

CS2200

Systems and Networks and Architecture

Spring 2024

Lecture 1: Introduction

Alexandros (Alex) Daglis
School of Computer Science
Georgia Institute of Technology
adaglis@gatech.edu

Introductions



Background

- PhD from EPFL (2018)
- Georgia Tech (since 2019)

Office: Klaus 2322

Office Hours: 30 min after each lecture



Alexandros (Alex) Daglis
Assistant Professor

School of Computer Science
Georgia Institute of Technology
Atlanta, GA 30332, USA

Email: adaglis@gatech.edu

Web: cc.gatech.edu/~adaglis/

Research Interests

- Computer Architecture
- Datacenters

- Networked Architectures and Systems Lab
- Current research areas:
 - Architectures for high-bandwidth, low-latency networked applications
 - Network-compute integration
 - FPGA-based SmartNICs
 - Next-generation memory system architectures

Two CS-2200 Sections

Prof Ramachandran (section A)
+ Prof Daglis (section B)



- **Same:** syllabus, resources, exams, assignments, TAs
- **Different:** lectures, slides, participation tracking
- Shared Canvas page

Teaching Assistants

- Head TA: Kaylia Mai



- Total TA team: ~18 TAs, final list TBA

Motivation

- How many of you are taking this because you're interested in the subject?
- How many of you are taking this because it's required?

Objectives

What's inside the box?



Learn a lot about systems

Have fun!

Prerequisites

CS 2110

- Sequential and combination logic
- Memory
- CPU/Datapath
- Instruction-set
- C programming
- Translation Process
 - Preprocessing, Compilation, Linking, Loading
- Linux
- etc.

Diagnostic

- A touchpoint for your prerequisites
- Released on Canvas, due this Friday!
- Worth a tiny bit of extra credit

☰ CS-2200-A, B, GR > Assignments > Diagnostic

Spring 2024

Home

Announcements

Syllabus

Modules

Assignments

Gradescope

TurningPoint

People

Quizzes

Diagnostic

Due	Friday by 11:59pm	Points	1	Available	Jan 8 at 12am - Jan 19 at 11:59pm
------------	-------------------	---------------	---	------------------	-----------------------------------

Please view and complete on Gradescope.

This is an assignment to gauge your preparedness for this course. It will not affect your overall grade.

You can obtain the assignment by using this link:

[Cs2200-diagnostic.pdf](#) 

Please print out the assignment and complete it on paper or with a writing tablet. For the adder question, we expect some kind of diagram or picture.

The assignment will be due on Friday 1/12/2024.

Course Outline

Broad exposure to computer systems

- Organization of the processor
- Memory hierarchy
- Storage devices
- Parallel processors
- Networking hardware
- Software abstractions in the operating systems for orchestrating their usage
- Networking protocols to connect the computer system to its environment

Major Topics - Semester Schedule

Weeks 1-6:

- Processor: datapath, interrupts, pipelining

Weeks 7-8:

- Process Abstraction and process scheduling

Weeks 9-11:

- Memory Management and Memory Hierarchy

Weeks 12-14:

- Parallel Systems
- Networking Protocols and Distributed Systems
- IO and Disk scheduling

Week 15:

- File Systems

Date	Day	Spring 2024	Wk	Lecture A/B		Reading (Chapter)	Dates		Labs
		Comments					Release	Due	
1/8/24	Mon		1						
	Tue	First Day		1	Introduction/Preassessment	1			
	Wed								
	Thu			2	Feedback/Processors	2			
	Fri								
1/15/24	Mon	MLK Day	2						
	Tue			3	Processors	2			
	Wed						HW1		Intro & CircuitSim install help
	Thu			4	Processors	2			
	Fri						P1		
1/22/24	Mon		3						
	Tue			5	Datapath	3			
	Wed						HW2	HW1	Datapath
	Thu			6	Datapath	3			
	Fri								
1/29/24	Mon		4						
	Tue			7	Interrupts	4			
	Wed						HW3	HW2	Datapath
	Thu			8	Interrupts/Performance	4, 5			
	Fri								
2/5/24	Mon		5						
	Tue			9	Interrupts/Performance	4, 5			
	Wed							HW3	Interrupts
	Thu			10	Pipelining	5			
	Fri						P2	P1: Processor	
2/12/24	Mon	Prog Rpts	6						
	Tue			11	Pipelining	5			
	Wed						HW4		Exam 1 & C/gdb tips
	Thu			12	Pipelining	5			
	Fri								

