Summary of last class

Definition

Algorithms are finite sequence of *basic instructions* to carry out a particular task. The actual instructions depends on the model of computation that we are talking about.

Summary of last class

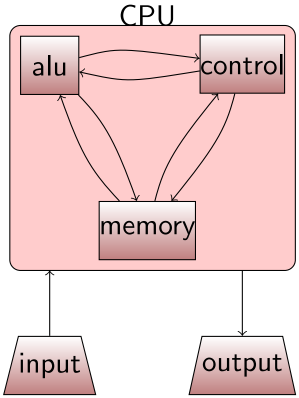
Definition

Algorithms are finite sequence of *basic instructions* to carry out a particular task. The actual instructions depends on the model of computation that we are talking about.

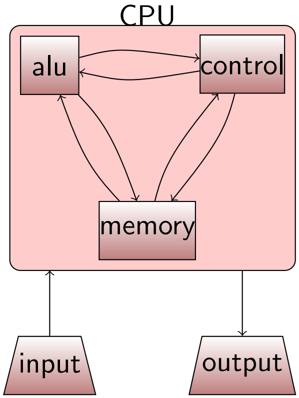
Examples

- Cooking recipies,
- Operational manual of a music system,
- Navigational directions,
- And of course computer programs.

von Neumann architecture

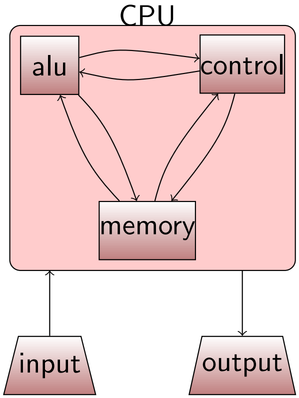


von Neumann architecture



Basic instructions

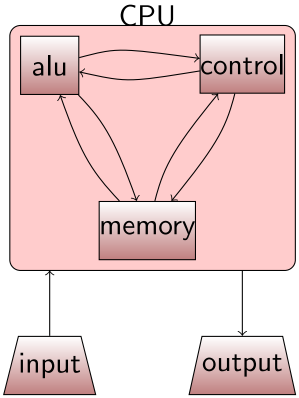
von Neumann architecture



Basic instructions

- Read/write data from memory/IO devices,

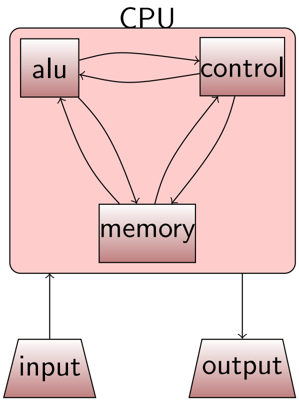
von Neumann architecture



Basic instructions

- Read/write data from memory/IO devices,
- Perform arithmetic logical operations on data,

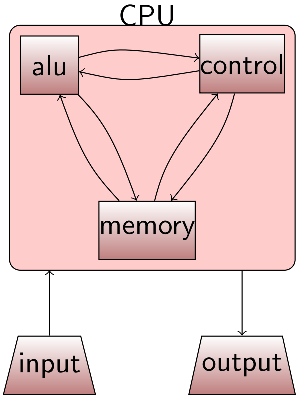
von Neumann architecture



Basic instructions

- Read/write data from memory/IO devices,
- Perform arithmetic logical operations on data,
- Read instructions from memory and interpret them.

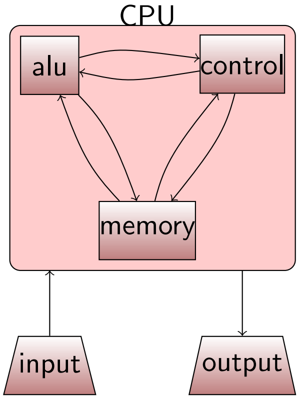
von Neumann architecture



Basic instructions

- Read/write data from memory/IO devices,
- Perform arithmetic logical operations on data,
- Read instructions from memory and interpret them.
- Proceed according to the instructions.

von Neumann architecture



Basic instructions

- Read/write data from memory/IO devices,
- Perform arithmetic logical operations on data,
- Read instructions from memory and interpret them.
- Proceed according to the instructions.

Programming language

One can think of a programming language as a way of expressing algorithms, or as a model of computation

Programming language

One can think of a programming language as a way of expressing algorithms, or as a model of computation

Kinds of programming languages

Programming language

One can think of a programming language as a way of expressing algorithms, or as a model of computation

Kinds of programming languages

- Low level languages; eg Assembly/Machine language

Programming language

One can think of a programming language as a way of expressing algorithms, or as a model of computation

Kinds of programming languages

- Low level languages; eg Assembly/Machine language
- High level language; eg Pascal, Python etc.

Programming language

One can think of a programming language as a way of expressing algorithms, or as a model of computation

Kinds of programming languages

- Low level languages; eg Assembly/Machine language
- High level language; eg Pascal, Python etc.
- Middle level language; eg C

Programming language

One can think of a programming language as a way of expressing algorithms, or as a model of computation

Kinds of programming languages

- Low level languages; eg Assembly/Machine language
- High level language; eg Pascal, Python etc.
- Middle level language; eg C

The higher the level of language the closer to human beings. The lower level language is closer to the machine.

Why high level language?

Why high level language?

Look at computers from Programmers perspective.

Why high level language?

Look at computers from Programmers perspective.

- Easier to write programs in.

Why high level language?

Look at computers from Programmers perspective.

- Easier to write programs in.
- Being generic, it is more portable.

Why then low level language?

Why high level language?

Look at computers from Programmers perspective.

- Easier to write programs in.
- Being generic, it is more portable.

Why then low level language?

Easier to realise on hardware.

```
#include<stdio.h>

int main()
{
  int n;
  int sum=0;
  scanf("%d",&n);

  while(n > 0)
  {
    sum = sum + n;
    n   = n - 1;
  }

  printf("%d",sum);
}
```

Assemblers, Compilers, Interpreters

Assemblers, Compilers, Interpreters

- Assemblers are the most basic translators. On Unix system the assembler is called `as`.

Assemblers, Compilers, Interpreters

- Assemblers are the most basic translators. On Unix system the assembler is called `as`.
- High level language requires compilers. C, Pascal

Assemblers, Compilers, Interpreters

- Assemblers are the most basic translators. On Unix system the assembler is called `as`.
- High level language requires compilers. C, Pascal
- Some languages are interpreted. eg Python, lisp etc

Other classification of programming languages

- Imperative programming language; eg C, Pascal, Python Glorified von Neumann architecture.
- Functional programming language; eg Haskell No states, only functions
- Object oriented languages; eg Smaltalk, Simula etc. Objects and messages to objects
- Logic based language; prolog Logic based techniques, Resolution.
- Scripting languages

What to expect in this course

We will study one imperative language C.
and a scripting language the shell.