CS10003: Programming & Data Structures

Spring 2026

Dept. of Computer Science & Engineering

Course Materials

- Course webpage: http://cse.iitkgp.ac.in/pds/current/
  - Slides will be available in the link to the section-specific page from the above webpage

Books:

1. Programming with C

   Byron Gottfried

2. Programming in ANSI C

   E. Balaguruswamy

1. The C Programming Language

   Brian W Kernighan, Dennis M Ritchie

3. Data structures

   S. Lipschutz, Schaum's Outline Series

Evaluation

Mid-semester	30%
End-semester	50%
Two class tests	20%

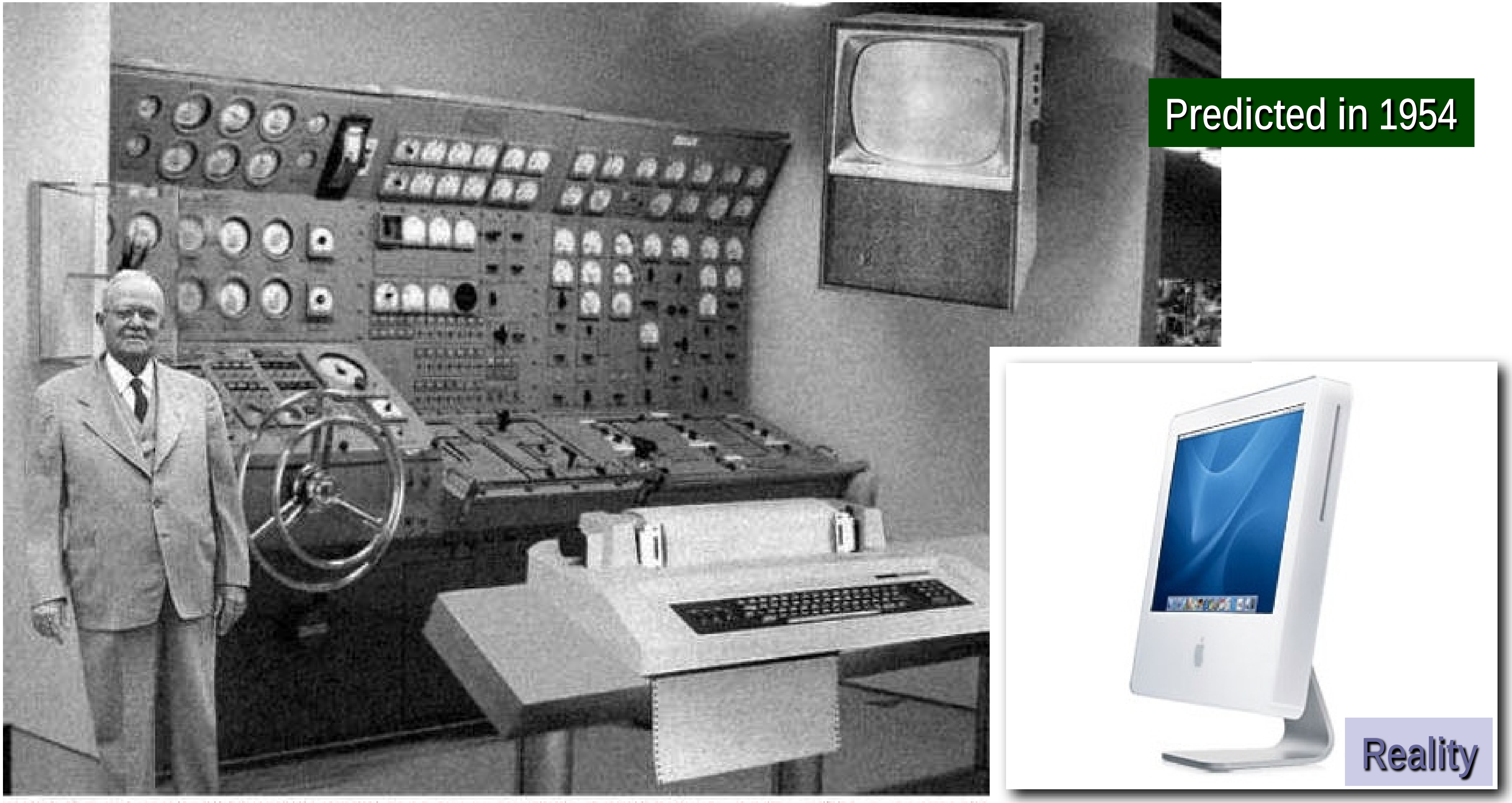
Important Dates

- Class Test 1: February 5, 2026 (19:00 – 20:00)
- Class Test 2: April 2, 2026 (19:00 – 20:00)

