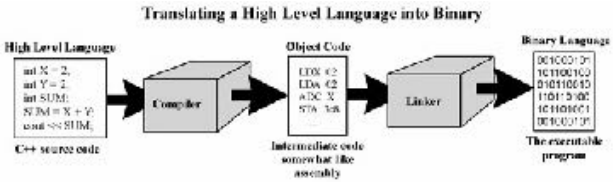
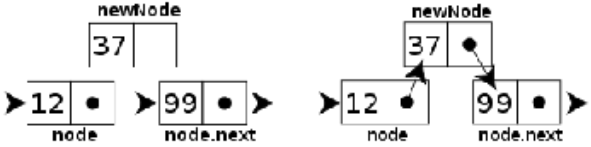



GitHub

# WELCOME TO CS 16!

Problem Solving with Computers-I

**https://ucsb-cs16-s18-mirza.github.io/**


Enrollment status: 117/105


# About me

- Diba Mirza (diba@ucsb.edu)
  - PhD (Computer Engineering, UCSD)
  - First year as faculty at UCSB!
  - Before this: Teaching faculty at UCSD for three years
- Office hours (starting next week 1/22):
  - M: 3:30p - 5p (right after lecture)
  - R: 11a – 1p
  - Or by appointment
  - Location: HFH 1155
  - Check the Google calendar on course website
  -
- You can reach me via
  - Piazza (highly recommended)
  - **Email: Include [CS16] on the subject line**

**Ask me about:**

- Course content!
- The how and why of what we are learning

**Tell me about:**

- Yourself!
- Experience in the class
- Interaction with the staff
- Climate of the labs

# Course staff

Sierra

Yanju

**TAs and peer mentors about:**

- One-one help in labs
- Feedback on code
- Answer questions on course content
- Available during “schedule” and “open labs” in Phelps 3525

Graham

Yossi

Bryanna

Annan

Barbara

Madhu

## Peer Mentors

# How to succeed in this course - first steps

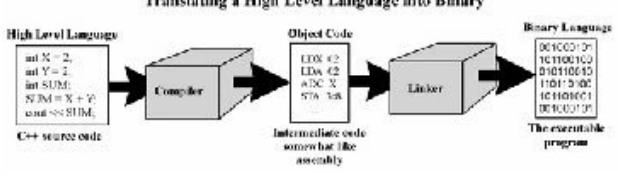
- Complete the questionnaire that is part of lab00 before tomorrow's section
- Come to instructor office hours and introduce yourself
- Setup a regular time to meet outside of section time with your
  - Mentor
  - Programming partner
- Communicate with the staff in person and on:

PIAZZA

# About this course

You will learn :

- **C++** (really the C part of C++) - why?
- Understand **what goes on under the hood** of C++ programs - why?
- Learn how to **debug** better
- **Solve fun problems :)**




Course Logistics:
more on the course website: https://ucsb-cs16-s18-mirza.github.io/

- Grading
  - Class and section participation (iclickers): : 2%
  - Homeworks/Quizzes (due every week) : 8%
  - Lab (programming) Assignments(due weekly) : ~~40%~~ 30%
  - Midterm exam: : ~~20%~~ 30%
  - Final exam : 30%

- No makeups for exams. Make sure you have no scheduling conflicts with exams
- You have 48 hours grace period to submit the labs – choose wisely. DO NOT contact the instructor or TAs for extensions unless you have a real emergency
- ATTENDENCE in sections and lectures is REQUIRED!
- To complete the labs you need a college of engineering account. If you don't have one yet, send an email to help@engineering.ucsb.edu

# iClickers: You must bring them

- Buy an iClicker at the Bookstore
- Register it on GauchoSpace (I will make an announcement on Piazza)
- Bring your iclicker to class

# Assigned Reading from

- Problem Solving with C++, Walter Savitch, Edition 9 6r 10

You must attend class and lab sections
You must prepare for class
You must participate in class

