Reminders

Bonus Assignment #1

1. **In Logo, create a program that will write your first name.**
   - Each letter in your name should be a function.
   - Be as creative as you can be.

2. **The deadline for this bonus assignment is on July 7, 2009, Tuesday.**

3. **Submit the following (which are stapled) @ C-114.**
   - Source code
   - Actual Logo screen that contains your name.

4. **What's the benefit?** ☺

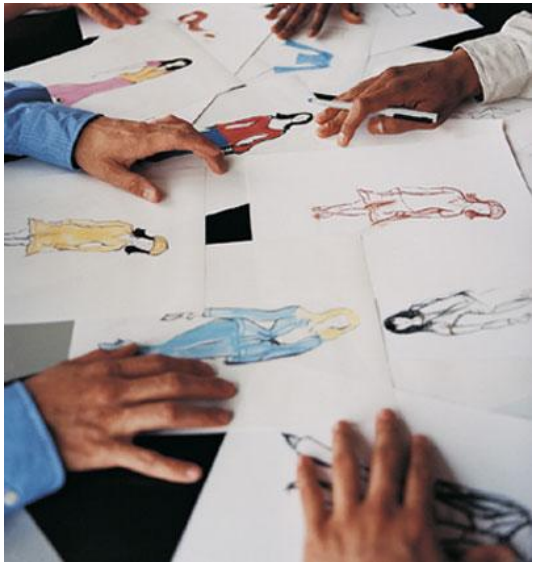
Be DM Zone's contact to access:

DMZONE.MULTIPLY.COM

124 Stuff and Others

Chapter 2: Language Design Issues

CMSC 124, 1st Semester, AY 2009-10



Something to Ponder

What has/have influenced the design of programming languages?

Something to Ponder

Chapter 2: Language Design Issues

Topics Covered

1. **Influences on Language Design:**
   - Computer Architecture
   - Program Design Methodologies

2. **Impact of Computer Architectures**

Chapter 2: Language Design Issues

Programming Methodologies

50's, 60's: Machine Efficiency

Late 60's: People Efficiency

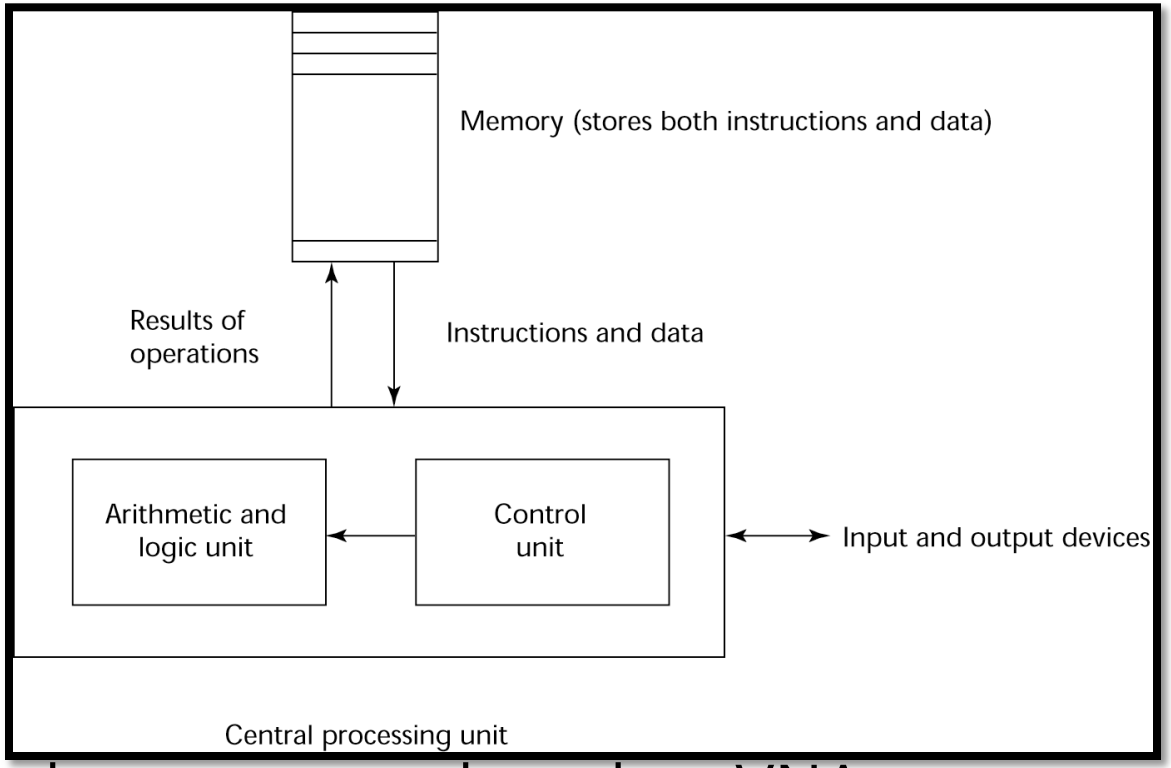
- Better control structures
- Structured programming
- Top-down design
- Stepwise refinement

