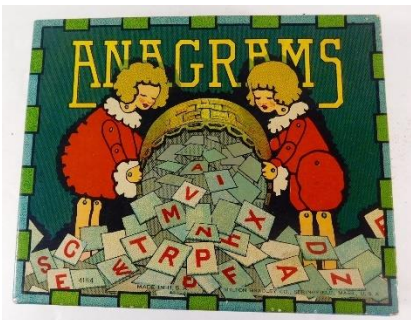


WELCOME TO CS 16!

Problem Solving with Computers-I

<https://ucsb-cs16-sp17.github.io/>



C++

```

#include <iostream>
using namespace std;

int main(){
    cout<<"Hola Facebook!";
    return 0;
}
  
```



Instructor

- Diba Mirza (dimirza@cs.ucsb.edu)
 - PhD (Computer Engineering, UCSD)
 - New teaching faculty at the department of Computer Science, UCSB!
 - Before this: Teaching faculty at UCSD for three years
- Office: HFH 1155
- Office hours (starting on Friday 04/07):
 - Tues: 10am-11am, Friday: 9am -10am
 - Or by appointment

Our teaching staff and brand new tutor program !



Angela Yung
(UG tutor)



Barbara Korycki
(UG tutor)



Jimmy Le
(UG tutor)



Sayali Kakade
(UG tutor)



Sean Shelton
(UG tutor)



Steven Fields
(UG tutor)

Andrew Huang
(UG tutor)

Bryanna Pham
(UG tutor)

Natasha Lee
(UG tutor)

Sherry Li
(UG tutor)

Shreyas
Radhakrishnan
(UG tutor)

Thien Hoang
(UG tutor)

About this course

Why C++?



We will learn:

- A new programming language: C++
- Computer hardware from a programmer's perspective
- Abstractions used in programming and problem solving
- Tools and practices used by professional programmers
- Big ideas that have shaped computing

Solving problems with computers...like a pro!

Clickers out – frequency AB

About you...

What is your major?

- A. Computer Science
- B. Computer Engineering
- C. Other

About you...

What is your past programming experience?

- A. Have never programmed.
- B. Have programmed before “just for fun”
- C. Have taken an introductory CS course
- D. I have a lot of programming experience

About you...

What is your familiarity/confidence with programming in C++?

- A. Know nothing or almost nothing about it.
- B. Used it a little, beginner level.
- C. Some expertise, lots of gaps though.
- D. Lots of expertise, a few gaps.
- E. Know too much; I have no life.

About you...

What is your familiarity/confidence with using version control with Subversion, Git or any other VCS?

- A. Know nothing or almost nothing about it.
- B. Used it a little, beginner level.
- C. Some expertise, lots of gaps though.
- D. Lots of expertise, a few gaps.
- E. Know too much; I have no life.

Have you been in a class that used peer instruction before?

- A. Yes
- B. No
- C. I'm not sure

Clickers, Peer Instruction, and PI Groups

- Find 1-2 students sitting near you. If you don't have any move.
- Introduce yourself.
- This is your initial PI group (at least for today)

iClickers: You must bring them

- Buy an iClicker at the Bookstore
- Register it on GauchoSpace by Friday (01/13)
- Bring your iclicker to class AND section

Assigned Reading from

- Problem Solving with C++, Walter Savitch, Edition 9

You must **attend** class and lab sections

You must **prepare** for class

You must **participate** in class

Course Logistics

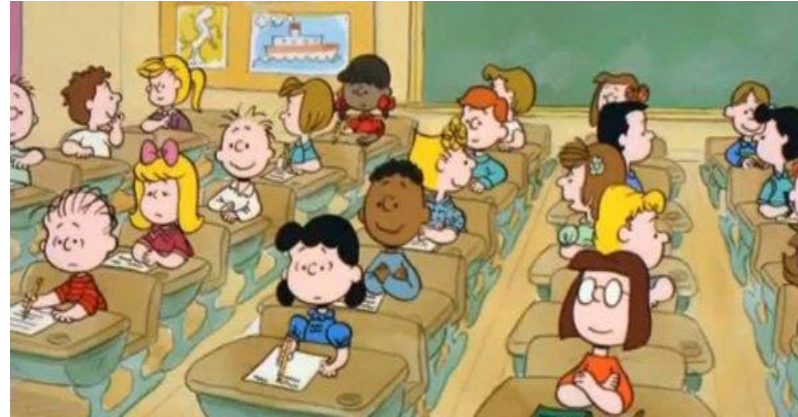
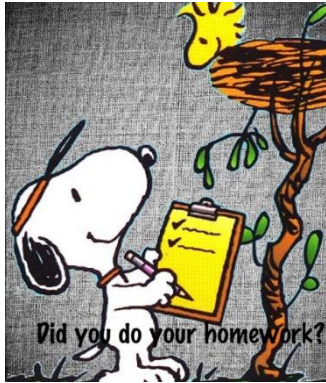
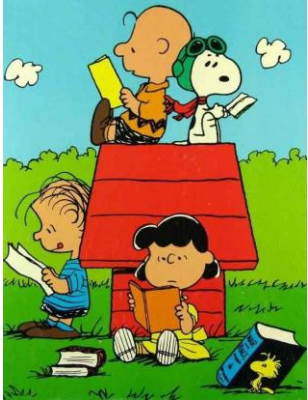
- Grading
 - Class and section participation (iclickers): : 2%
 - Homeworks (due every lecture) : 13%
 - Lab (programming) Assignments(due weekly on Fridays) : 35%
 - Midterm exams: (two, 15% each) : 30%
 - Final exam : 20%
- Less than 75% iClicker response \equiv missing a class/section
- No makeups for exams. Make sure you have no scheduling conflicts with exams
- No LATE submissions unless you have a real emergency!

