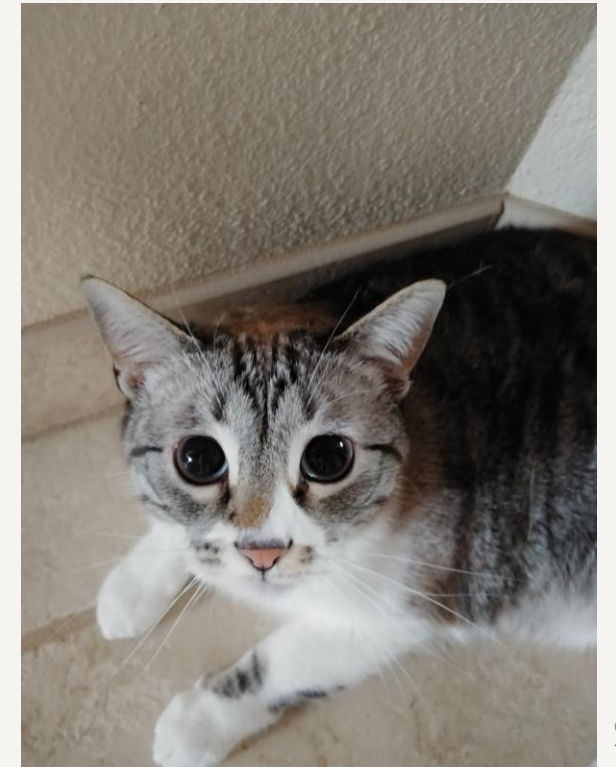


CPSC 230 Intro to Computer Science Day 1

SAM BERNSEN

Graduate Student Instructor & EECS Masters Student

- B.S. Computer Engineering at Chapman
- 1st semester teaching
 - Feedback welcome!
- San Diego native
- Hobbies: Baseball, golf, volleyball, frisbee



Cadence Design Systems Internship

- Summer hardware verification internship
- Programming language: Verilog & SystemVerilog
- Team project: Sous Vide Controller RTL verification
 - Worked alongside Digital and Analog teams working on synthesis, place and route, and process node migration

The Cadence logo is displayed in a bold, lowercase, sans-serif font. The letter 'a' is unique, featuring a red horizontal bar above it. The letters 'c', 'd', 'e', 'n', and 'e' are dark grey, while the first 'c' is black.

CLASSROOM DISCUSSION

- WHO ARE YOU?
 - NAME
 - YEAR (FRESHMAN, SOPHOMORE, ETC.)
 - MAJOR
 - LEAST FAVORITE DRINK
 - ANY PRIOR PROGRAMMING EXPERIENCE?

Quick Syllabus Overview

QUESTIONS?

- Myself
- Course content
- Class policies
- Requirements

Day 2: Computer Basics

HOW DO COMPUTERS "THINK" AND OPERATE?

TERMINOLOGY

PROGRAMMING LANGUAGES

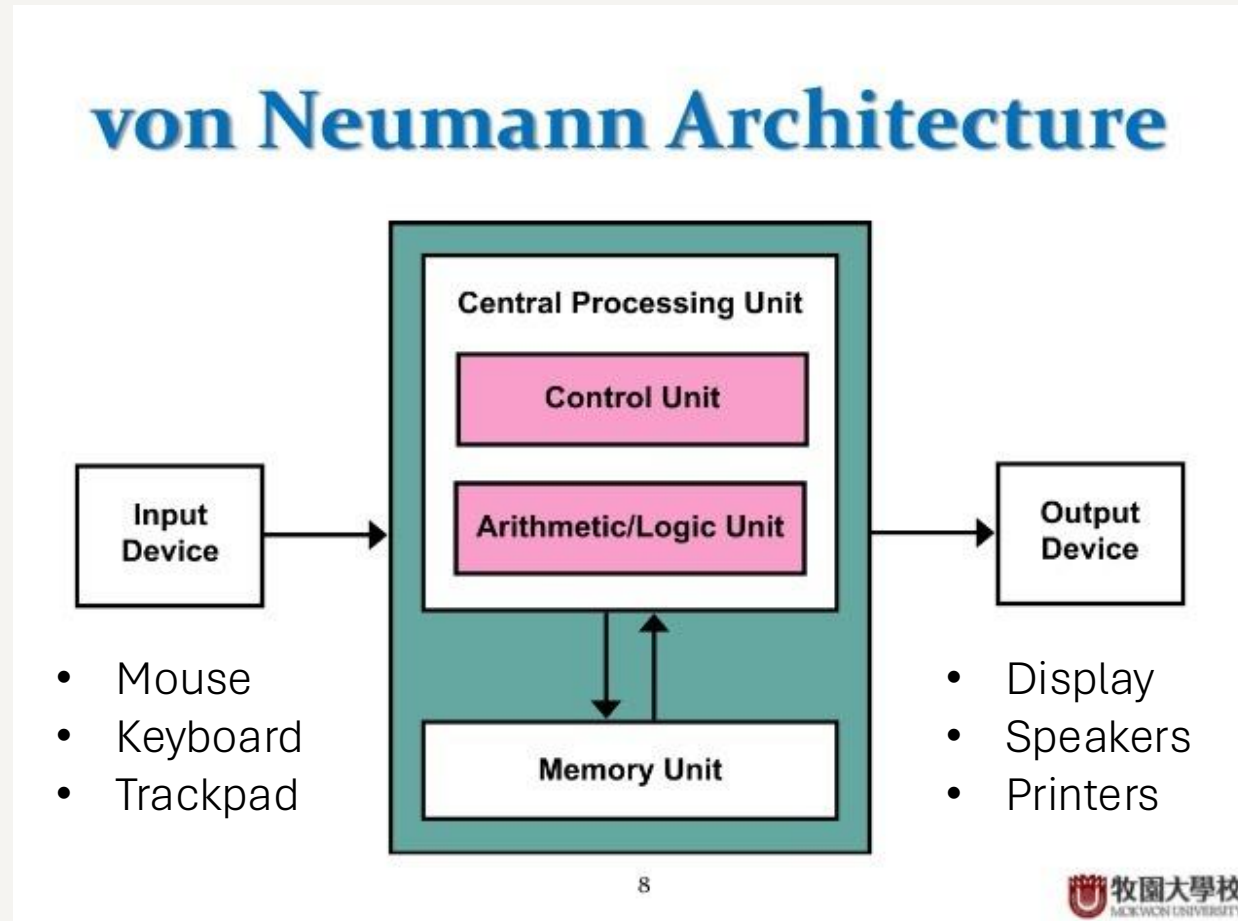
What is a Computer?