Late 70's: Process-Oriented, Data-Oriented

Middle 80's: OOP

Chapter 2: Language Design Issues

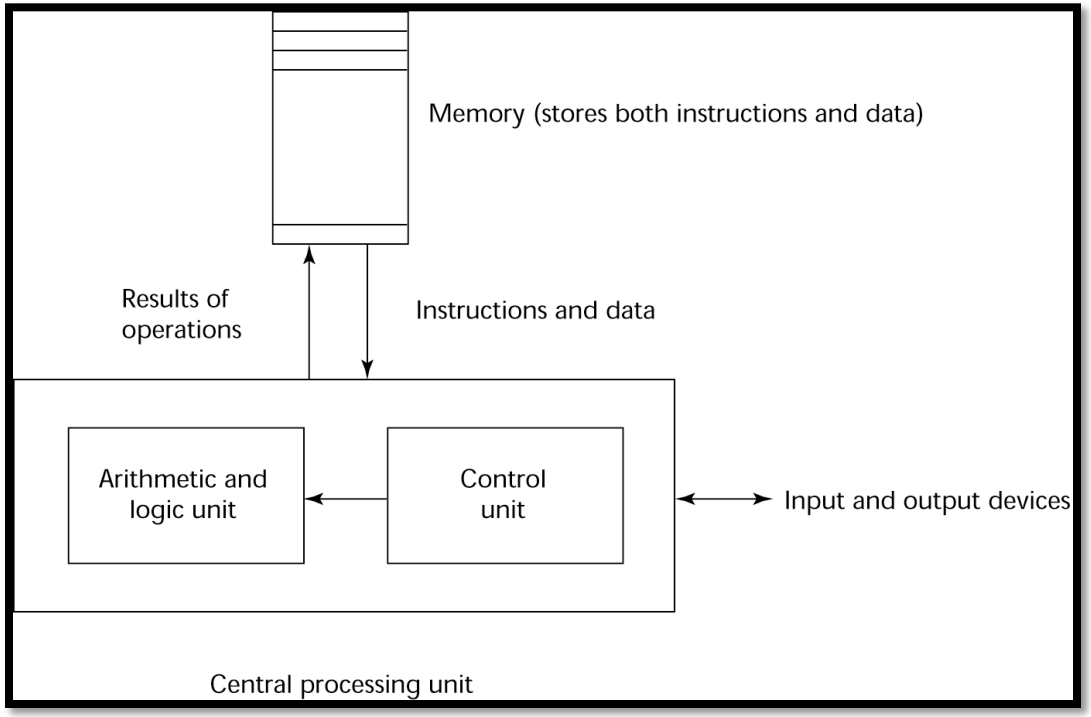
Computer Architecture : Von Neumann



- Imperative languages are based on VNA.
- Data and program are stored in same memory.
- Memory is separate from CPU.
- Instructions and data are piped from memory to CPU.