Course website!

<https://ucsb-cs16-sp17.github.io/>

- * ATTENDENCE in sections and lecture is REQUIRED!
- * To complete the labs you need a college of engineering account. Send me an email before tomorrow's section if you don't have an account

Structure of the class



In class, learn by:

- Taking notes
- Discussing with your peers to identify gaps in your knowledge
- Exercise your meta-cognitive skills!

Before class:

Do the readings and homework!

In class:

- Submit your homework
- Follow class policy on the use of electronic devices



In sections, learn by

- Doing the programming assignments
- Working closely with your pair partner
- Brainstorming with our TAs and tutors!

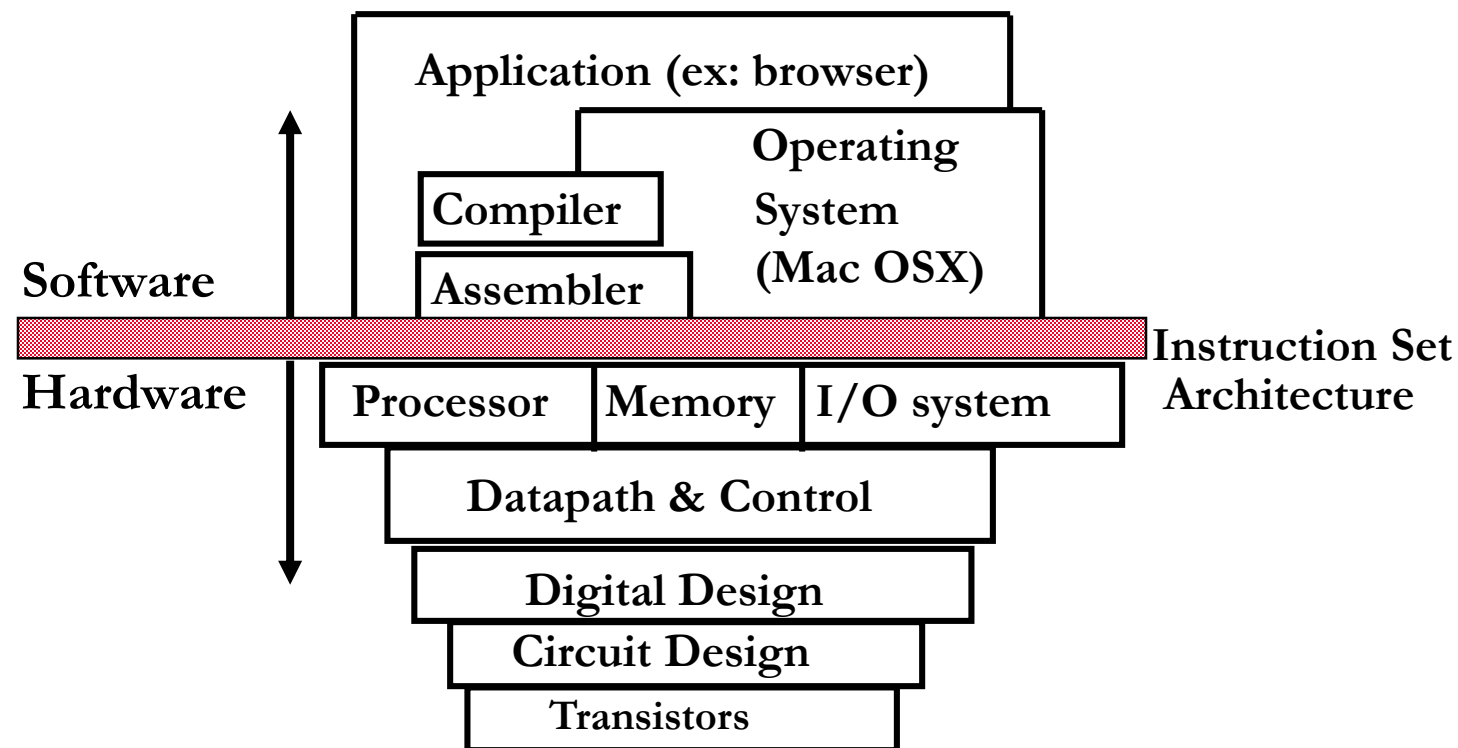
Assignment Calendar

Week	S	M	T	W	R	F	S
1	04/02	04/03 h01 assigned h02 assigned lect01 : Course overview, a gentle intro to C++ - Standard I/O, variables, if-else control structure First day of classes	04/04 lab00 assigned	04/05 lect02 : Evaluating C++ expressions, simple flow control- for, while loops, nested and multi-way if-else	04/06	04/07	04/08
2	04/09	04/10 h01 due 02:00pm h02 due 02:00pm h03 assigned h04 assigned lect03 : Nested loops, git, intro to lab01	04/11 lab00 due 11:59am lab01 assigned	04/12 lect04 : C++ functions and function call mechanics, passing parameters to programs	04/13	04/14	04/15

- For more information, see our Assignment Calendar: <https://ucsb-cs16-sp17.github.io/info/calendar/>
- All sections will be in PHELPS 3252
- Open labs: CSIL in Harold Frank Hall
- The schedule for sections, office hours and open lab hours is available on our class Google Calendar: <https://ucsb-cs16-sp17.github.io/info/schedule/>

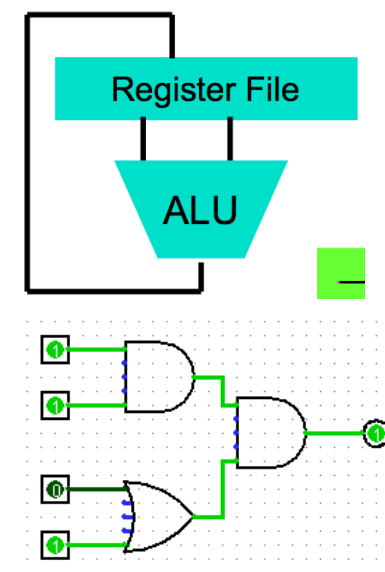
Basic components of a computer

How do we handle complexity?



```
temp = v[k];  
v[k] = v[k+1];  
v[k+1] = temp;  
ldr  r0, [r2]  
ldr  r1, [r2, #4]  
str  r1, [r2]  
str  r0, [r2, #4]
```

```
0000 1001 1100 0110 1010 1111 0101 1000  
1010 1111 0101 1000 0000 1001 1100 0110  
1100 0110 1010 1111 0101 1000 0000 1001  
0101 1000 0000 1001 1100 0110 1010 1111
```



- Big idea: Coordination of many *levels of abstraction*

Lab 00: Must be done individually

Key learning goals:

- Connect remotely to the CSIL unix servers (csil-0X.cs.ucsb.edu)
