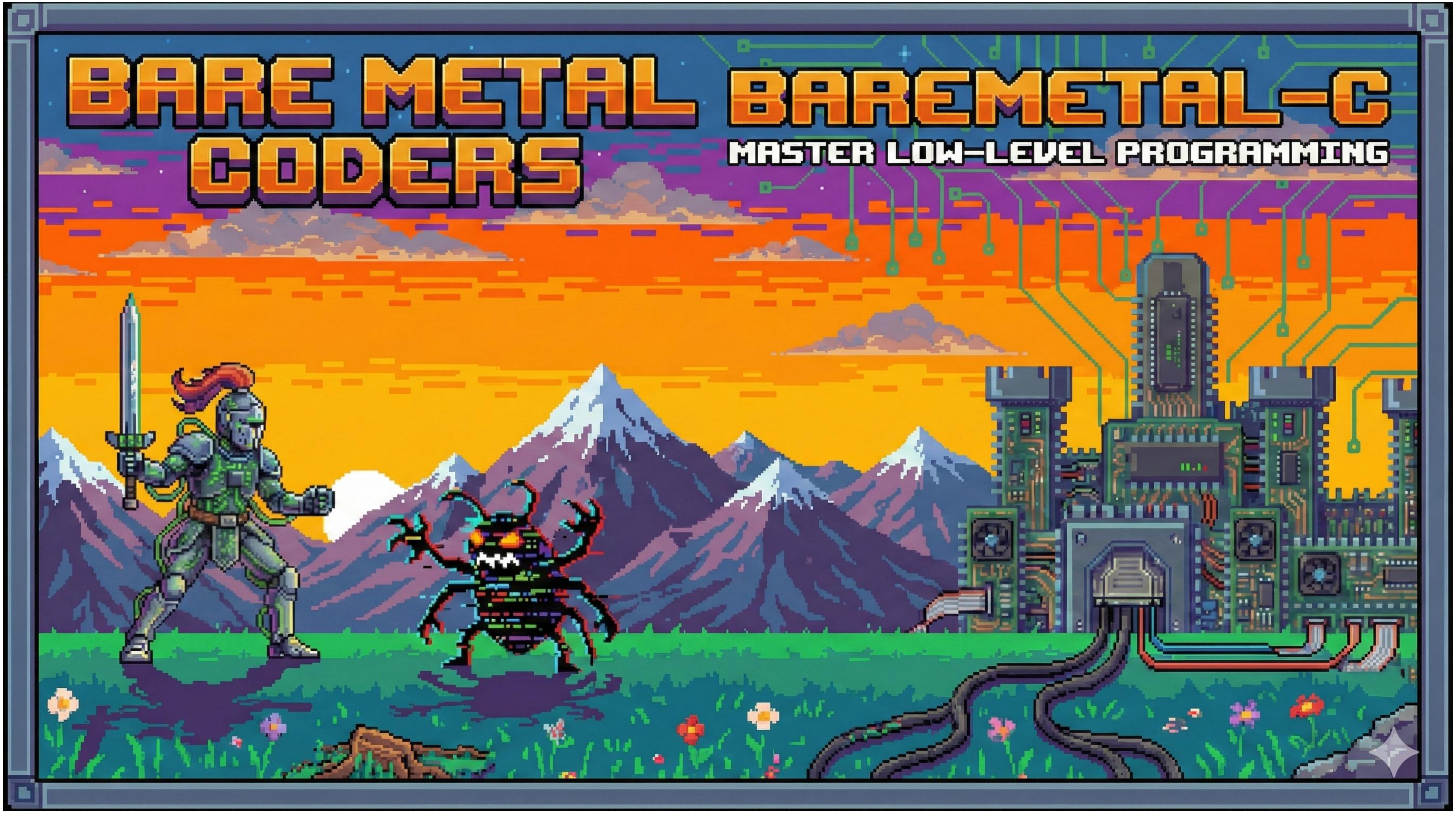
# Programming for Semiconductor Engineering:

# The First Half: Bare-Metal C

Kong Kritayakirana


BARE METAL CODERS
BAREMETAL-C
MASTER LOW-LEVEL PROGRAMMING

# About the instructor

- Name: Kong Kritayakirana kong.k@chula.ac.th
- Education: Ph.D. in EE, Stanford University, 1997
- Industry Experience:
  - Researcher: Hewlett-Packard Laboratories
  - ASIC Design Engineer: Juniper Networks, Inc.
  - Former Senior Advisor: Impact Electrons Siam Co., Ltd.
- Teaching Experience
  - VME, Assumption University
  - ISE, Chulalongkorn University

