

COSC1101 – PROGRAMMING FUNDAMENTALS CREDIT HOURS 4(3,1)

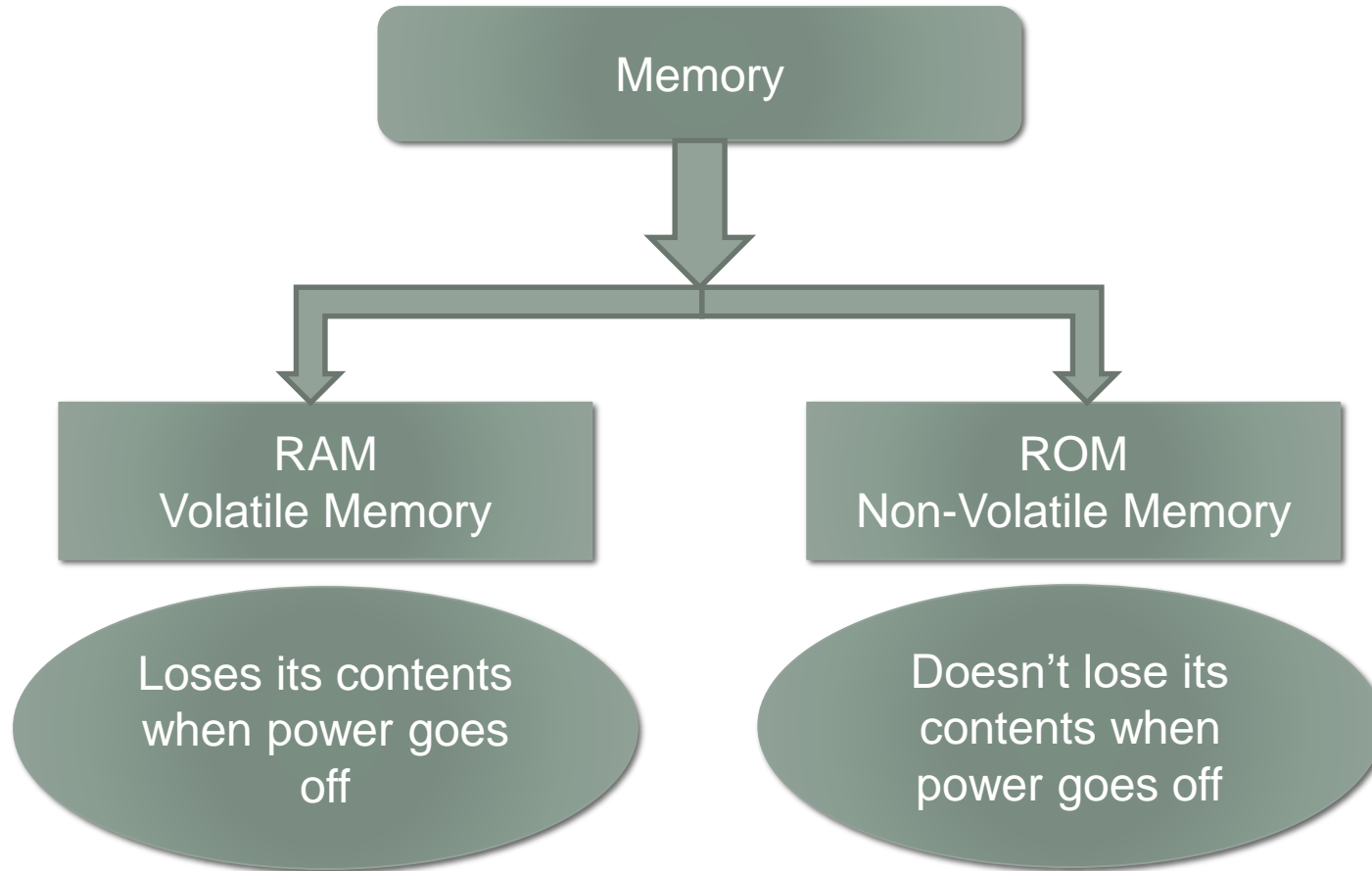
Instructor:

Maham Khan

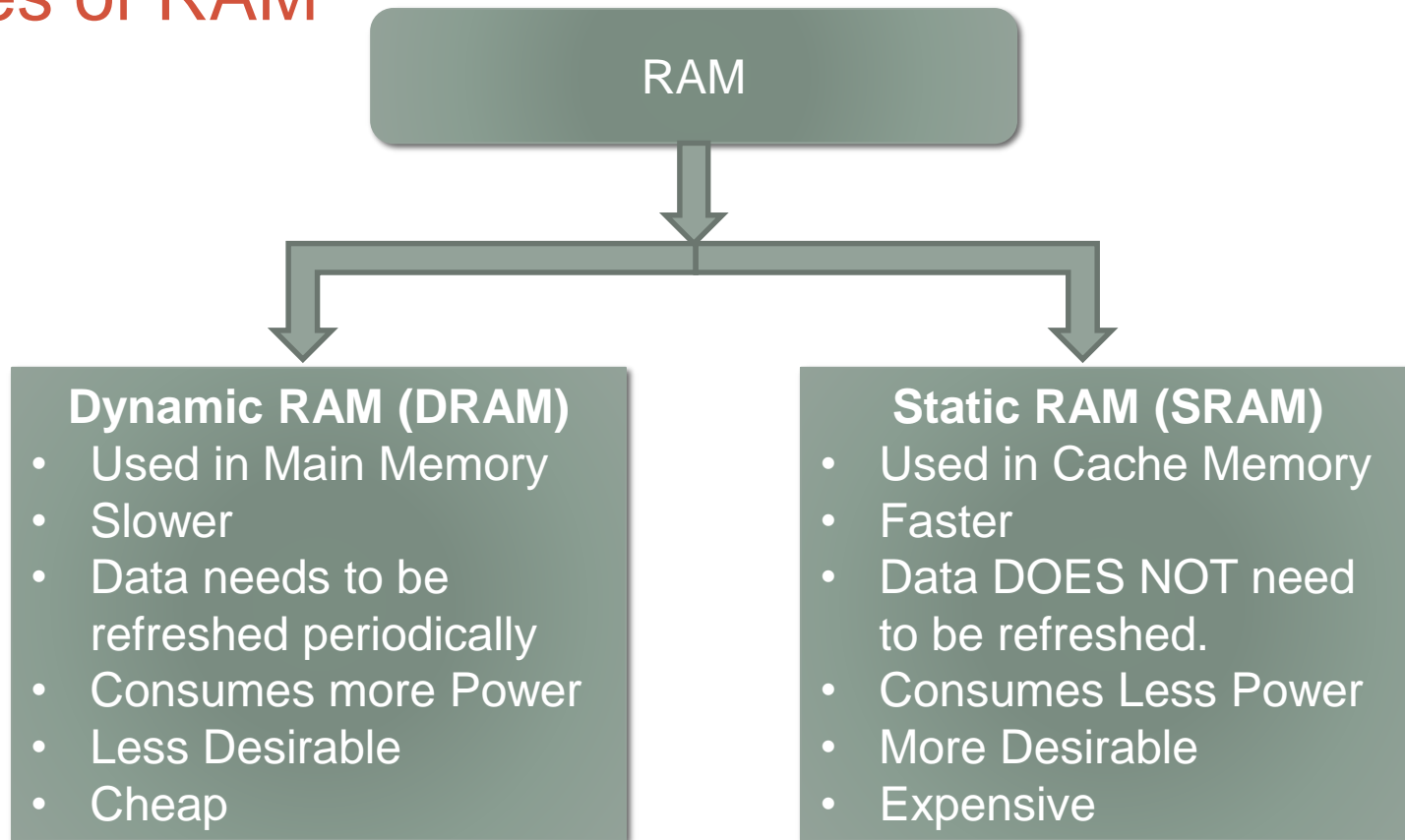
Additional References:

1. Reference book:
“Programming Languages and Methodologies”
by Robert J. Schalkoff Publisher Jones and Barrlett
- 2. http://en.wikipedia.org/wiki/Flash_memory