Chapter 2: Language Design Issues

Computer Architecture : Von Neumann



- Basis for imperative languages:
  - Variables model memory cells.
  - Assignment statements model piping.

Chapter 2: Language Design Issues

Impact of Computer Architectures

Operation of a Computer

- Computer Hardware
- Firmware Computers
- Translators and Software Simulation

Virtual Computers

- Virtual Computers and Language Implementations
- Hierarchies of Virtual Machines

Chapter 2: Language Design Issues

Operation of a Computer

Computer

- An integrated set of algorithms and data structures capable of storing and executing programs.
- Programs are executed on a hardware computer or a virtual computer.

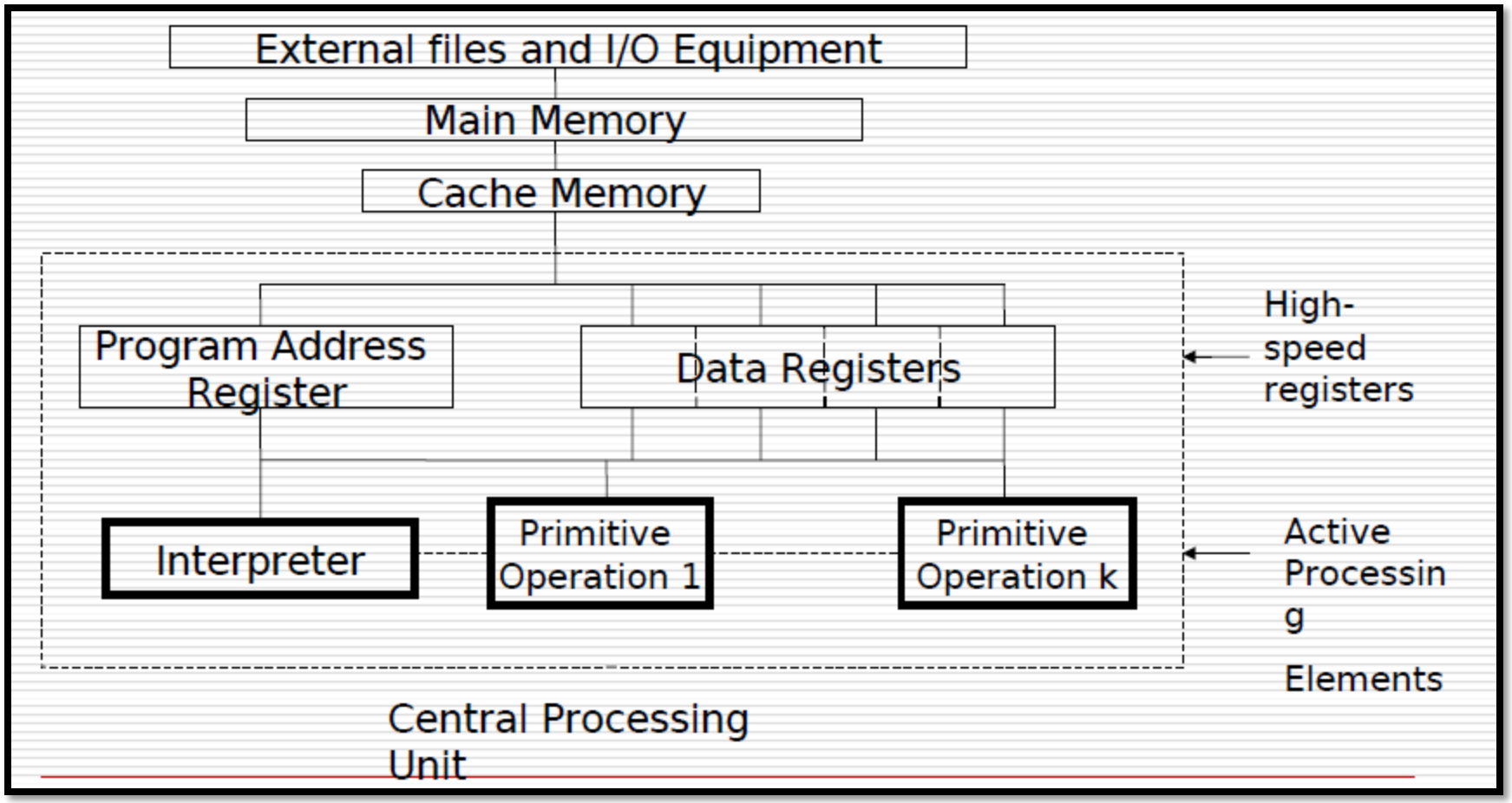