Details

- Exams
- Class Website
 - On Canvas
- Ed Discussion

Homework/Projects

- Significant homework and project component to go with each topic
- 3 lab hours in the course credit
- An **excellent** knowledge of C programming is essential for completing the projects

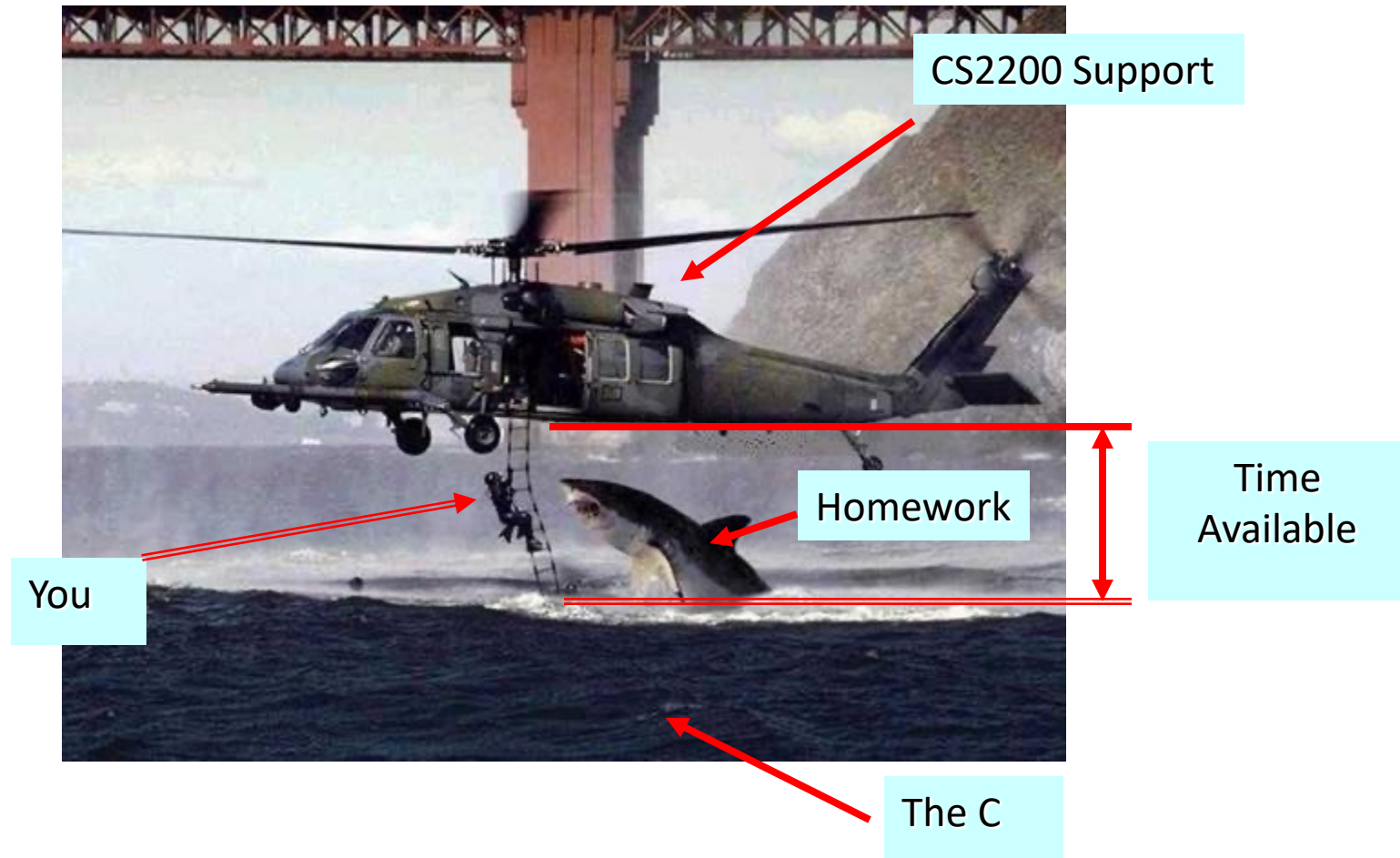
Assignments

9 Homeworks

5 Projects

1. LC2200 Datapath/Control
 2. I/O Interrupt handler
 3. Virtual Memory
 4. Process Scheduling
 5. Networking
- + optional 6th project for extra credit