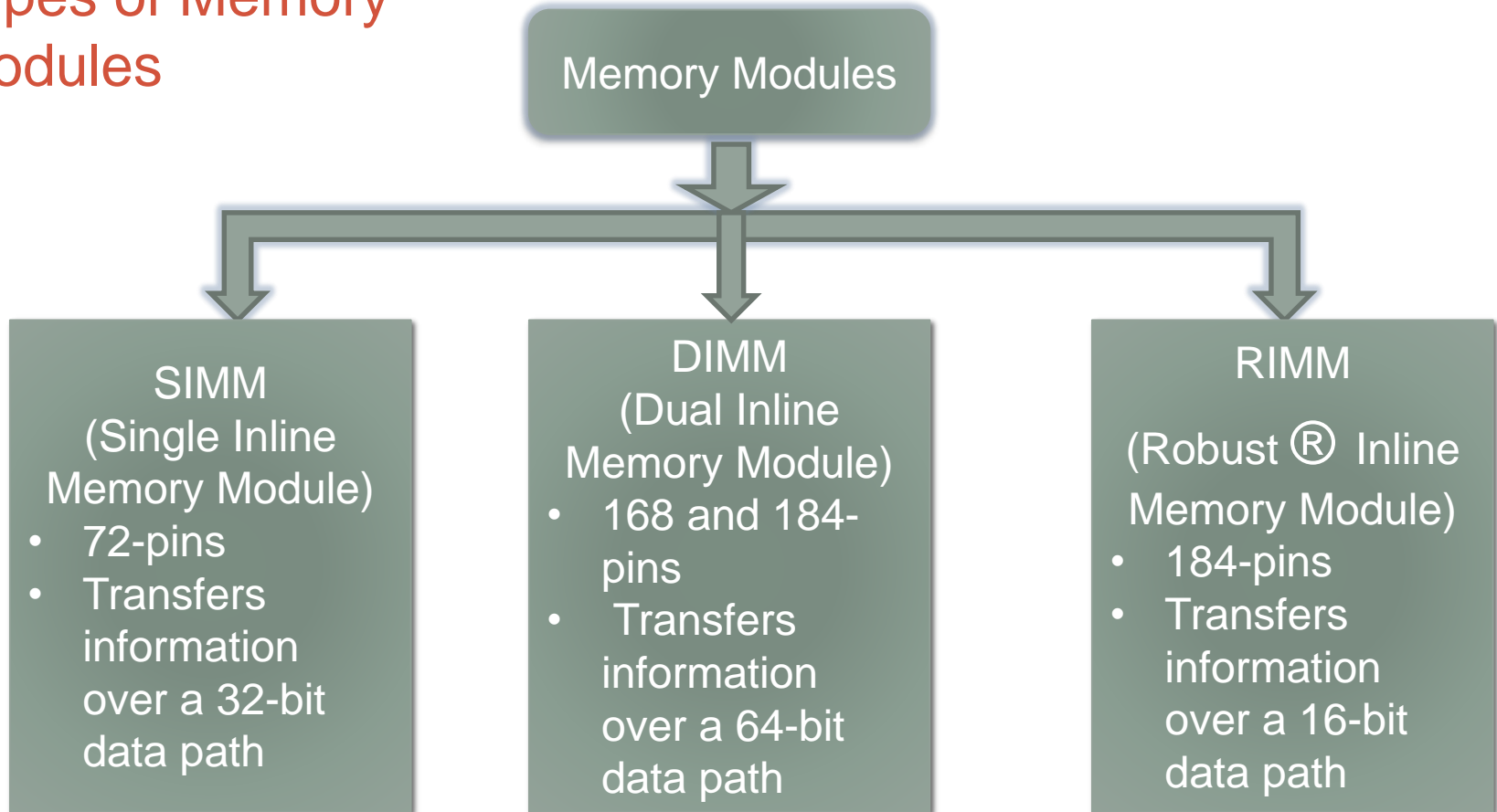
Types of Memory



Types of RAM



Types of Memory Modules



SIMM, DIMM and RIMM



Source:
Webanswers.com

SIMM



DIMM



RIMM

Cache Memory

- The cache is a small amount of high-speed memory, usually with a memory cycle time comparable to the time required by the CPU to fetch one instruction.
- The most frequently used instructions are kept in cache memory so that CPU can look there first. Saved time which was wasted by CPU in swapping instructions in and out of main memory.
- It can be located on Microprocessor chip or on a separate chip next to CPU.

