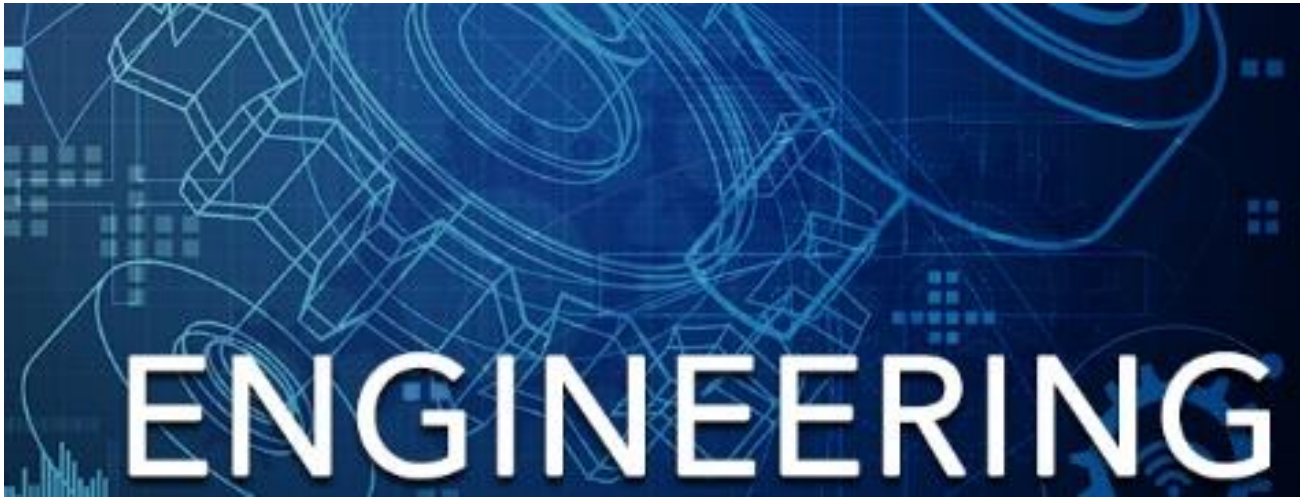
CS471 Lecture 01

Software Engineering Introduction and Motivation
Sommerville Ch2

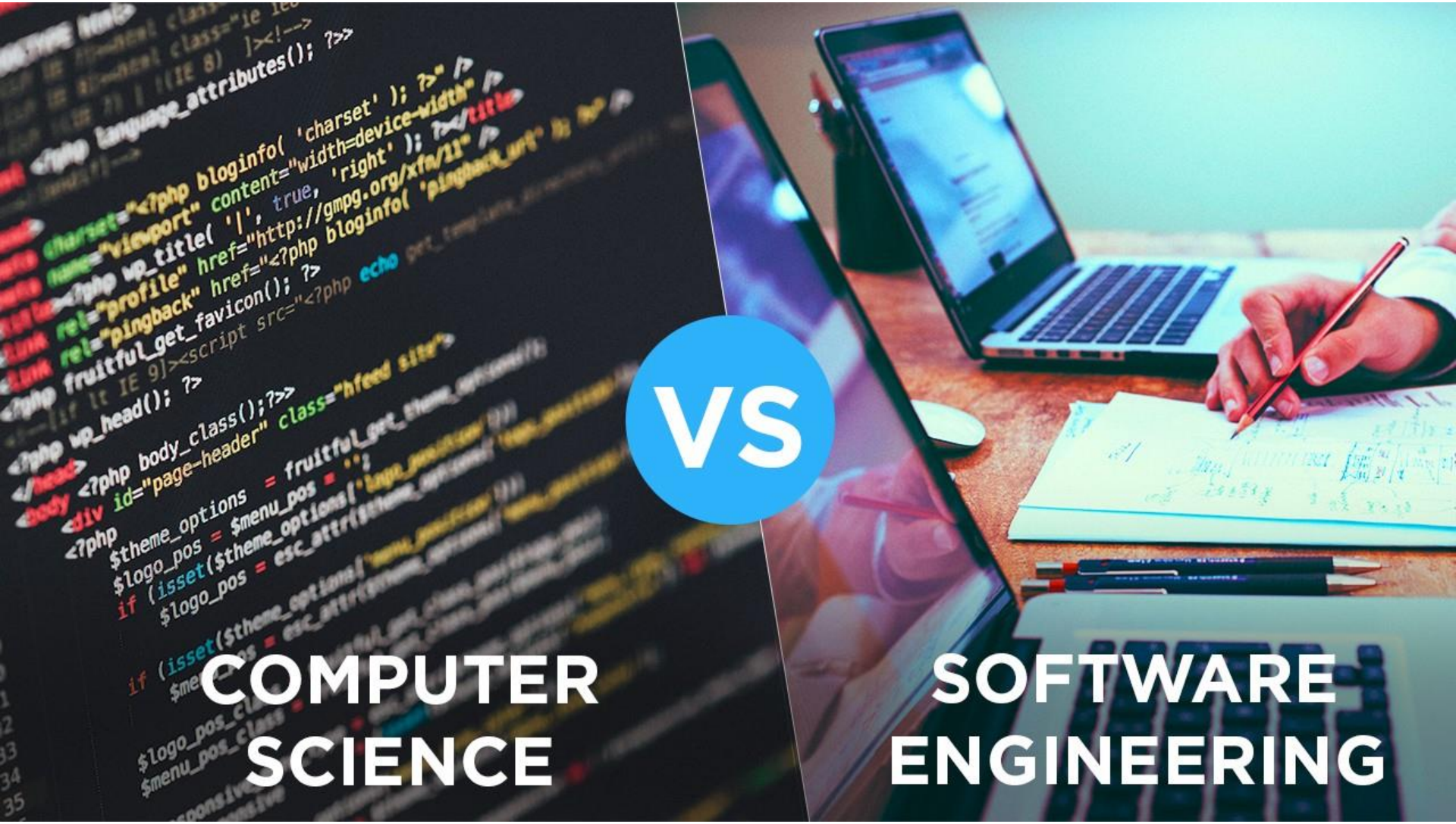
What is Engineering?