- Mid-semester and End-semester: As per Institute’s Academic Calendar

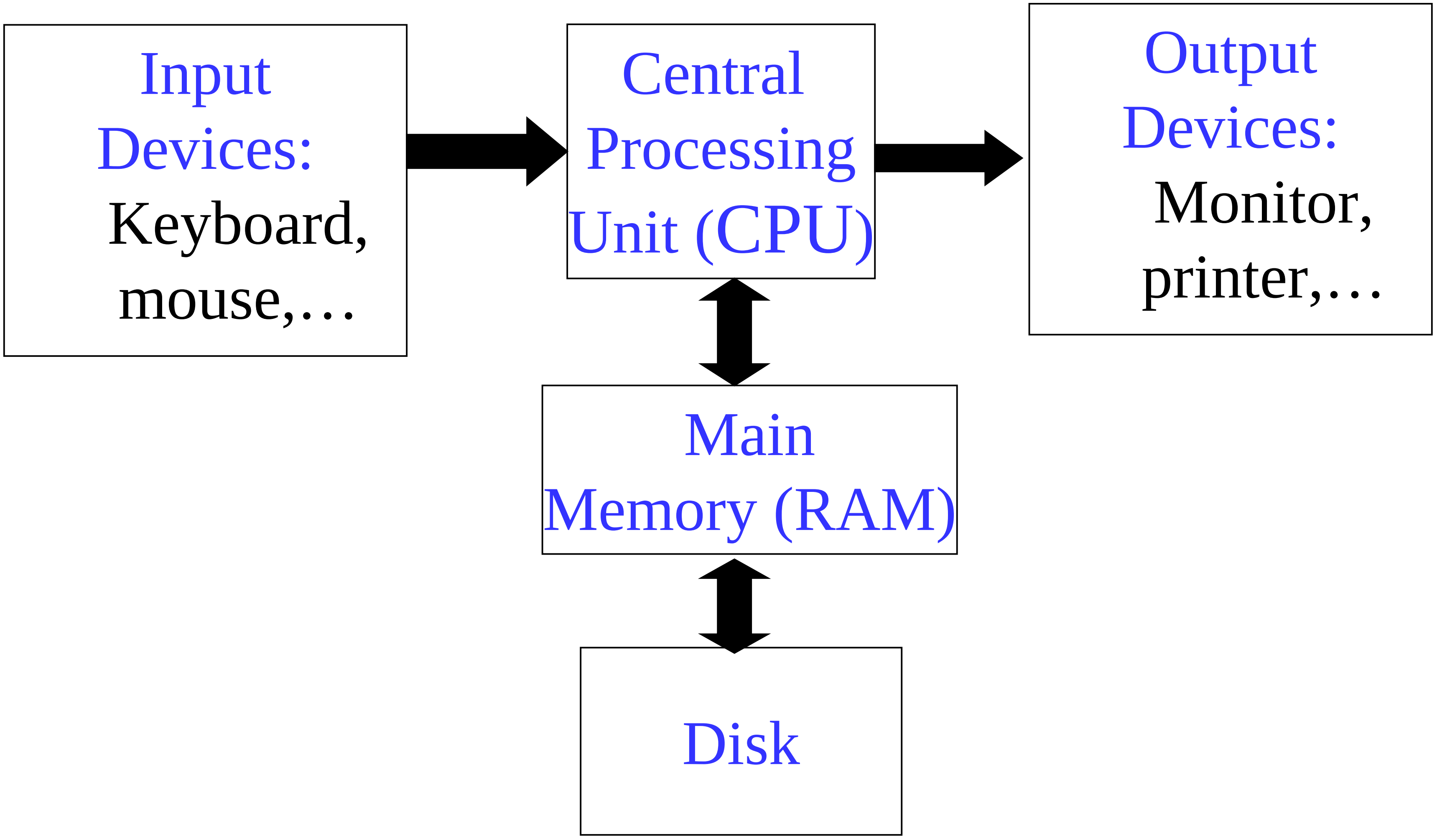
Introduction

Home Computer@2004: *Predicted versus Real*



Scientists from the RAND Corporation have created this model to illustrate how a "home computer" could look like in the year 2004. However the needed technology will not be economically feasible for the average home. Also the scientists readily admit that the computer will require not yet invented technology to actually work, but 50 years from now scientific progress is expected to solve these problems. With teletype interface and the Fortran language, the computer will be easy to use.