A Word About Assignments



Exams

- 4 exams
- Online: Canvas + Honorlock
- Exams will take place first 75 minutes of Wednesday lab session
- 70% of questions will be released three days in advance
 - Remaining 30% questions will be hidden

Grading Scale

4 exams	60%
5 projects	25%
10 homeworks	10%
Participation (PS & Ed Discussion)	5%

Canvas/Gradescope

- Homework Assignments
- Assignment turnin
- Assignment retrieval
- Grades

Collaboration

- Collaboration is allowed and encouraged in this class
- Collaboration means that you can discuss assignments and help one another out but the code you submit must be your own.
- This is not a group project class.
- Each student is required to turn in their own homework or project.
- Each student will be required to **demo** their submission and will be expected to be able to understand and explain every part of it (project “interview”).

Project Collaboration Exceptions

- Best option A: Submitted project is individual effort
 - max credit 100%
- Acceptable option B: Submitted project is combined effort with peer
 - max credit 50%
 - Collaborative submission **must be declared in advance**
- Acceptable option C: Submitted project is a peer's, but well-understood
 - max credit 25%
 - Must be able to demo and fully explain solution to TAs during “interview”
 - **Must also be declared in advance**

Collaboration

- You are expressly forbidden to supply a copy of your homework to another student via electronic means. If you supply an electronic copy of your homework to another student and they are charged with copying you will also be charged. This includes public repositories. That is, you must not store your code in a location where others may access it, such as a public github repo.
- Homework and projects with suspicious degree of similarity will be sent to the Dean of Students Office of Student Integrity for investigation and possible prosecution. This includes assignments where nearly identical assignments are altered to make them appear different.

CS 2200 Rules and Regulations

- I. Ed Discussion messages are your best communication option. Please use email only if absolutely necessary. If you email the instructor and TA's please make sure to include "cs2200" in the subject. Also, please sign with your real name.

CS 2200 Rules and Regulations

2. You are responsible for turning in assignments on time. This includes allowing for unforeseen circumstances. In other words, plan ahead, start early, finish early.
3. In general, programming assignments should be turned in with a Makefile and all files needed to compile and run the program. The TA grading your submission should be able to make and run your program without adding files, repairing things, etc.
4. Tests must be taken at the scheduled date and time.

CS 2200 Rules and Regulations

5. If you need a certain grade in order to stay in school, maintain a scholarship, etc. the time to worry about this is **right from the beginning of the course** not during the week before finals. Grades are based on demonstrated performance not individual need based on factors external to the course. Please do not request special consideration based on this type of situation.
6. Final grades will be available from OSCAR normally sometime the week after finals. You may review your final and discuss your grades during the following semester in which you are attending GT. Grades will not be discussed over break.

CS 2200 Rules and Regulations

7. If you have any personal problems (family/illness/etc.) please email the Dean of Students' office. They are equipped and authorized to verify the problems and they will issue a note to all your instructors making them aware of the problem and requesting whatever extension, etc. is necessary.
8. Canvas announcements should be checked daily. Official announcements about course matters will be posted there. Complaints, questions about your personal problems, etc. should be discussed with your instructor in person or via private Ed Discussion notes.

CS 2200 Rules and Regulations

9. Out of consideration to your fellow students please turn off cell phones, beepers, wristwatch alarms, etc. Also, make every effort to be on time for class.
10. Complaints about any aspect of the course should be directed to the course instructor during office hours or via private Ed Discussion notes. Feedback is always welcome.