# Clickers out – frequency AB

# 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 UNIX command line

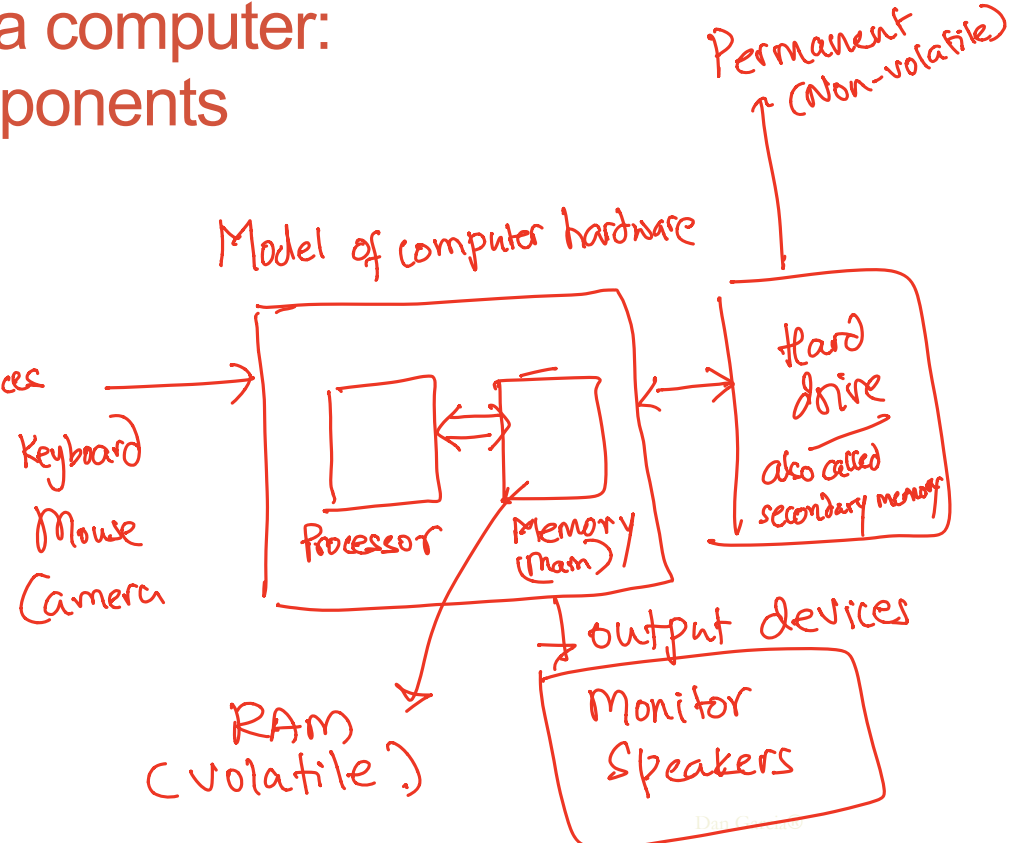
A. Know nothing or almost nothing about it.
B. Used it a little, beginner level.
C. Some expertise, lots of gaps though.
D. Lots of expertise, a few gaps.
E. Know too much; I have no life.

# 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)

# Abstracted view of a computer: Five hardware components

- Input devices
- Output devices
- Processor – How fast?
- Main memory – How large?
- Secondary memory

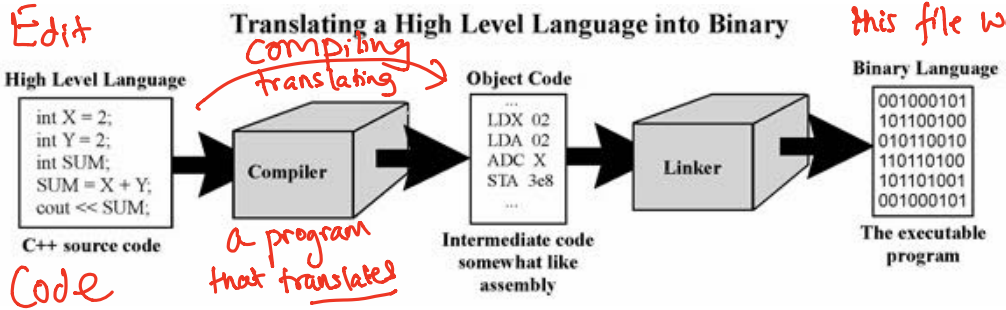


# The different stages of writing C++ code

- Editing – basically entering code in a text file
- Compiling – converting your code in a form the processor can understand (using another program called a compiler)
- Running – executing the binary version of your program on the processor