Basic Components in a Computer



Input Devices: Keyboard, mouse,…

Central Processing Unit (CPU)

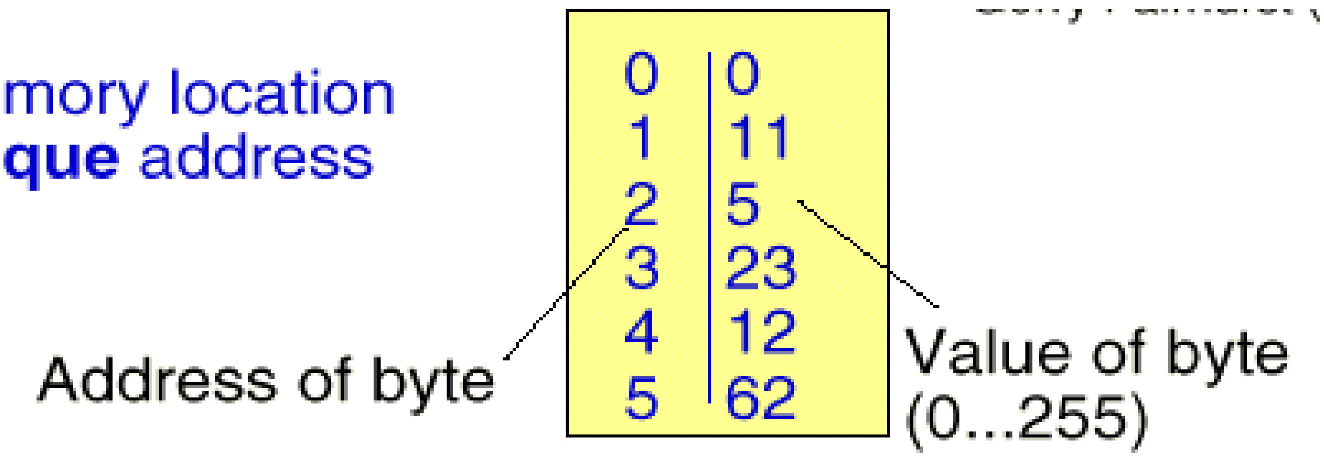
Output Devices: Monitor, printer,…

Main Memory (RAM)

Disk

Memory: Address and Values

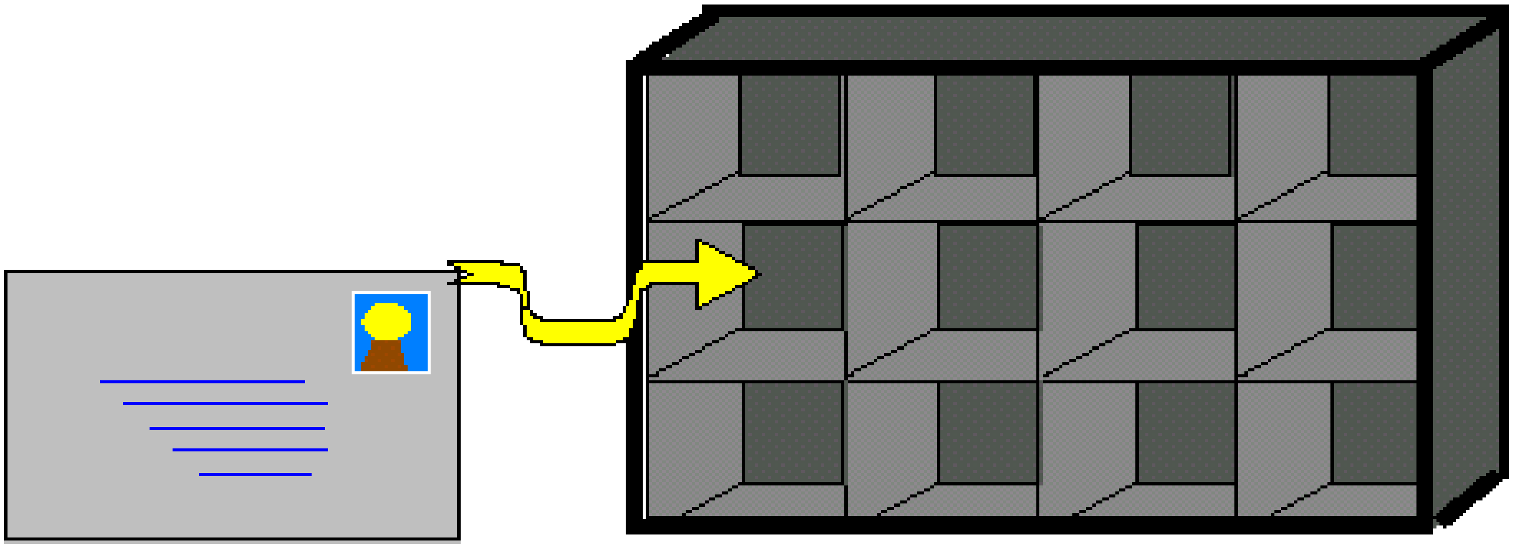
Every memory location has a **unique** address



