

# ESC101N

## Fundamentals of Computing

Arnab Bhattacharya  
arnabb@iitk.ac.in

Indian Institute of Technology, Kanpur  
<http://www.iitk.ac.in/esc101/>

1<sup>st</sup> semester, 2010-11  
Tue, Wed, Fri 0800-0900 at L7

# Algorithms

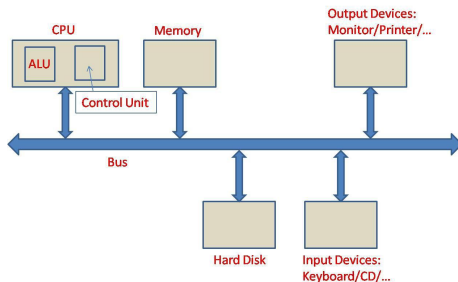
- An **algorithm** is a step-by-step procedure to achieve some objective
- Example: adding two numbers
- Input(s) and output

# Algorithms

- An **algorithm** is a step-by-step procedure to achieve some objective
- Example: adding two numbers
- Input(s) and output
- Properties of an algorithm
  - Precise
  - Finite number of steps
  - Steps must be ordered
  - Must terminate (either successfully or otherwise)
  - Must work for all inputs within a specified domain

- A machine that can carry out *any* computational task
- Properties
  - Precise: Will do exactly what you ask it to do – no more, no less
  - Error-free: Will not commit errors
  - Dumb: Has no intelligence to work on its own – requires detailed instructions
  - Faster than humans
  - Cannot perform everything that humans can

# Anatomy of a computer



- CPU: Central Processing Unit – the part where all computations actually take place
  - ALU: Arithmetic and Logic Unit – all arithmetic and logic operations are performed here
  - Control Unit – executes different instructions
- Memory – for storage
- Hard disk – bigger and slower storage
- Input and output devices – for communication to the world
- Bus – for transportation of data

# Binary system

- Computers can use only **binary format**: only 1's and 0's
- Binary number system is equivalent to decimal system (or any other number system)
- Number “6” in decimal system is equivalent to “110” in binary system
  - 110 is interpreted as  $0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 = 6$

# Execution in a computer

- To execute any operation in a computer, precise instructions are required
- Algorithms are set of instructions
- Algorithms are written as **programs** in a computer
- So, a program is nothing but a finite set of instructions

# Programming languages

- Instructions are in binary format as well
- Example: adding two numbers 2 and 3
  - 0011000
  - 0011010
  - 0011011
  - 0010100
- This is in **machine language**
- Above programs are hard for humans to write and understand
- But, computers understand only these



# Programming languages

- Instructions are in binary format as well
- Example: adding two numbers 2 and 3
  - 0011000
  - 0011010
  - 0011011
  - 0010100
- This is in **machine language**
- Above programs are hard for humans to write and understand
- But, computers understand only these
- Programming languages such as C alleviate these problems
- Example
  - `a = 2;`
  - `b = 3;`
  - `c = a + b;`
  - `output c;`
- Above instructions are converted (“**compiled**”) to machine language by a **compiler** so that they can be executed

# Writing C programs in Linux environment

- Create files with `.c` suffix – example, `addition.c`
- Use `gcc` to compile – `gcc addition.c`
- Will create `a.out` – run by typing `./a.out`
- Linux is an operating system that is (still) mostly command-based
- Some useful commands
  - `cd`: change directory
  - `mv`: move or rename
  - `rm`: delete
  - `mkdir`: create directory
  - `vim filename.c`: opens file `filename.c` for editing
  - `ls`: lists contents in a directory
  - `pwd`: shows the present working directory
- Practice to get familiar with and to learn more commands

# Execution of a program

