# CSCI 310 — Computer Organization

Jeremy Iverson

College of Saint Benedict & Saint John’s University

# logistics

instructor

Jeremy Iverson (`jiverson002@csbsju.edu`)
MAIN 258, (320) 363-5542
office hours: MTThF 12–1pm

textbook

*Computer Systems*, 5th Edition, Warford

website

`https://csbsju.instructure.com/courses/16318`

# objectives

identify and describe the key components of a computer system and how the components interact

explain the layered architecture of computer systems and how each layer relates to the others

explain how data and programs are represented inside computers

explain the role of operating systems in managing storage, processes and programs

# evaluation

seven (7) assignments

- mix of worked problems and programming
- do your own work
- use each other as resources only
- due dates are strict (no partial credit for late assignments)

three (3) topical exams and one (1) cumulative

- closed book / open note

point distribution

assignments: 32% total
exams: 14% each
final: 26%

# need more info...

see `course syllabus`!

# a word to the wise

computers $\neq$ magic
programming $\neq$ magic

# a word to the wise

$$\text{computers} \neq \text{magic}$$
$$\text{programming} \neq \text{magic}$$

banned phrases

“ it worked when I tried it earlier”

“I don’t know, it just works”

“but it gives the right answer”

# a dose of honesty

We are not learning new problem-solving abstractions.

# a dose of honesty

We are not learning new problem-solving abstractions.

We are unpacking the abstractions that we depended on in previous courses — seeing how the expressiveness of our favorite programming languages is represented and carried out by a computer.

We are building a foundation for understanding how future abstractions can be understood by a computer.

# a dose of honesty

We are not learning new problem-solving abstractions.

We are unpacking the abstractions that we depended on in previous courses — seeing how the expressiveness of our favorite programming languages is represented and carried out by a computer.

We are building a foundation for understanding how future abstractions can be understood by a computer.

This process is often tedious and mechanical.

# a good question

## why bother?

better debugger

better programmer

better problem-solver

# a look ahead

become proficient in C

internalize the von Neumann architecture

develop a mental model for program execution

get very comfortable with binary

# a bit of parting advice

remember that this is a 300-level course

# a bit of parting advice

remember that this is a 300-level course

if something is confusing, tell me

# teaching philosophy

Your job is to empower those you teach; when you do for them what they should be doing for themselves, you create dependency rather than empowerment.

It is easy to give in to the frustration that results from seeing amazing possibilities for the people you are teaching, and you want it more for them than they want it for themselves.

Don’t give in to that frustration!

— Based on passage from “Resisting Happiness” by Matthew Kelly questions?

# activity

download this slide deck and follow the instructions on the next slide

# next slide

find out what the reading is for tomorrow