Chapter 2: Language Design Issues

Major Components of a Computer

1. Data
2. Primitive Operations
3. Sequence Controls
4. Data Access
5. Storage Management
6. Operating Environment

Chapter 2: Language Design Issues

Organization of a Conventional Computer



Chapter 2: Language Design Issues

Computer Hardware

1. **Data**
   - **Data-Storages**
     - Main-memory
     - Registers
     - Cache
     - External files
   - **Primitive Data Types**
     - Int
     - Reals
     - Fixed-length char strings
     - Programs

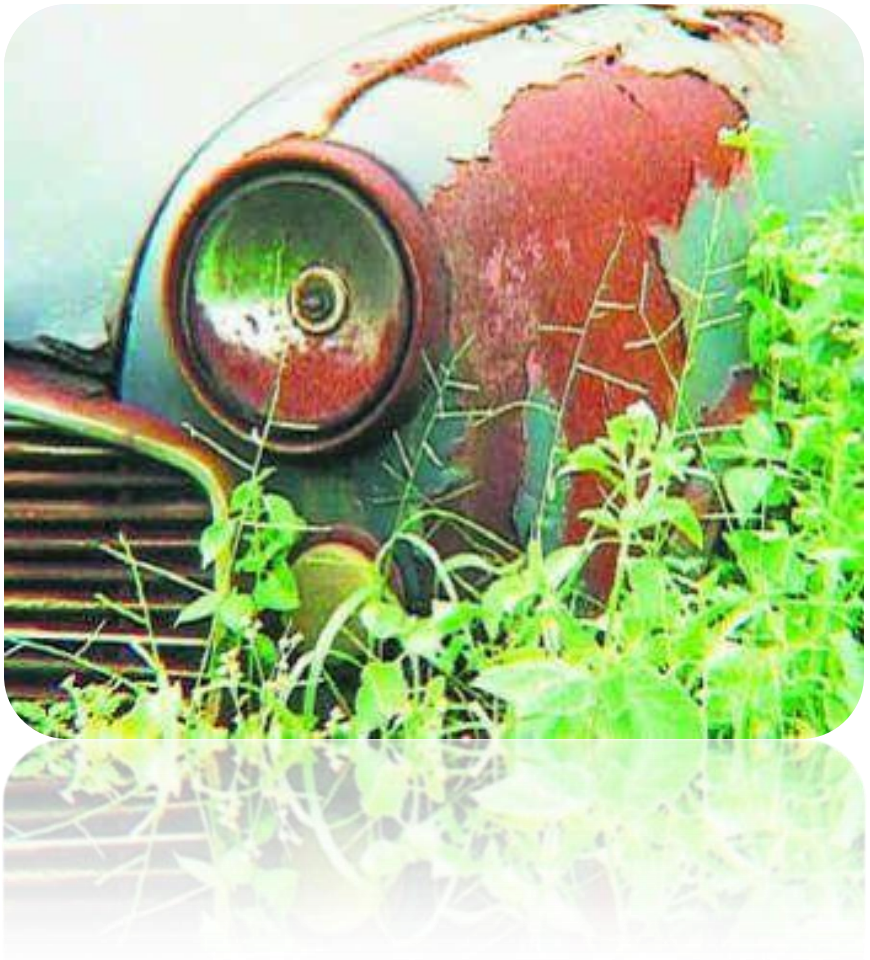
2. **Operations**
   - Arithmetic primitives
   - Testing properties of data items
   - Access/modify Data
   - I/O
   - Sequence control