CS 2200 Rules and Regulations

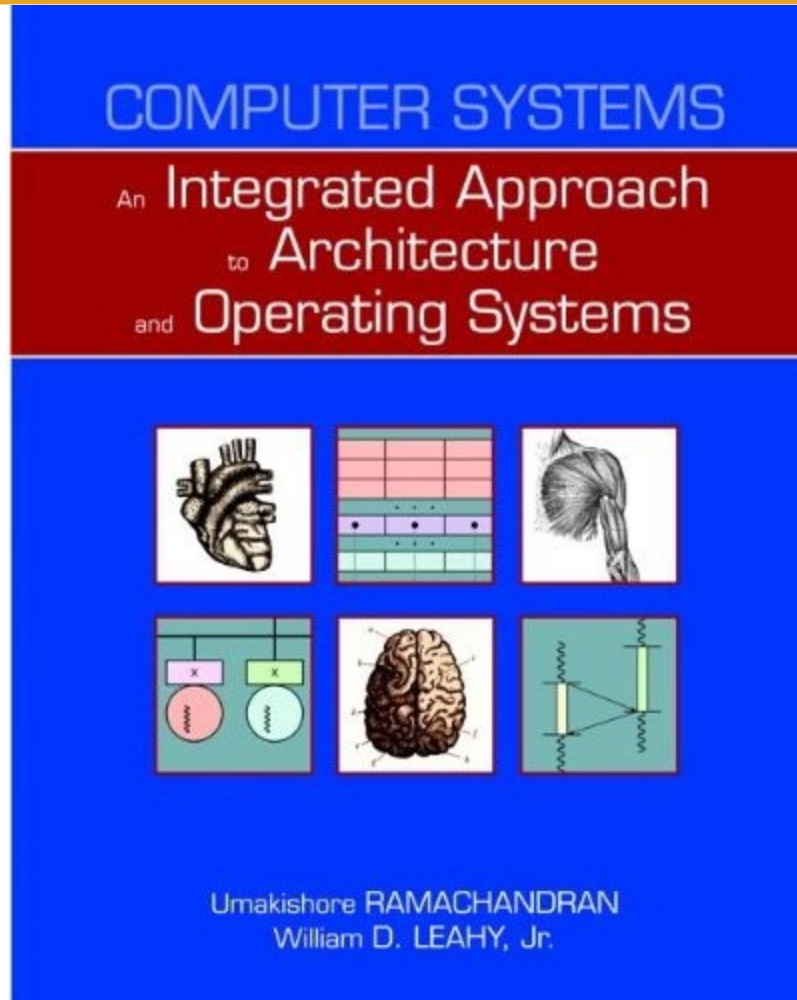
11. The deadline for regrades is **2 weeks** after an assignment grade is posted or returned to you. After this deadline no grade changes will be made.

In general, please read Syllabus on Canvas carefully!

CircuitSim

- We will be using CircuitSim for the Projects
- [Google](#): CircuitSim

Textbooks



- Paperback: 784 pages
- Publisher: Addison Wesley; 1 edition (August 9, 2010)
- Language: English
- ISBN-10: 0321486137
- ISBN-13: 978-0321486134

- Also check errata, available on Canvas

PointSolutions

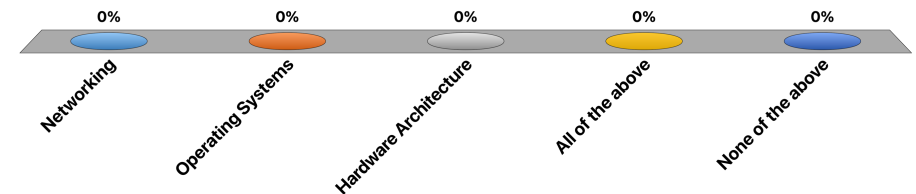
- Used for in-lecture questions and as graded participation component!
- Download “PointSolutions” app on your phone
- Sign in with your GT credentials
- Session ID shown on top left (“dag2200” in this case)
- Question will pop up on your screen

- You will see this  on the top left corner of each PS slide



What topic are you interested in the most?

- A. Networking
- B. Operating Systems
- C. Hardware Architecture
- D. All of the above
- E. None of the above



Questions?