Levels of Cache Memory

- L1 (Primary or Internal Cache):
 - Located directly on the processor chip
 - Runs at the same speed as the CPU
 - Small Capacity
- L2 (Secondary or External Cache):
 - Located directly on the processor chip
 - Runs at CPU speeds (or nearly so)
 - Larger capacity than L1 but slower than L1.
- L3:
 - Part of the system's motherboard.
 - Present in high-performance systems like servers.
 - Larger capacity than L2 but slower than L2.

Types of ROM

- ROM (Read-Only Memory)
- PROM (Programmable Read-Only Memory)
- EPROM (Erasable Programmable Read-Only Memory)
- EEPROM (Electrically Erasable PROM)
 - Flash EEPROM

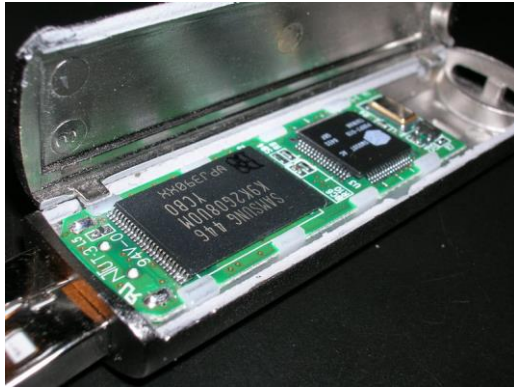
- ROM (Read-Only Memory)
 - Also known as Firmware
 - Manufactured with permanently written data.
 - Cannot be erased or written on by the user.
- PROM (Programmable Read-Only Memory)
 - one-time programmable non-volatile memory
 - manufactured blank
 - depending on the technology, PROM can be programmed at final test.
- EPROM (Erasable Programmable Read-Only Memory)
 - Multiple times erasable programmable non-volatile memory
 - It requires UV box to erase the EPROM

EEPROM (Electrically Erasable PROM)

- user-modifiable read-only memory (ROM)
- can be erased and reprogrammed (written to) repeatedly through the application of higher than normal electrical voltage generated externally or internally
- For Example: Flash Memory.

Flash Memory

- Non-volatile Electrically Erasable Programmable Read-Only Memory
- Examples: memory cards, USB flash drives, solid-state drives

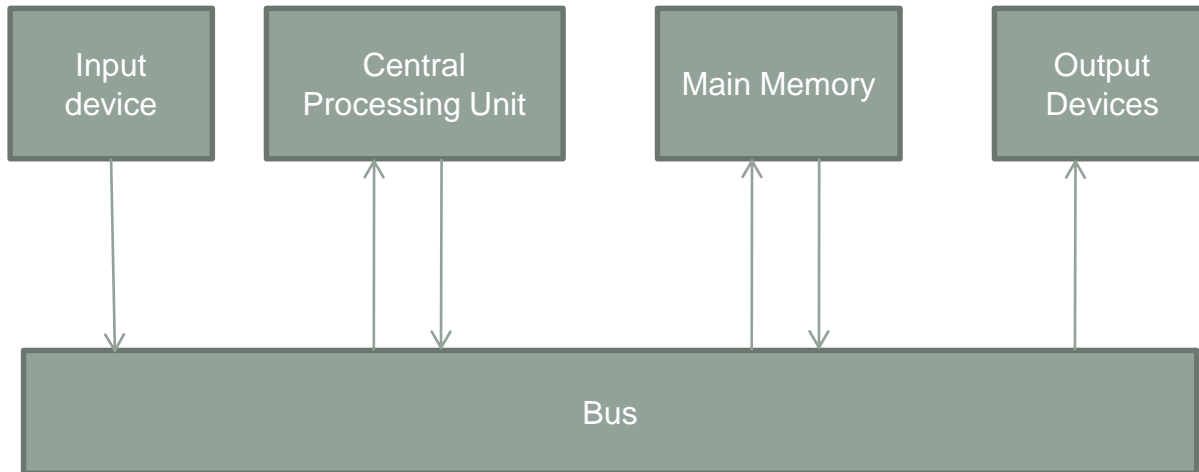
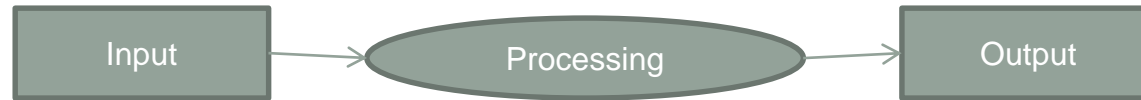