- "the application of mathematics, science, economics, empirical evidence, etc. to invent, innovate, design, build, maintain, research, and improve structures, machines, tools, systems, components, materials, processes, solutions, and organizations."



https://en.wikipedia.org/wiki/Engineering
http://vijayshekharacademy.com/various-engineering-streams-find-best-faculty-for-engineering-in-vijay-shekhar-academy/engineering-banner/

What is Software Engineering?



https://medium.freecodecamp.org/computer-science-vs-software-engineering-which-one-is-a-better-major-88482c38446b

Software Engineering Definitions

- "…an engineering discipline that is concerned with all aspects of software production from initial conception to operation and maintenance"

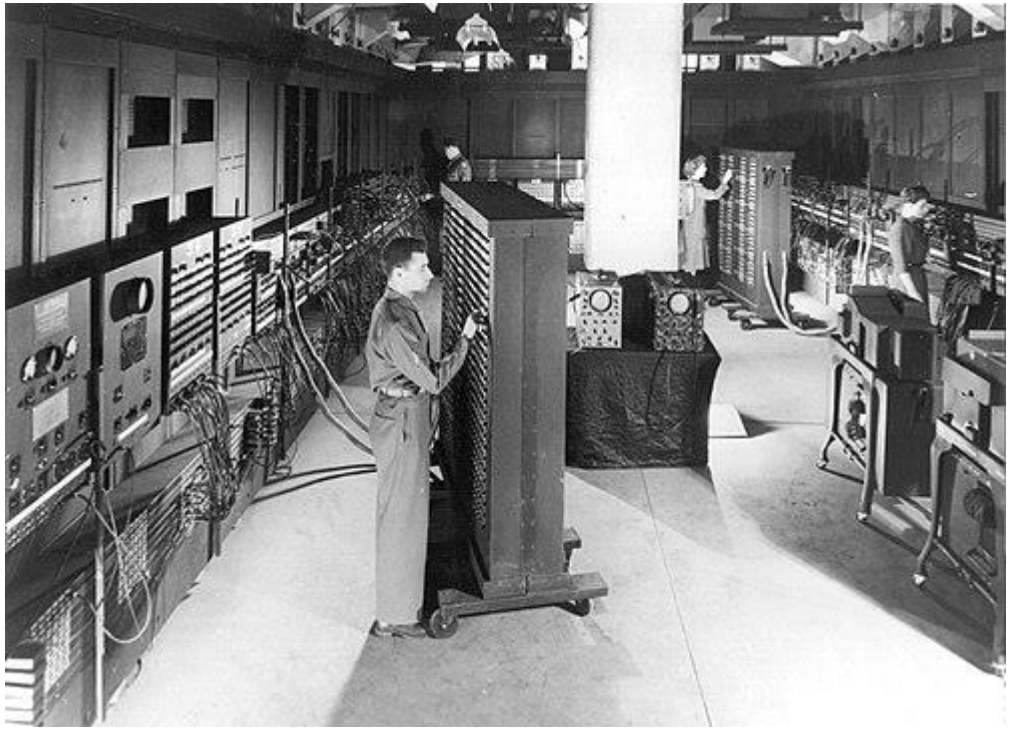
  -Sommerville

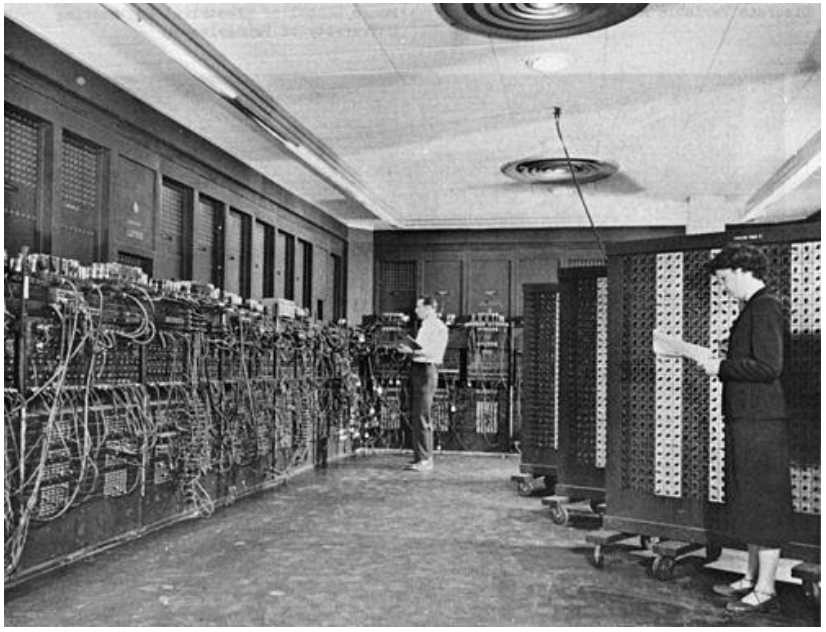
- "…the application of a systematic, disciplined, and quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software…"

  -IEEE

Software Engineering

- First software (early 50's)
  - cost of hardware dominates
  - programs seem to be less important





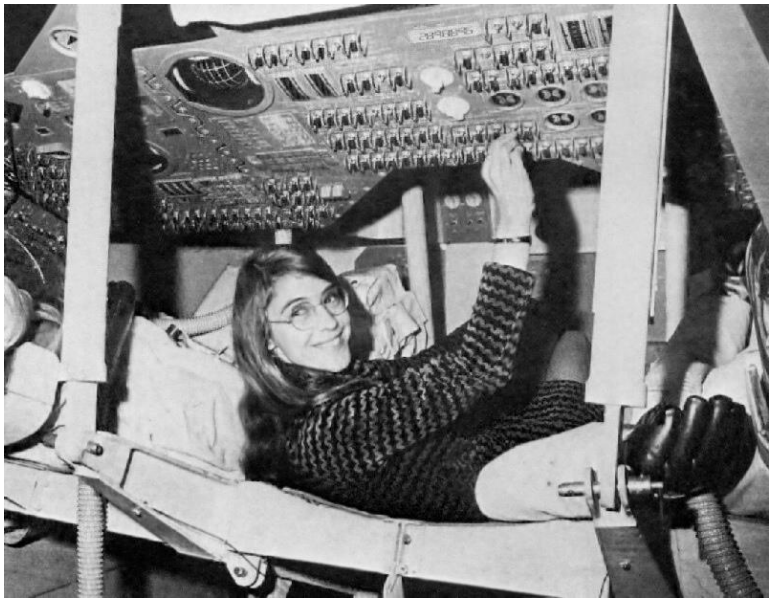
https://en.wikipedia.org/wiki/ENIAC

Software Engineering

- First software (early 50's)
  - cost of hardware dominates
  - programs seem to be less important
- Software crisis (late 60's)
  - hardware becomes cheaper
  - custom software becomes complex and expensive
  - software production lags behind the need
  - software engineering discipline is born (early 70's)