Chapter 2: Language Design Issues

Computer Hardware

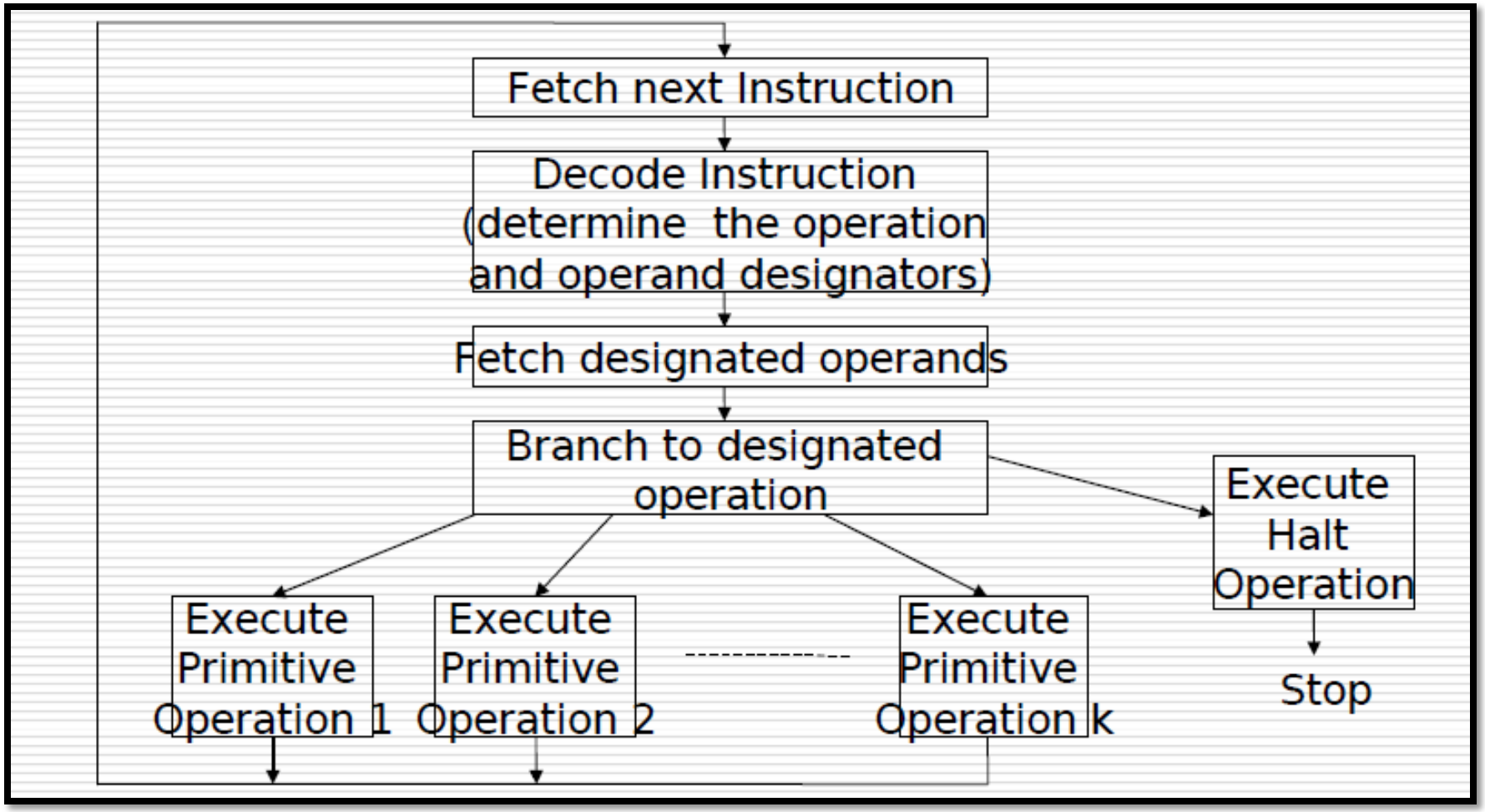
3. **Sequence Control**
   - Program Address Register/Location Counter
   - Fetch-Decode-Execute