Source: http://en.wikipedia.org/wiki/Flash_memory

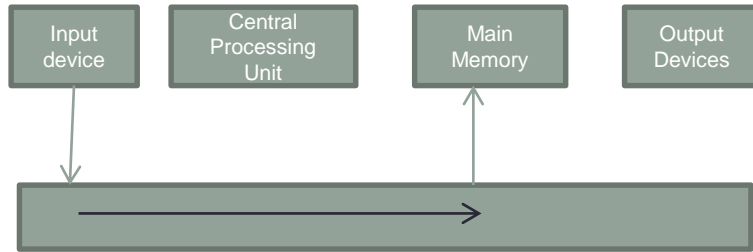
CMOS Memory (BIOS Memory)

- Small memory on PC motherboards that is used to store BIOS settings.
- Traditionally called as CMOS RAM
- It uses a low-power Complementary metal-oxide-semiconductor (CMOS) SRAM which is powered by a small battery to retain information when power goes off.
- Stores configuration information about the computer
 - Disk drives
 - Keyboard
 - Mouse
 - Monitor
 - Time/date.

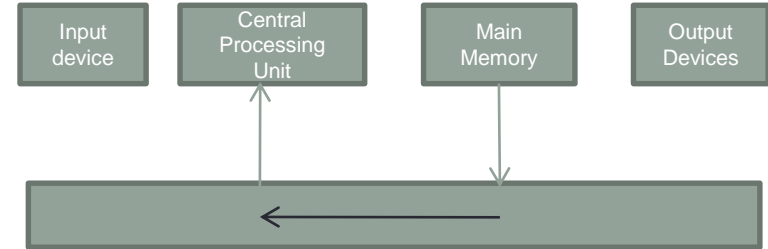
Processing



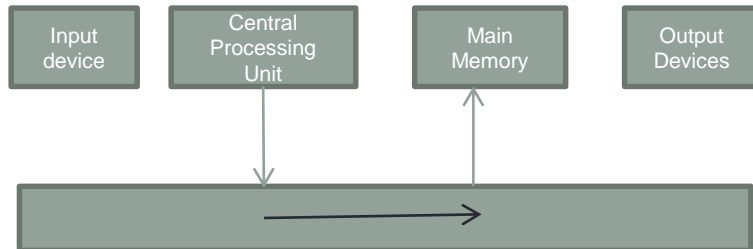
Data flow for a complete job



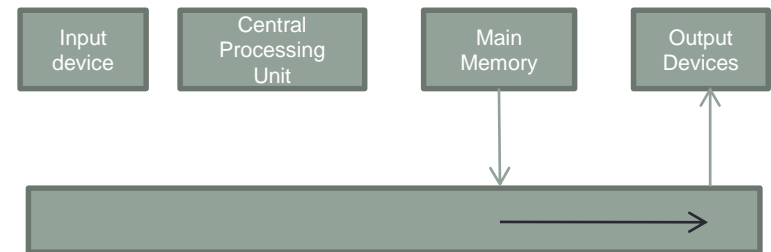
(a) First data flows from the input device into the main memory



(b) Next CPU brings the data into its registers for processing



(c) Then CPU sends processed data back to main memory



(d) Finally the results go to the output device

Steps (b) and (c) usually repeat many times

Programming Language

Programming Language:

- A programming Language is a notational system intended primarily to facilitate human-machine interaction.
- The notational is understood both by human and machine.
- The programming language has ***Syntax***, and language elements have ***Semantics***.

What is a program?

- A ***program*** is something that is produced using a programming Language.
- A ***program*** is a structured entity with ***Semantics***.

What is programming?

What is programming?

- ***Programming is a Science:***

Because it implement the algorithms describe by mathematics and science.

- ***Programming is a Skill:***

Because it requires design efforts.

- ***Programming is an Engineering:***

Because it requires a tradeoffs between program size, speed, time (required for development and debugging) and maintainability among many solutions.

- ***Programming is an Art***

It requires creativity and employ imagination.