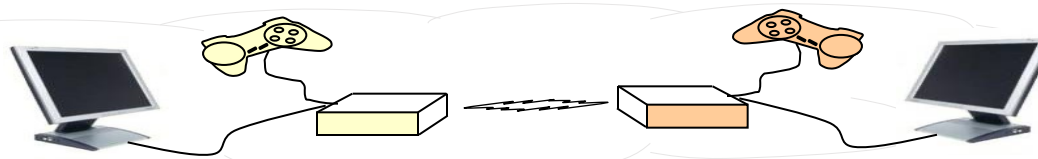
**Looking at the
big picture!**

Abstraction

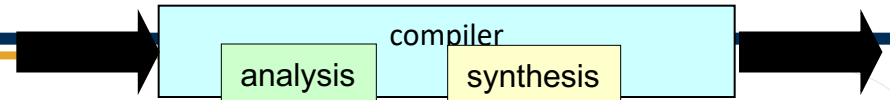
Levels of Abstraction



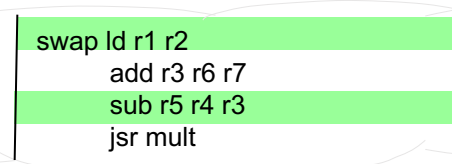
Networked
Video Game



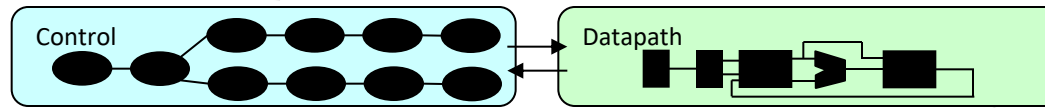
Compiler



Instruction
Set



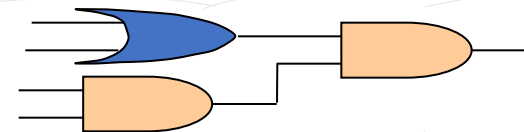
Datapath &
Control



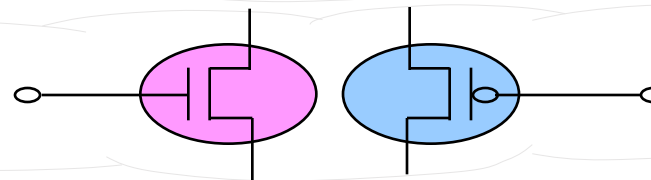
Logic
Elements



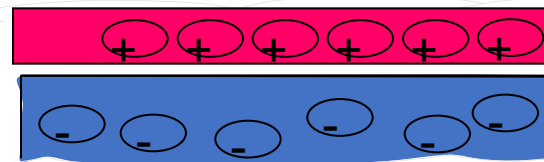
Gates




Transistors



Electrons &
Holes



What will we talk about in this course?



Application (Algorithms expressed in High Level Language)
System software (Compiler, OS, etc.)
Computer Architecture
Machine Organization (Datapath and Control)
Sequential and Combinational Logic Elements
Logic Gates
Transistors
Solid-State Physics (Electrons and Holes)

The Role of the Operating System?

- Resource manager—sharing
- Provide consistent interface to resources
- Job scheduler

Client Application (Halo 3)



*Player clicks mouse
cursor on target*

Client Application (Halo 3)



*Player clicks mouse
cursor on target*

**OS: Recognizes interrupt
Sends it to client application**

*It's a mouse
interrupt!*

Client Application (Halo 3)



*Player clicks mouse
cursor on target*

**OS: Recognizes interrupt
Sends it to client application**

*It's a mouse
interrupt!*

CLIENT

Client Application (Halo 3)



*Player clicks mouse
cursor on target*

**OS: Recognizes interrupt
Sends it to client application**

*It's a mouse
interrupt!*

CLIENT

Client Application creates
message to send to server
application



Client Application (Halo 3)



OS: Recognizes interrupt
Sends it to client application

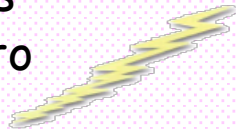
*It's a mouse
interrupt!*

CLIENT

Client Application creates
message to send to server
application



OS: Sends
Message to
server



Client Application
(Halo 3)



OS: Recognizes interrupt
Sends it to client application

It's a mouse interrupt!