A device that processes data. It can:

- Execute software programs
- Perform mathematical operations
- Transmit data via a computer network system
- Assist in daily activities



Basic von Neumann Architecture



Basic Computer Components

- CPU/Processor: processes instructions from software
- Volatile memory
 - Main memory/RAM
- Non-volatile storage
 - Hard disk drive
 - Flash memory (e.g. Solid state drives)
 - Read only memory
- Input/Output
- Network: communication between computers

How Do Computers "Think?"

- All about DATA and INSTRUCTIONS
 - Tell the computer what data to operate on and what to do with the data
- Fetch, Decode, Execute, Store (Repeat)
- Binary: base 2 number system
 - Examples:
 - $3_{10} = 11_2$
 - $20_{10} = 10100_2$



What is Programming?

- Telling the computer what to do
- A set of instructions
- Manipulating values in memory
 - Sent to CPU for processing and output

Terminology

- Computer science: the study of computer and computational systems
 - Computer scientists design and analyze algorithms to solve problems
- Code: plain text computer program written in a particular programming language
 - In our case, we will be using the Python programming language
- Software: programs, scripts, or applications that run on a computer
 - Provide instructions of hardware to perform operations
- Script: a code file designed to execute as a program
- Programming language: a tool we can use to write these instructions
 - Allows us to direct the computer's operation without using machine language
 - How tedious would that be !?!

Programming Languages



- High level languages: easy for humans to read; hard for computer to understand
 - Python
 - JavaScript
 - C++(ish)
- Low level languages: harder for humans to read, easier for computer to understand (i.e. less steps to compile code into binary)
 - C
 - Cobol
 - Fortran
 - Assembly
- Final form of code: binary
 - All languages eventually get converted to binary for the computer to understand

Why is CS Important?

- So many industries rely on it!
 - Finance, transportation, semiconductor industry, healthcare, entertainment, meteorology, education, etc.
- Connected world requires reliable computing and software
- Computers without well optimized software are much less effective
- Who doesn't use a computer on a daily basis?

Cool Application: Short Demo

Day 3: Python Introduction & Environment Setup

WHY PYTHON?

THE POWER OF PYTHON

Why Python?

- Conceived in the late 80s by Guido van Rossum
- Named after Monty Python
- Python 3.0 released in 2008
- General purpose and high-level
 - Easier to understand for new programmers
 - Closer to pseudo-code
- FREE and open source
- Loosely typed: implicit data type inference

Loosely typed:

a = 5

v.s.

Strongly typed:

int a = 5

The Power of Python

- Language most similar to English
- Interpreted, not compiled
 - No need to explicitly invoke a compiler
- Many use-cases:
 - Game development
 - Machine learning
 - Web development
 - GPU programming with PyCuda



Benefits Of Python



Text Editors & Setup

- Setting up the environment
- Let's dive in!
- Possible local text editors (ideally most work done locally)
 - [VSCode](#)
 - PyCharm
 - Notepad++
- Cloud services
 - GitHub Codespaces
 - Google Colaboratory
 - Programiz Online Interpreter



Hello World!