Types of programming Languages

Levels/Generations of Programming Languages

- 1st Generation Programming language (1GL)
 - Machine Language: 0s or 1s
- 2nd Generation Programming language (2GL)
 - Assembly Language : Mnemonics
- 3rd Generation Programming language (3GL)
 - High-Level Languages ; (procedure oriented or Object Oriented)
- 4th Generation Programming language (4GL)
 - Very-High-Level Languages
- 5th Generation Programming Language
 - Natural Languages

Machine language (1GL)

- The lowest level of language.
- The language used to program the first-generation computers.
- The instructions in 1GL are made of binary numbers, represented by 1s and 0s.
- 1s and 0s correspond to the on and off states of electrical switches.
- Suitable for the understanding of the machine but very much difficult to interpret and learn by the human programmer.

Assembly language (2GL)

- Low-level language that allows a programmer to use abbreviations or easily remembered words instead of numbers.
- These Observations are called Mnemonics.
These Mnemonic are Opcode and Operands

For Example: ADD AX, BX
 MOV CX, AX
 INC CX

Op-code; ADD, MOV, INC

Operands AX, BX, CX

Assembly language (2GL)

- Programmer can write instructions faster but it is still not an easy language to learn.
- **Drawback:** The language is specific to a particular processor family and environment. (Machine Dependent Language)
- **Assembler** – A program that translates the assembly language program into machine language.

High Level languages (3GL)

- A High-Level Language is an English-like language.
- It is a refinement of a second-generation programming language.
- It allowed users to write in familiar notation, rather than numbers or abbreviations.
- Most High-level languages are not Machine Dependent.
- Translator for High-level languages is either a Compiler or an Interpreter.
- Examples of High-level languages:
 - FORTRON
 - COBOL
 - BASIC
 - C and C++

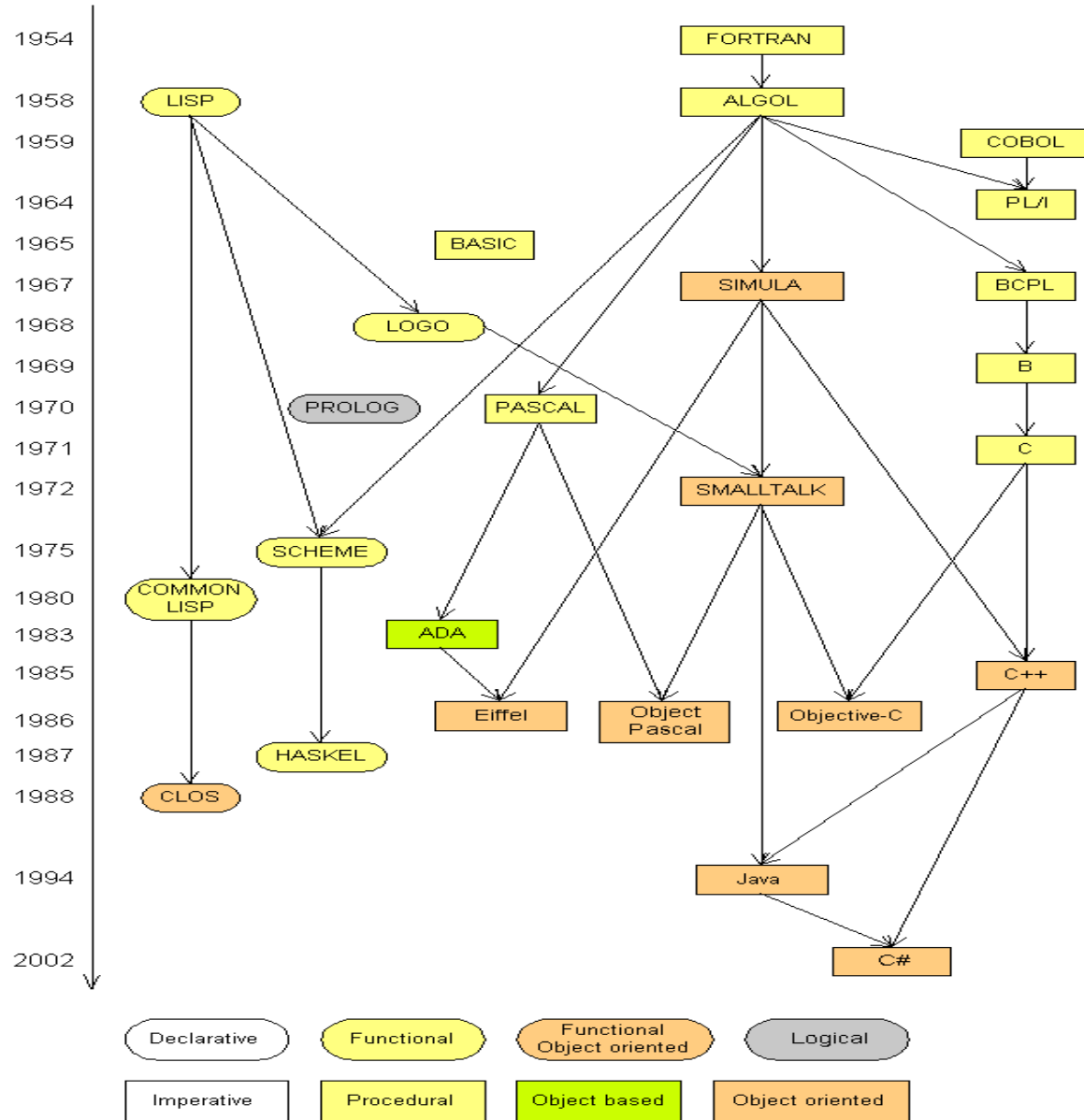
Very-High-Level Languages (4GL)

