEASE READ IT!!!!</u>

Required textbook: Problem Solving with C++ (10th Edition) by Walter Savitch (9th ed. is ok, but references may be different)

Lectures: Uses the readings, but also adds it own components. All assignments are based on what is said in lecture. Slides are placed on website afterwards.



Homework: 1-2x a week. Announced on the website, done on GauchoSpace! Late policy: After due date, you get 20% off for "late" (< 24 hrs) or ZERO if after 24 hrs.

So... let's take a look at that syllabus... Full electronic version found at: http://cs.ucsb.edu/~zmatni/syllabi/CS16S18_syllabus.pdf PLEASE READ IT!!!!

Labs: Once a week. Description sheet is placed on website on weekend. Lab is on **Wednesday**. YOU MUST USE submit.cs TO TURN IN ALL LABS. PLEASE FOLLOW INSTRUCTIONS ON SHEET! Must be turned in before **Monday** AT NOON.

Late policy: After due date, you get 20% off for "late" (< 24 hrs) or ZERO if after 24 hrs.

Most labs are solo-work, but some must be done in pairs (i.e. groups of 2 ppl).

Exams: 2 Midterms + 1 Final (cumulative). No re-dos, no early takes, no late takes (except for extreme cases), no make-ups.

Laptop Policy: Can only use it to take notes, do class-related coding work (no social media!) Cell Phone Policy: Put that thing back where it came from or so help me...

4/3/18

Switching About In The Labs...

... is frowned upon 😁

Please stick to the lab time that you have per your registration

 The labs are pretty full and at capacity

IF YOU WANT TO SWITCH LAB SECTIONS, YOU MUST:

 Find a person in the other lab to switch with you
 2. Get the OK from <u>BOTH</u> T.A.s This is because the labs have limited space

Matni, CS16, Sp18

4/3/18

Should I Buy/Download a C++ Programming Environment/Suite??

- There are a few GUI-based IDEs out there for C++
 - You are <u>NOT REQUIRED</u> to get them for this class!!!!! (so, no)
- Regardless, you <u>ARE REQUIRED</u> to become proficient on our UNIX/Linux CSIL machines' C++ compiler and environment

– "g++" and "makefile", etc...

• Real programmers use command-lines...!

What YOU have to do **before THURSDAY'S LECTURE?**

- Log into Piazza and have a look around
- Go to the class main website and have a look around
 - Go to: <u>https://ucsb-cs16-s18-matni.github.io/</u>
- Read the lab assignment (lab01) before you go into your lab:

BE PREPARED

Anything Else???

YES! YOU HAVE ANOTHER LECTURE ON THURSDAY!!!

- Do the required reading!!! (Chapter 1 and start Chapter 2)
- WAAAY before you go to class, DO HOMEWORK #1
 - 1. Go to the class' GauchoSpace
 - 2. Find Homework #1 module
 - 3. Do it
 - 4. Submit it
 - 5. Profit?

A Refresher on Computers



Computer Systems

• Hardware

- The physical
 - CPU and Memory ICs
 - Printed circuit boards
 - Plastic housing, cables, etc...

• Software

- The instructions and the data
 - Programs and applications
 - Operating systems



5 Main Components to Computers

- 1. Processor
- 2. Main memory
 - Usually inside the computer, volatile
- 3. Inputs
- 4. Outputs
- 5. Secondary memory
 - More permanent (non-volatile) memory for mass storage of data

Computer Memory

- Usually organized in two parts:
 - Address
 - Where can I find my data?
 - Data (payload)
 - What is my data?
- The smallest representation of the data
 - A binary bit ("0"s and "1"s)
 - A common collection of bits is a byte (8 bits = 1 byte)
 - What information can one store in 1 byte?

A number? A letter? A program? A book? A picture? A movie?

<u>Scales of a Byte</u>				
1B	(byte)			
1 kB	(kilobyte)			
1 MB	(megabyte)			
1 GB	(gigabyte)			
1 TB	(terabyte)			
???				

What is the Most Basic Form of Computer Language?

- Binary *a.k.a* Base-2
- Expressing data AND instructions in either "1" or "0"
 - So,

 $"01010101\ 01000011\ 01010011\ 01000010\ 00100001\ 00100001"$

could mean an *instruction* to "calculate 2 + 3"

Or it could mean a *number* (856,783,663,333)

Or it could mean a *string of 6 characters* ("UCSB!!")

Computer Software

- All the data
- All the programs
- All the applications
- The operating system(s)
- What is firmware?

