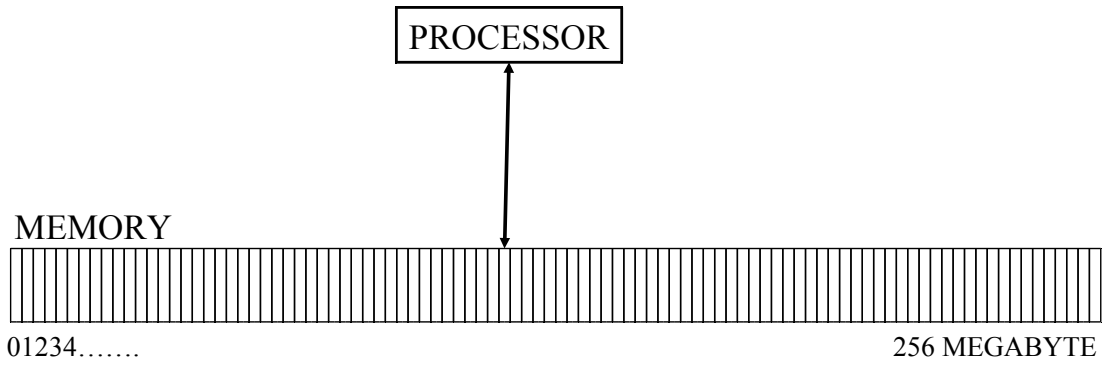
IC152 Lecture 3
Intro to Computation

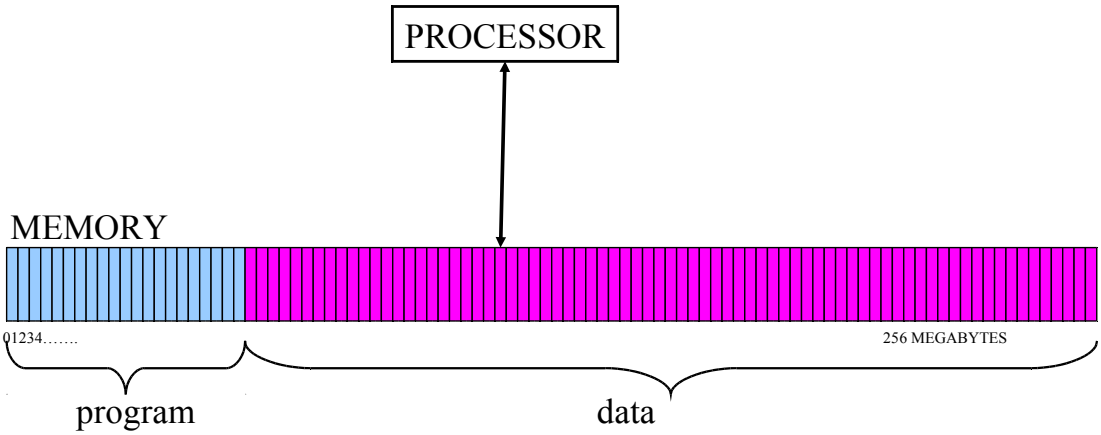
Timothy A. Gonsalves

The computing machine



The computer has a *processor* and a *memory*. The memory is a series of *locations* to store information.

The stored program *von Neumann* computer



- A *program* is a sequence of *instructions* for some task
- Most instructions operate on *data*
- Some instructions *control* the sequence of the instructions
- It is even possible to treat programs as data. By doing so a program could create another program, *or even modify itself*

Variables

- A memory location has an address and contains data
- The memory location is given the name of a variable for ease of use by the programmer

	100	104	108	112	116	<= address
data =>	23	6	138			
	len	ht	area			<= name

- Type of a variable defines the kind of data
  - e.g. integers (1, 175, 25649), or characters (‘a’, ‘M’, ‘n’, ‘i’, ‘d’)
- All data is stored as a sequence of bits, `0`’s and `1`’s, in a word of a fixed size. 1 byte = 8 bits

E.g. `01001101` = 77 `01001101` = 'M'

Type: int char

Program = Instructions + Data

- Program = sequence of machine instructions that operate on a set of variables
- Most instructions do some operation on a variable and store the result in another variable
- The instruction "X←X+Y" on integer variable:
  Take the integers stored in locations X and Y, add them, and store the sum back in X
- Other kinds of instructions E.g.
  "jump" to an instruction out of sequence
  terminate the program
  read from the keyboard

Programs

- A program is a sequence of instructions
- The processor works as follows:
  Step A: pick next instruction in the sequence
  Step B: get data for the instruction to operate upon
  Step C: execute instruction on data (or "jump")
  Step D: store results in designated location (variable)
  Step E: go to Step A
- Such programs are *imperative programs*

Programming paradigms

- *Imperative programs* are sequences of instructions. They are abstractions of how the *von Neumann machine* operates
  - Pascal, C, Fortran, Perl, Python, …
- *Object Oriented Programming (OOP)* models the problem as objects and interactions between them
  - Simula, CLOS, C++, Java, … (also Python)
- *Logic programs* use logical inference as the basis of computation
  - Prolog, …
- *Functional programs* take a mathematical approach of functions
  - LISP, ML, Haskell, …

A Limitation – Computer Arithmetic

- Number of digits that can be stored is limited
- Causes serious problems

Consider a computer that can store:

Sign, 3 digits and a decimal point

Eg: `212, -212, -21.2, -2.12, -.212`

More Examples

```
113. + -111. = 2.00
2.00 + 7.51  = 9.51
```

`-111. + 7.51 = -103.49` (exact arithmetic)

But our computer can store only 3 digits.

So it rounds `-103.49` to `-103`

Very important to know as a programmer. Why?

Why?

Consider `113. + -111. + 7.51`

To us addition is associative

(a+b)+c yields the same as a+(b+c)

For our 3-digit computer:

```
(113. + -111.) + 7.51 = 2.00 + 7.51 = 9.51
113. + (-111. + 7.51) = 113. + (-111. +8.)
                      = 113. - 103. = 10.0
```

The order of evaluation can affect the results

The Nature of Programming

Computer problem-solving can be summed up in one word -- it is *demanding*!

It is an intricate process requiring much thought, careful planning, logical precision, persistence, and attention to detail.

At the same time it can be a challenging, exciting, and satisfying experience with considerable room for personal creativity and expression.

If computer problem-solving is approached in this spirit then the chances of success are greatly amplified.

R.G. Dromey, *How to Solve it by Computer*, 1982

Conclusion

- Computer is fast – millions of operations per second
- Needs a precise sequence of instructions
- Solves well-defined problems
- We must learn to use its speed
- And manage its limitations

Data Science + Python:
write useful programs with a little effort!

Books

- J. Vanderplas: *Python Data Science Handbook*
- J. Grus: *Data Science from Scratch*
- M. Dawson: *Python Programming for the Absolute Beginner*
- R. G. Dromey: *How to Solve It By Computer*