- 4GLs are much more user-oriented and allow programmers to develop programs with fewer commands compared with 3GLs.
- Non-Procedural Language; Programmers don't have to specify all the programming logic, only tell the computer what they want done.
- Saves a lot of time.
- 4GLs consist of report generators, query languages, application generators, and interactive database management system
- For example:
 - RPG III (Report Generator)
 - SQL (Structured Query Language)
 - NOMAD and FOCUS (DBMS)

Natural Languages (5GL)

- Two types
 - Ordinary Human Languages; like English.
 - Programming language that use human language to give people a more natural connection with computers.
- 5GLs are designed to make the computer solve a given problem without the programmer.
- Natural languages are part of the field of study known as *Artificial Intelligence*.
 - *Develop machines to emulate human-like qualities such as learning, reasoning, communicating, seeing and hearing.*

Evolution of Programming



Programming Paradigms

A number of programming paradigms are:

- Procedural/Imperative Programming
- Functional Programming
- Declarative Programming
- Object Oriented Programming
- Event driven Programming
- Parallel Programming

Procedural or Imperative:

- Imperative programs emphasize “tell what to do”
- When we say “ Do this, then do this, then do this, and if xx, do this, otherwise do this” This is imperative programming:
- They focus on evaluating expressions and storing results in a variable.
- The most common imperative language consists of statements such as:

`a = 10;`

`b = 5;`

`c = a + b;`

Procedural or Imperative: ... Contd

Example of imperative languages are:

- Assembly language
- COBOL
- Pascal
- C and C++

Is there any Best Programming Language?

- Programming Language is probably used most efficient if it is well suited For a specific task.
- For example
 - Business applications are often written in **COBOL**.
 - Beginners to programming use **BASIC**.
 - Scientific programming is often undertaken with either **FORTON**, **PASCAL** or **C**.

Language Evaluation Criteria

- **Readability:** measure programmer ease in reading source code
- **Orthogonality:** include context sensitive restrictions.
- **Applicability:** this is best suited as “use the right tool for the job”
- **Writ ability:** includes simplicity and orthogonality and support for the abstraction
- **Reliability:** include type checking and inspection handling
- **Cost:** includes learning and writing cost, productivity, compilation cost, execution cost, debugging cost etc.
- **Other:** flexibility of control statements and availability of data structures

History of C Language

- A general-purpose computer programming language.
- Developed between 1969 and 1973 by Dennis Ritchie at the Bell Telephone Laboratories for use with the UNIX operating system.
- Descendant of Ken Thompson's language "B"
- The first C programming language standard was published in 1978 as "The C Programming Language" by Kernighan and Ritchie (K&R).
- "ANSI C" was the second C programming language standard published in 1989.
- ISO approved "ANSI C" in 1990.
- The latest stable release is C11.