Nascent Software Engineering Applications: Aerospace

- Margaret Hamilton, Lead Flight Software Designer, Apollo Program
- Prevented an abort of the Apollo 11 lunar landing



Nascent Software Engineering Applications: Aerospace

- See source code at:

  https://github.com/chrislgarry/Apollo-11

- Clone and count lines of code (`cloc`*)

```
~/scratch/Apollo-11-master$ cloc .
```

*https://github.com/AlDanial/cloc

Nascent Software Engineering Applications: Aerospace

- See source code at:

  https://github.com/chrislgarry/Apollo-11

- Clone and count lines of code (cloc*)

```
~/scratch/Apollo-11-master$ cloc .
     182 text files.
     182 unique files.
       4 files ignored.

github.com/AlDanial/cloc v 1.72  T=1.00 s (181.0 files/s, 131059.0 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
Assembly                       176          14761          35732          80122
Markdown                         5             88              0            356
-------------------------------------------------------------------------------
SUM:                           181          14849          35732          80478
```

*https://github.com/AlDanial/cloc

Nascent Software Engineering Applications: Aerospace

- 80KLOC (i.e., 80,000 Lines of Code) written in Assembly
- 176 files
- 35K lines of comments
- Software Engineering has evolved considerably

Original Software Engineering Objectives

- Improve the following competing resources
  - Quality
  - Schedule
  - Cost

Original Software Engineering Objectives

- Improve the following competing resources
  - Quality
  - Schedule
  - Cost

- Largely focused on the development of large aerospace and enterprise applications:
  - Banks, financial
  - Telecommunication
  - Airports
  - Stock market, etc.

Questions about software

- Why does it take so long to get software completed?
- Why are costs so high?
- Why can't all errors be found before the software is put into production?
- Why is it difficult to measure the progress at which software is being developed?

High-level Explanations to Questions about software

- Software is developed (or engineered), not "manufactured" (in the classical sense)
- Most software is custom built rather than assembled from existing components
- Software does not "wear out" (as do traditional concrete products), but it "deteriorates" during requirements, design, development, maintenance

Software Engineering vs. [other] Engineering

- One of the essential technologies of today
  - essential for economy
  - essential for security
- Technology of the same importance as
  - mechanical engineering
  - electrical engineering, etc.
- How does software and engineering differs from other engineering fields?

Software Engineering vs. [other] Engineering

- Other branches of engineering use standardized tools and metrics to produce systems with predictable outcomes

- Mechanical and electrical engineers have big catalogs of standard parts they recycle into their creations vs. "reinventing the wheel"

Software Engineering vs. [other] Engineering

- Other branches of engineering use standardized tools and metrics to produce systems with predictable outcomes

- Mechanical and electrical engineers have big catalogs of standard parts they recycle into their creations vs. "reinventing the wheel"
  - Trust / compatibility of existing software components?

Software Engineering vs. [other] Engineering

"Local" implications

amazon web services

bitcoin

"Global" implications

Properties of Software*

*Brooks, Fred P. (1986). "No Silver Bullet — Essence and Accident in Software Engineering". Proceedings of the IFIP Tenth World Computing Conference: 1069–1076

Properties of Software – Accidental

- Accidental properties change from time to time
- Examples:
  - Programming language
  - Hardware speed, memory size
  - Architecture of the program
    - functional
    - object oriented

Properties of Software – Essential

- Intrinsic to software – determine its nature
- These do not change!

- Complexity
- Conformity/Interoperability
- Changeability
- Invisibility
  - not tangible
  - cannot use senses

What to Expect in CS471?

What to Expect in CS471: "Question all the Answers"

- Software Engineering is an active area of research
- Best practices continue to emerge from this research
- We supplement our texts with selected research papers

What to Expect in CS471: "Question all the Answers"

- You won't leave CS471 with all the answers
- You will leave thinking critically about the answers!
- You will leave on the trail of continuous education!

CS471: Related Courses

- CS472 provides a deep dive into software design
- CS474 provides a deep dive into software quality
- CS481 (Spring'19) provides a deep dive into a real world software project
- Continuing education following graduation!