The Operating System

- Is it a program?
 - In a general sense, yes!
 (or more precisely, a bunch of programs acting in concert)
- What does it do?
 - Allocates the computer's resources like memory
 - Allows us to communicate with the computer via I/O
 - Responds to user requests to run other programs

Algorithm vs. Program

Complete this: "Computer Science is about studying how to

use ______ to solve problems"

Algorithms

Sequences of precise instructions that leads to a solution

• Programs

- Algorithms expressed in a language the computer can understand

High-Level Computer Languages

- A computer language that closely mimics "natural language"
 - As opposed to just being 0s and 1s (that's "machine language")
- *High-level languages* provide *high abstraction* to the CPU Instructions
 - Your programs very much look like *algorithms*
- A program that "translates" a High Level Language into Low Level Language (like machine language) is called a compiler
 - Why are compilers necessary???
 - Because CPUs ONLY understand their instructions in Machine Language

Compilers

- Language-specific
 - Compiler for Python will not work for C++, etc...
- Linux/UNIX OS have different built-in compilers
 - e.g. g++ for C++, clang for C, etc...
- Source code
 - The original program in a high level language (text file)
- Object code
 - The translated version in machine language (binary file)

4/3/18

Introduction to the C++ Language

When was it invented?

In the 1980s...

Was it based on anything else? The C Language

Is it still popular?

Yes! In the top-3 of most used today with Python and Java

A Sample C++ Program

A common and simple C++ program begins this way:

#include <iostream>
using namespace std;
int main()
{

// This is a comment (optional)

And ends this way

```
return 0;
}
```

```
#include <iostream>
 1
    using namespace std;
 2
    int main()
 3
 4
    {
 5
         int number_of_pods, peas_per_pod, total_peas;
         cout << "Press return after entering a number.\n";</pre>
 6
 7
         cout << "Enter the number of pods:\n";</pre>
         cin >> number_of_pods;
 8
 9
         cout << "Enter the number of peas in a pod:\n";</pre>
10
         cin >> peas_per_pod;
         total_peas = number_of_pods * peas_per_pod;
11
         cout << "If you have ";</pre>
12
13
         cout << number_of_pods;</pre>
         cout << " pea pods\n";</pre>
14
15
         cout << "and ":
16
         cout << peas_per_pod;</pre>
         cout << " peas in each pod, then\n";</pre>
17
         cout << "you have ";</pre>
18
         cout << total_peas;</pre>
19
         cout << " peas in all the pods.\n";</pre>
20
21
         return 0;
22
    }
```

4/3/18

33

```
#include <iostream>
 1
 2
    using namespace std;
    int main()
 3
 4
        >int number_of_pods, peas_per_pod, total_peas;
 5
 6
     \leftarrow cout << "Press return after entering a number.\n";
 7
     cout << "Enter the number of pods:\n";</pre>
      cin >> number_of_pods;
 8
       Cout << "Enter the number of peas in a pod:\n";</pre>
 9
10
      cin >> peas_per_pod;
11
      tabbed
         total_peas = number_of_pods * peas_per_pod;
         cout << "If you have ";</pre>
12
13
         cout << number_of_pods;</pre>
14
         cout << " pea pods\n";</pre>
      use
15
         cout << "and ";</pre>
      cout << peas_per_pod;
cout << " peas_in cont</pre>
16
17
         cout << " peas in each pod, then\n";</pre>
      cout << "you have ";
18
19
         cout << total_peas;</pre>
20
         cout << " peas in all the pods.\n";</pre>
21
         return 0;
22
```

Press return after entering a number. Enter the number of pods: 10 Enter the number of peas in a pod: 9 If you have 10 pea pods and 9 peas in each pod, then you have 90 peas in all the pods.

1-4: 5:	Program start Variable declaration	
6-20:	Statements	
21-22:	Program end	

cout << "some string or another" ;
//output stream statement</pre>

cin >> some_variable;
//input stream statement

cout and cin are **objects** defined in the **library** iostream

4/3/18

1 2 3	<pre>#include <iostream> using namespace std; int main()</iostream></pre>	Mind the	Svntax!!
4	{		.n a pod:
5	<pre>int number_of_pod</pre>	ls, peas_per_pod, total_peas;	9
6 7 8 9	← cout << "Press re ← cout << "Enter th ← cin >> number_of_	eturn after entering a number.\n"; ne number of pods:\n"; pods; ne number of peas in a pod:\n":	If you have 10 pea pods and 9 peas in each pod, then you have 90 peas in all the pods.
10 11	cin >> peas_per_p	pod; per_of_pods * peas_per_pod;	1-4: Program start
12 13	of tapped cout << "If you h cout << number_of	nave "; ⁻ _pods;	6-20: Statements 21-22: Program end
15 16 17	cout << "peu pou rout << "and "; cout << peas_per_ cout << " peas in	_pod; each pod, then\n";	<pre>cout << "some string or another" ; //output stream statement</pre>
18 19 20	cout << "you have cout << total_pea cout << " peas in	e "; as; a all the pods.\n";	cin >> some_variable; //input stream statement
21 22	return 0; }		cout and cin are objects defined in the library iostream
	4/3/18	Matni, CS16, Sp18	35

YOUR TO-DOs

- Sign up on Piazza
- Go to the class website
- Go to the class GauchoSpace
- Read Lab1 TODAY and do it TOMORROW (Wed.)
 - It's due to be turned in by Monday at NOON!
- Do HW1 and submit it BEFORE the start of Thursday lecture
- Solve world hunger
- Reverse global warming