- Get familiarized with basic UNIX commands
- Create your first C++ program, compile and run it
- Get started with github
- Let us know if you don't have a CoE account before coming into section

LIVE DEMO

Which code produces a compile-time error?

A.

```
int main(){  
    cout<<"Enter two numbers:";  
    cin>>a >> b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
    return 0;  
}
```

B.

```
int main(){  
    int a, b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
    return 0;  
}
```

C.

Both **A** and **B**

D.

Neither **A** or **B**

C++ Variables and Datatypes

- **Variables** are containers to store data
- **C++** variables must be “declared” before they are used by specifying a datatype
 - `int`: Integers
 - `double`: floating point numbers
 - `char`: characters

```
int main() {  
  
    cout<<"Enter two numbers:";  
    cin>>a >> b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
}
```

Will the above code work?

C++ Uninitialized Variables

- Value of uninitialized variables is “undefined”
- Undefined means “anything goes”
- Can be a source of tricky bugs
- What is the output of the code below?

```
int main() {  
    int a, b;  
    cout<<"The sum of "<< a << " and " << b<< " is:"<< a+b<<endl;  
}
```

Variable Assignment

- The values of variables can be initialized...

```
int myVariable = 0;
```

-or-

```
int myVariable;  
myVariable = 0;
```

Variable Assignment

- ...or changed on the fly...

```
int myVariable = 0;  
myVariable = 5 + 2;
```


Variable Assignment

- ...or even be used to update the same variable!

```
int myVariable = 0;  
myVariable = 5 + 2;  
myVariable = 10 - myVariable;  
myVariable = myVariable==0;
```

Variable Assignment

- ...or even be used to update the same variable!

```
int myVariable = 0;  
myVariable = 5 + 2;  
myVariable = 10 - myVariable;  
myVariable = myVariable==0;
```

Control flow: if statement

- The `condition` is a **Boolean expression**
- These can use relational operators

```
if ( 1 < 2 ) {  
    cout<< "foo" ;  
}
```

```
if ( 2 == 3 ) {  
    cout<<"foo" ;  
}
```

```
if ( Boolean expression) {  
    // statement 1;  
    // statement 2;  
}
```

Next time

- Evaluating C++ expressions
- simple flow control- for, while loops, nested and multi-way if-else