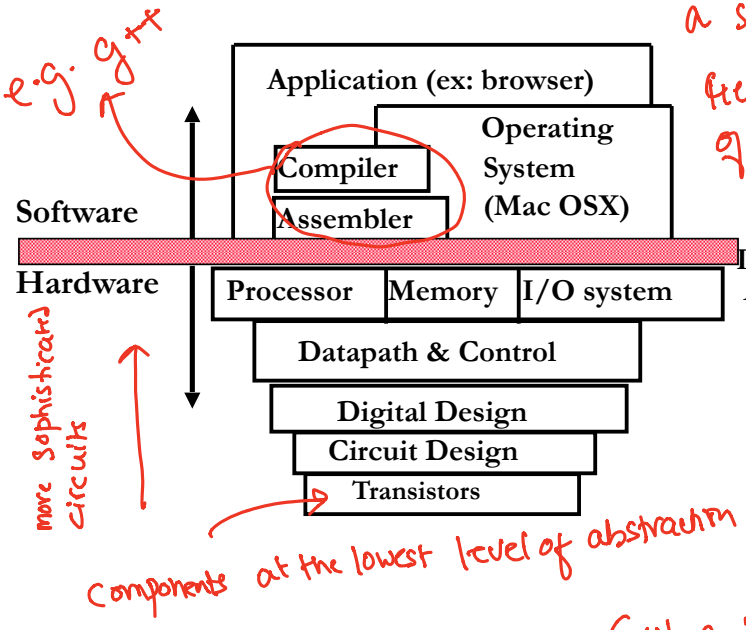
Our compiler of choice is g++

In the example demo shown in class this file was called a.out




LIVE DEMO of writing a simple C++ program

# How do we handle complexity?




a symbolic representation of machine code

```
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]
```

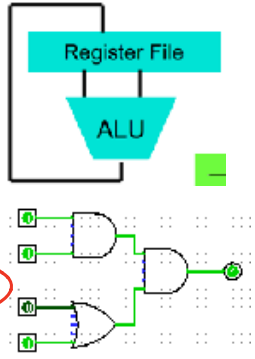
C++ code translated by compiler (eg g++) to machine code

```
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
```

Instruction Set Architecture

machine code (version of a program that can be executed on a processor)

(not readable by humans)




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

# Q: Which of the following converts a high level language to machine language

A. Main Memory
B. Secondary Memory
C. Processor
D. Compiler
E. Operating System

# Lab 00: Must be done individually

Before coming to the lab:

- Read the lab00 writeup
- Complete the “About you” questionnaire on lab00
- Get a CoE account if you don’t have one already.
- You can check if you have a working account by trying to remotely log into csil-02.cs.ucsb.edu

Key learning goals of lab00:

- 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

LIVE DEMO

# Basic structure of a C++ program

```
// name of the program as a comment: hello.pp
// Everything after the double slash is a comment
#include <iostream>
// Include the “modules” needed for basic input output
using namespace std; // using the Standard C++ library

int main(){  Program execution starts here
   //Write code here
 return 0;
}
```

# Next time

- Github
- simple flow control- for, while loops, nested and multi-way if-else

In class demo:

- Opened a terminal (also called shell program)

  In the terminal I did the following:

  - navigated the file system using unix commands
  - created new directories
  - created new files (using an editor like vim or emacs)

- Wrote a simple "helloworld" program in C++

Refer to the code and the saved shell session published on the class website under lecture-1 notes.