CLIENT

Client Application creates message to send to server application



SERVER

OS: Sends Message to server

OS: Receives Message sends to server application

Got a message!

Client Application (Halo 3)



OS: Recognizes interrupt
Sends it to client application

It's a mouse interrupt!

CLIENT

Client Application creates message to send to server application



OS: Sends Message to server

SERVER

Application examines message and state of game and determines Master Chief dies! Sends message back to client.

OS: Receives Message sends to server application

Got a message!

Client Application
(Halo 3)



*Player clicks mouse
cursor on target*

**OS: Recognizes interrupt
Sends it to client application**

*It's a mouse
interrupt!*

CLIENT

Client Application creates
message to send to server
application



SERVER

OS: Sends
Message to
server

*Application examines
message and state of
game and determines
Master Chief dies! Sends
message back to client.*

OS: Receives
Message sends
to server application

Got a message!

OS: Sends
Message to client

Client Application
(Halo 3)



*Player clicks mouse
cursor on target*

**OS: Recognizes interrupt
Sends it to client application**

*It's a mouse
interrupt!*

CLIENT

Client Application creates
message to send to server
application



SERVER

OS: Sends
Message to
server

*Application examines
message and state of
game and determines
Master Chief dies! Sends
message back to client.*

OS: Receives
message and
sends it to
application

OS: Receives
Message sends
to server application

Got a message!

OS: Sends
Message to client

Client Application (Halo 3)



**OS: Recognizes interrupt
Sends it to client application**

It's a mouse interrupt!

CLIENT

Client Application creates message to send to server application



Client Application generates required images, etc.
Sends I/O requests to OS

SERVER

OS: Sends Message to server

Application examines message and state of game and determines Master Chief dies! Sends message back to client.

OS: Receives message and sends it to application

OS: Receives Message sends to server application

Got a message!

OS: Sends Message to client

Client Application (Halo 3)



*Player clicks mouse
cursor on target*

**OS: Recognizes interrupt
Sends it to client application**

*It's a mouse
interrupt!*

CLIENT

OS changes I/O devices to show
Master Chief blowing up!!!

uh oh!

Client Application creates
message to send to server
application



Client Application generates
required images, etc.
Sends I/O requests to OS

SERVER

OS: Sends
Message to
server

*Application examines
message and state of
game and determines
Master Chief dies! Sends
message back to client.*

OS: Receives
message and
sends it to
application

OS: Receives
Message sends
to server application

Got a message!

OS: Sends
Message to client

Client Application (Halo 3)



OS: Recognizes interrupt
Sends it to client application

It's a mouse interrupt!

CLIENT

OS changes I/O devices to show
Master Chief blowing up!!!

uh oh!

Client Application creates
message to send to server
application



SERVER

Client Application generates
required images, etc.
Sends I/O requests to OS

OS: Sends
Message to
server

OS: Receives
message and
sends it to
application

Application examines
message and state of
game and determines
Master Chief dies! Sends
message back to client.

