# CS240: Programming in C

## Lecture 1: Class overview.

# Why learn C (1)

- **C is one of the foundations for CS:**
  - Contains/applies principles from programming languages, computer architectures, operating systems, network communication, database, graphical user interface (GUI), graphics, image processing, parallel processing, multi-threads, real-time systems, device drivers, data acquisition, algorithms, numerical analysis, and computer game.

# What does this buy you?

- **Understanding:** understand better the interaction between machine and software:
  - "…teaches individuals how computers really work"
  - "…built a foundation you'll be thankful for every 300+ level course "

# Why learn C (2)

- **C is the most commonly used programming language in industry.**
  - Next two popular are Java and C++
  - Language of systems programming: low-level control over the OS, networking, crypto operations, email, games, embedded systems have higher performance when written in C

*http://www.langpop.com*

# What does this buy you?

- **Helps you be as prepared as possible for a job:**
  - Most of the employers want candidates to know multiple languages
  - Will prepare you better for a job interview
  - Gives you more opportunities within a company

# Why learn C (3)

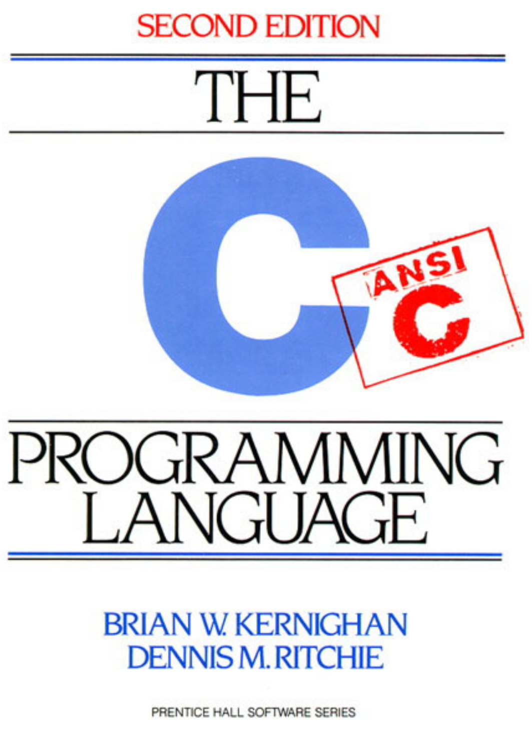
- **C is the base for almost all popular programming languages.**
- Because of the performance and portability of C, almost all popular cross-platform programming languages and scripting languages, such as C++, Java, Python, Objective-C, Perl, Ruby, PHP, Lua, and Bash, are implemented in C and borrowed syntaxes and functions heavily from C.
- Almost all languages can interface with C and C++ to take advantage of a large volume of existing C/C++ libraries. Many of their toolkits, modules or packages are written using C or C++.

# What does this buy you?

- **It will help you learn quickly other languages**
- **It will allow you to interface with many other languages**

# Reference material

- The C Programming Language, Brian W. Kernighan and Dennis M. Ritchie, 2nd Edition
- Lecture slides posted online