Address	Value
0	0
1	11
2	5
3	23
4	12
5	62

Address of byte

Value of byte (0...255)



Programming and Software

- Computer needs to be programmed to do such tasks
- Programming is the process of writing instructions in a language that can be understood by the computer so that a desired task can be performed by it
- Program: sequence of instructions to do a task, computer processes the instructions sequentially one after the other
- Software: programs for doing tasks on computers

Contd.

- CPU understands machine language
  - Different strings of 0’s and 1’s only!!
  - Hard to remember and use
- Instruction set of a CPU
  - Mnemonic names for this strings

Instruction Set

- **Start**
- **Read M**
- **Write M**
- **Load Data, M**
- **Copy M1, M2**
- **Add M1, M2, M3**
- **Sub M1, M2, M3**
- **Compare M1, M2, M3**
- **Jump L**
- **J_Zero M, L**
- **Halt**

Instruction Set

- **Start**
- **Read M**
- **Write M**
- **Load Data, M**
- **Copy M1, M2**
- **Add M1, M2, M3**
- **Sub M1, M2, M3**
- **Compare M1, M2, M3**
- **Jump L**
- **J_Zero M, L**
- **Halt**

Program

0: Start

1: Load 33, 10

2: Load 24, 11

3: Add 10, 11, 12

4: Halt

Problems with programming using instruction sets directly

- Instruction sets of different types of CPUs are different
  - Need to write different programs for computers with different types of CPUs even to do the same thing
- Still hard to remember
- Solution: High level languages (C, C++, Java,…)
  - CPU neutral, one program for many
  - Compiler to convert from high-level program to low level program that CPU understands

High-Level Program

```
Variables x, y;
Begin
Read (x);
Read (y);
If (x = y) then Write (x)
                else  Write (y);
End.
```

High-Level Program

```
Variables x, y;
Begin
Read (x);
Read (y);
If (x = y) then Write (x)
            else  Write (y);
End.
```

Low-Level Program

```
0: Start
1: Compare 20, 21, 22
2: J_Zero 22, 5
3: Write 20
4: Jump 6
5: Write 21
6: Halt
```

Three steps in writing programs

Step 1: Write the program in a high-level language (in your case, C)

Step 2: Compile the program using a C compiler

Step 3: Run the program (as the computer to execute it)

What will you need in a program?

- Programs work on data
  - Need ways to handle different types of data (Data Types)
    - Integer, real, character, …..
  - Need place to store data (Variables)
- Need to give input to the program from outside and get the results out
  - Input/Output statements

What will you need in a program?

- Need ways to specify logic (how to get the results out from given input and other data)
  - Different types of expressions like arithmetic expressions, logical expressions, …
  - Different types of statements like assignment statements, conditional statements, loop statements, …
- Need to write good, modular code
  - Functions

Binary Representation

- Numbers are represented inside computers in the base-2 system (Binary Numbers)
  - Only two symbols/digits 0 and 1
  - Positional weights of digits: 2^0, 2^1, 2^2,…from right to left for integers
- Decimal number system we use is base-10
  - 10 digits, from 0 to 9, Positional weights 10^0, 10^1, 10^2,…from right to left for integers
  - Example: $723 = 3x10^0 + 2x10^1 + 7x10^2$

Binary Numbers

Dec	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000

Binary Numbers

Dec	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000

Binary to Decimal Conversion

101011 ➔ $1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 43$

$(101011)_2 = (43)_{10}$

111001 ➔ $1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 57$

$(111001)_2 = (57)_{10}$

10100 ➔ $1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 20$

$(10100)_2 = (20)_{10}$

Bits and Bytes

- Bit – a single 1 or 0
- Byte – 8 consecutive bits
  - 2 bytes = 16 bits
  - 4 bytes = 32 bits
- Max. integer that can represented
  - in 1 byte = 255 (=11111111)
  - In 4 bytes = 4294967295 (= 32 1's)
- No. of integers that can be represented in 1 byte = 256 (the integers 0, 1, 2, 3,….255)