4. **Data Access**
   - Operands –usually stored in registers
   - Store results of operations
   - Associate addresses with memory locations



Chapter 2: Language Design Issues

Fetch-Decode-Execute Cycle



Chapter 2: Language Design Issues

Computer Hardware

5. **Storage Management**
   - Single-user
   - Multiprogramming
   - Paging/Dynamic Relocation
   - Cache Memory

6. **Operating Environment**
   - Peripheral storage
   - I/O Devices

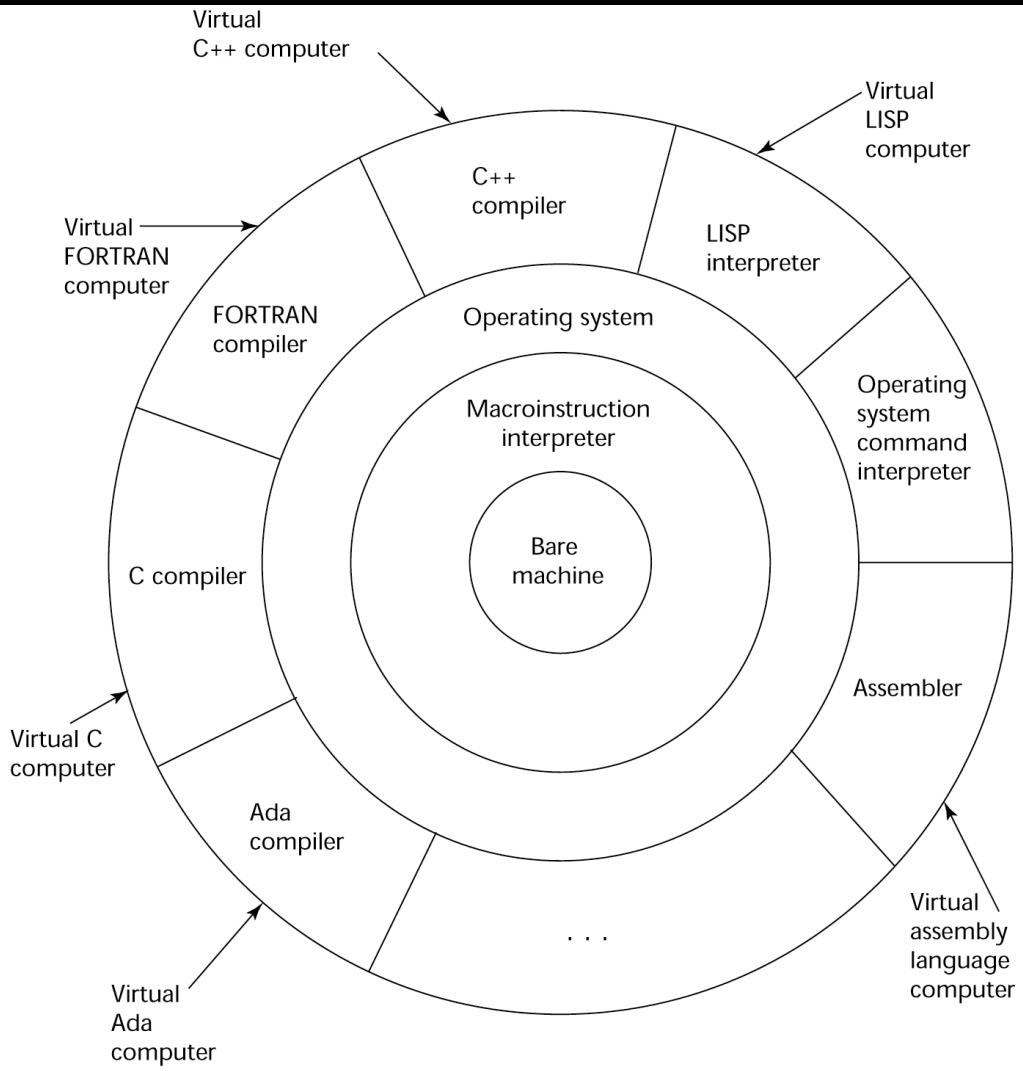
Chapter 2: Language Design Issues