OS: Receives
Message sends
to server application

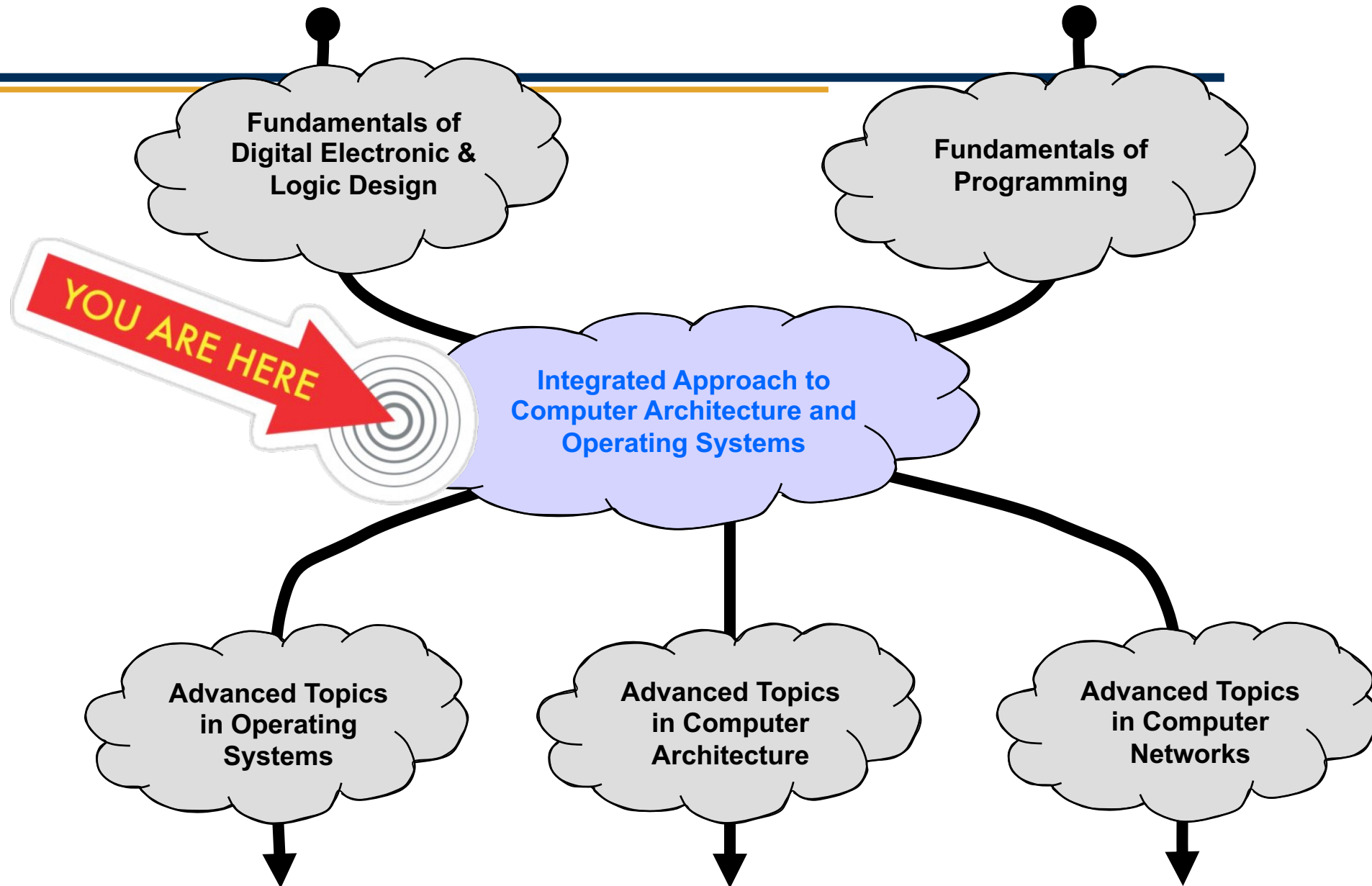
OS: Sends
Message to client

Got a message!

Why are we studying this?

- Very cool!
- Need to have working knowledge one level down
- Somebody has to understand how the darn things work
- Somebody has to tell CmpE's what we need
- Be able to predict GameStop stock movement





STAY CALM & TUTOR ON!

LET US HELP YOU FACE YOUR MIDTERMS

FIND YOUR SUCCESS BY ASKING FOR HELP AND
TAKING ADVANTAGE OF THE AVAILABLE
RESOURCES AT TECH!



TUTORING

SCHEDULE A FREE 1-ON-1 TUTORING SESSION! CHECK
OUT OUR SUPPORTED COURSES AND SCHEDULE AN
APPOINTMENT HERE:

[TUTORING.GATECH.EDU/TUTORING](https://tutoring.gatech.edu/tutoring)

PLUS

JOIN A PLUS SESSION! MANY INTRODUCTORY
COURSES HAVE FREE WEEKLY STUDY SESSIONS LED
BY PEERS WHO HAVE SUCCESSFULLY TAKEN THE
COURSE. INFO MAY BE ON YOUR COUSE CANVAS
SITE, AND YOU CAN CHECK IT OUT HERE:

[TUTORING.GATECH.EDU/PLUS-SESSIONS](https://tutoring.gatech.edu/plus-sessions)

Questions?