SECOND EDITION

THE C PROGRAMMING LANGUAGE

ANSI C

BRIAN W. KERNIGHAN
DENNIS M. RITCHIE

PRENTICE HALL SOFTWARE SERIES

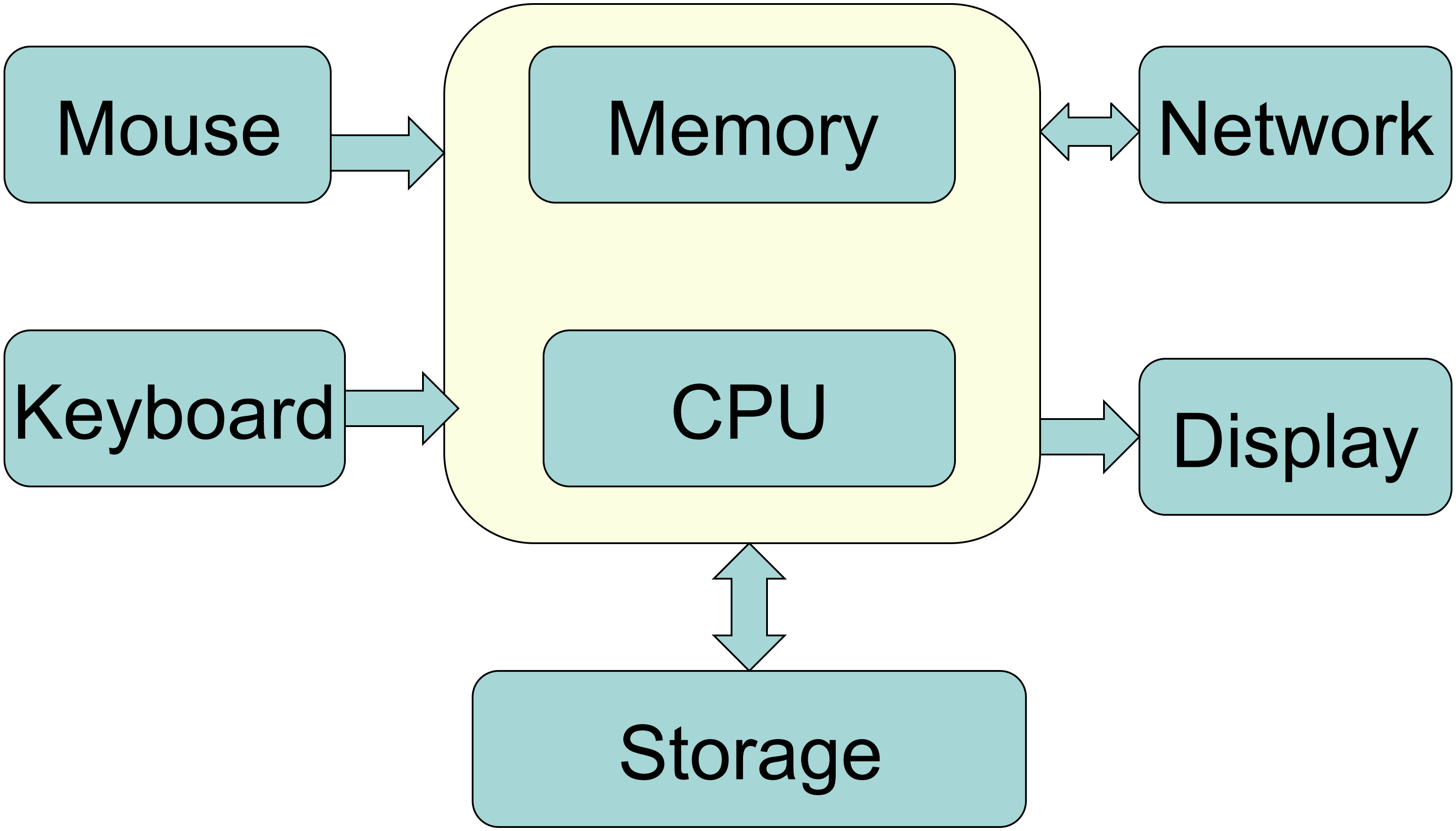
# How to study

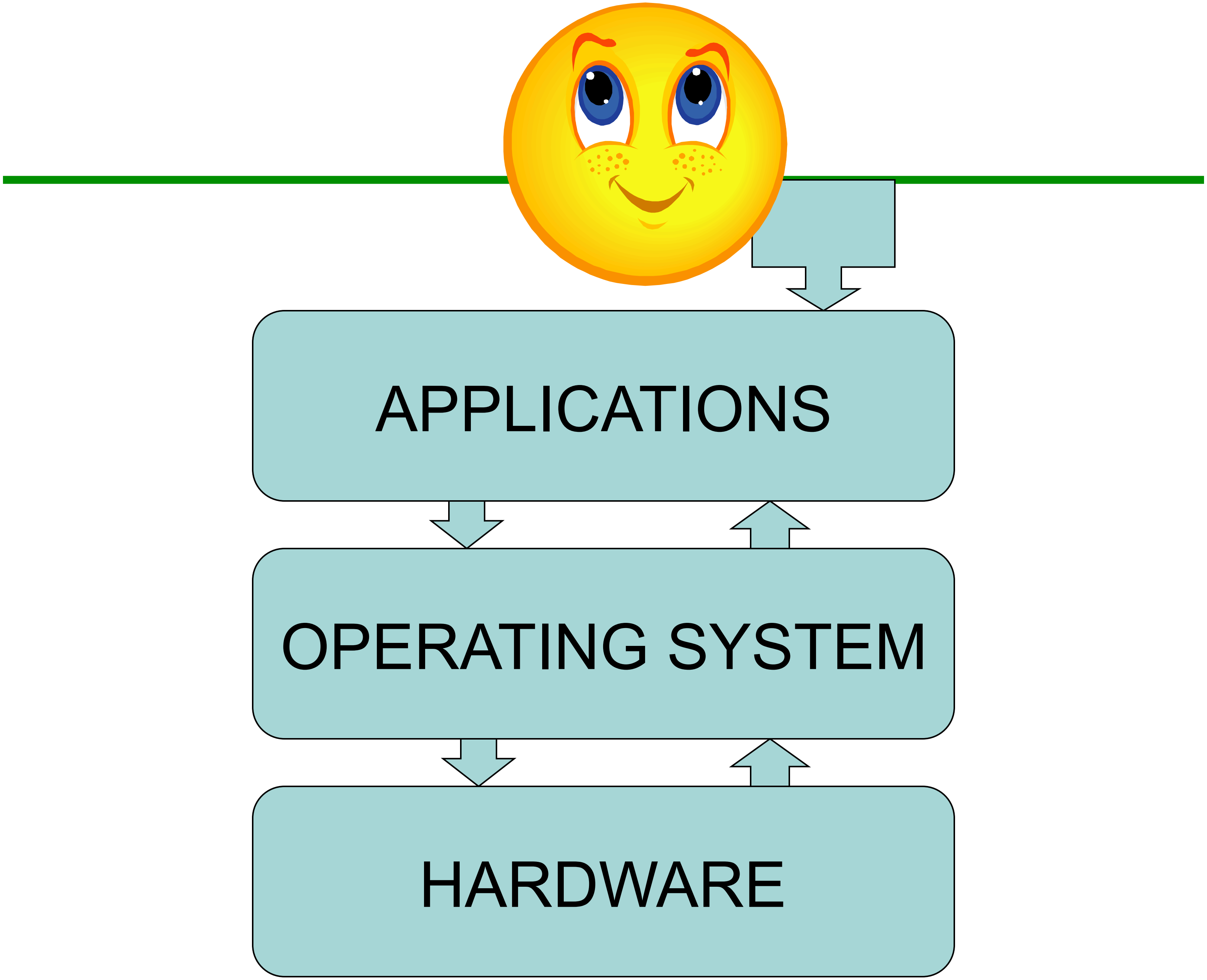
- Read the book, reference material, slides, man pages
- Code each example from class, don’t just read it, code it
- Do the labs and projects
- Do the practice exercises from lectures
- Start small, then add functionality
- Make mistakes and observe output
- Make sure you always understand why it did not work and why the solution works

# Terminology

- What’s a computer?
- What is hardware/software
- What’s an algorithm ?
- What’s a program?
- What’s an operating system?
- What’s a programming language ?
  - Machine language
  - Assembly language
  - High-level language

# Computer architecture





APPLICATIONS

OPERATING SYSTEM

HARDWARE

# OS Job

- Management of the processes and their access to resources
  - Memory
  - CPU access
  - I/O
  - Network
  - Other devices
- Interaction with the user
  - Graphic interface
  - Other devices

# Algorithm/Program

- **<u>Algorithm</u>**: procedure for solving a problem in finite steps
- **<u>Program</u>**: set of instructions to the CPU, stored in memory, read and executed by the CPU

# Machine and assembly language

- **Machine language** : binary information, specific to a CPU
  - How a CPU interprets data: e.g. how are memory addresses represented, how is an instruction coded, etc
  - This is the **binary or executable code**
- **Assembly language**: easier to write for people, using symbols, requires an assembler
  - Still need to think in terms of low level CPU steps
  - Still hardware-specific

# High-level language

- Closer to human language
- Needs a compiler to convert it to machine language
- One can write programs in many high-level languages for the same CPU
- More portable
- Examples: C, C++, C#, Objective C, Java, SmallTalk, also Cobol, Basic, Pascal …

# Readings for next lecture

K&R Chapter 1: A tutorial introduction