# 00 Intro to the first half of the course

- Learning Objectives
  - Modern computers are extremely complex

Approximate Hierarchy

| Endpoints | |
|---|---|
| Applications | Software |
| Operating System | |
| Computer Architecture | Hardware |
| Logic Circuits | |
| Semiconductor Devices | |
| Device Physics | |

# 00 Intro to the first half of the course

- Learning Objectives
  - Modern computers are extremely complex

|  | Examples |
|---|---|
| **Endpoints** | - Humans, pets, mouse, screen, other computers (networking) |
| Applications | - Chrome, Line, Games, ChatGPT, Web Server |
| Operating System | - Windows, MacOS, Linux |
| Computer Architecture | - CPU, GPU, RAM, Flash, Thumb Drive |
| Logic Circuits | - AND, OR, INV (NOT), Flip-Flops |
| Semiconductor Devices | - CMOS Transistors and wires |
| Device Physics | - Physics and chemistry of those transistors and wires |

# 00 Intro to the first half of the course

- Learning Objectives
  - Modern computers are extremely complex

What this course will cover

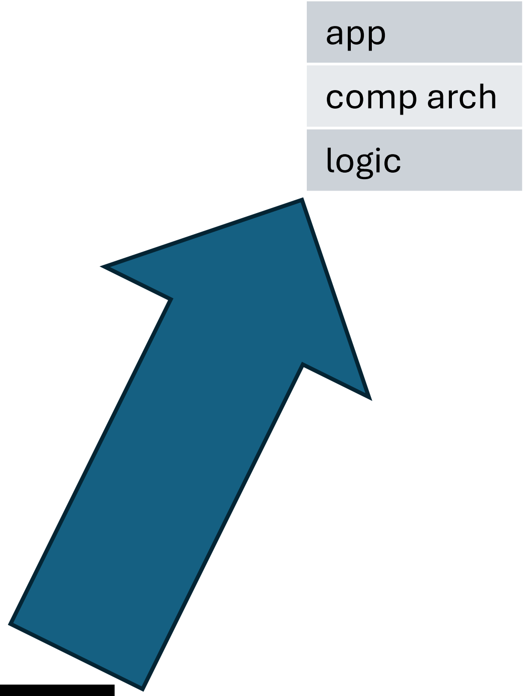
| Endpoints |
|---|
| Applications 70% |
| ~~Operating System~~ |
| Computer Architecture 15% |
| Logic Circuits 15% |
| Semiconductor Devices |
| Device Physics |

# 00 Intro to the first half of the course

- Learning Objectives
  - Modern computers are extremely complex

→ Bridge software and hardware

| Endpoints | |
|---|---|
| Applications 70% | Software |
| ~~Operating System~~ | |
| Computer Architecture 15% | Hardware |
| Logic Circuits 15% | |
| Semiconductor Devices | |
| Device Physics | |



app

comp arch

logic

# 00 Intro to the first half of the course

Slides are divided into 6 categories by symbols on *the top left*:

1. (No symbol) – teaching material – *covered on exam*
2. ⭐ Important – Critical concepts – *covered on exam*
3. 📌 Summary Slide – summarizing what's *covered on exam*
4. 📖 Self study – *covered on exam*
5. 🚀 Advanced topics – (optional) Read to gain deeper knowledge
6. ℹ️ Side information – (optional) Supplementary materials

# 00 Intro to the first half of the course

- Learning Objectives
  - Modern computers are extremely complex
    - We will learn some basic logic and computer architecture
  - Learn about a standard software programming language C
    - That can bridge to hardware easily
  - See how software “drives” hardware around
    - See physical lights being turned on or off (albeit in a simulator)
  - See how software “reads” from hardware
    - Read switches and keypads
  - Build a higher-level software concepts
    - Understanding hardware-software interaction facilitates learning

# 00 Intro to the first half of the course

- Assumptions and philosophy for this course
  - Very little pre-requisite
    - Knowing high-school logic is enough
    - Hope you passed the first programming course
  - **We expect a step-up in academic maturity – no pre-requisite ≠ easy**
  - Work more, earn more.
    - We will give you assignments that take a lot of time. You put in more effort, you earn better grades
  - No hardware design, but if you're interested, let us know
    - Won't need to use SimulIDE to do any hardware design.
    - We will give you all designs you need. You only write C and load into the simulator.
  - Weekly homeworks
    - It's easy to think you understand in class, doing the work proves true understanding.

# 00 Intro to the first half of the course

How to use software demo

Notation: [05-204]

05 is the hardware version in `BareMetal-C/sim`

204 is the software version in `BareMetal-C/code/_lectures`

Remember the 5 symbols:

⭐ 📌 📖 🚀 ℹ️