Firmware Computers

- An alternative to hardware realization of a computer.
- Simulated by a microprogram running on a special microprogrammable hardware computer.
- Extremely low-level set micro-instructions.
- Simple computers maybe programmed to look like any broad range computers.
- Microprogram simulates the operation of the desired computer.

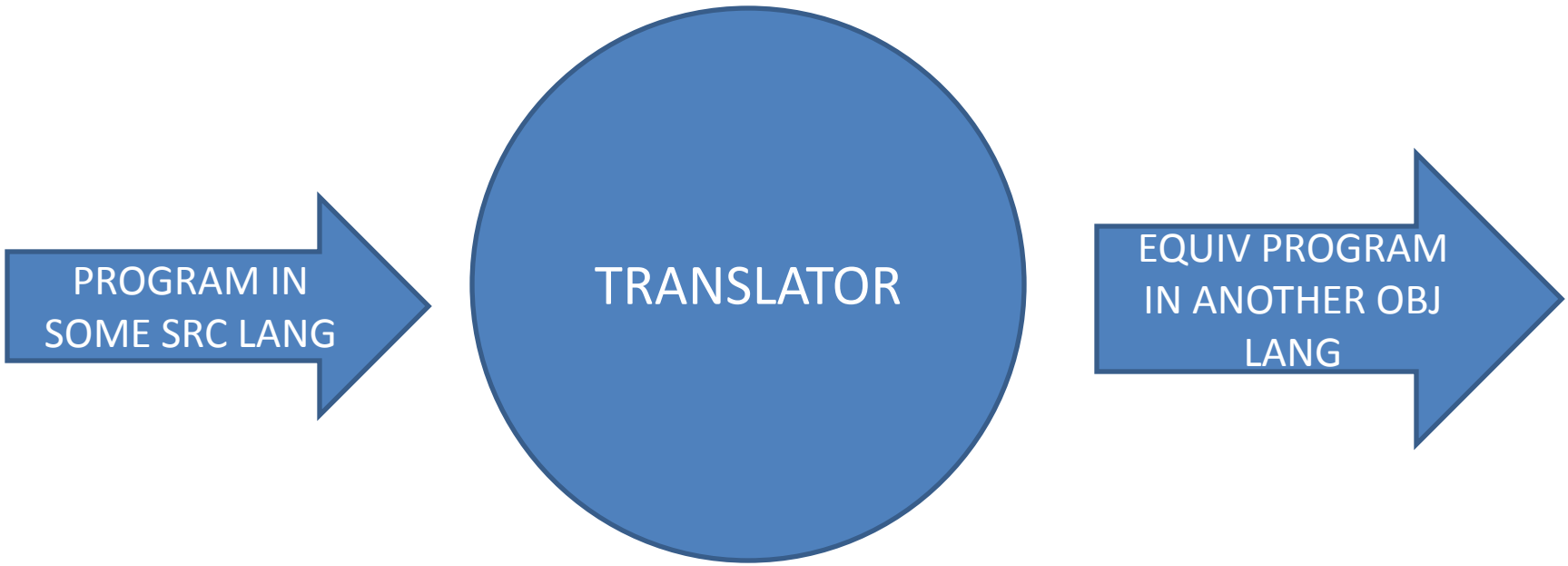
Chapter 2: Language Design Issues

Layered View of a Computer



Chapter 2b: Language Design Issues

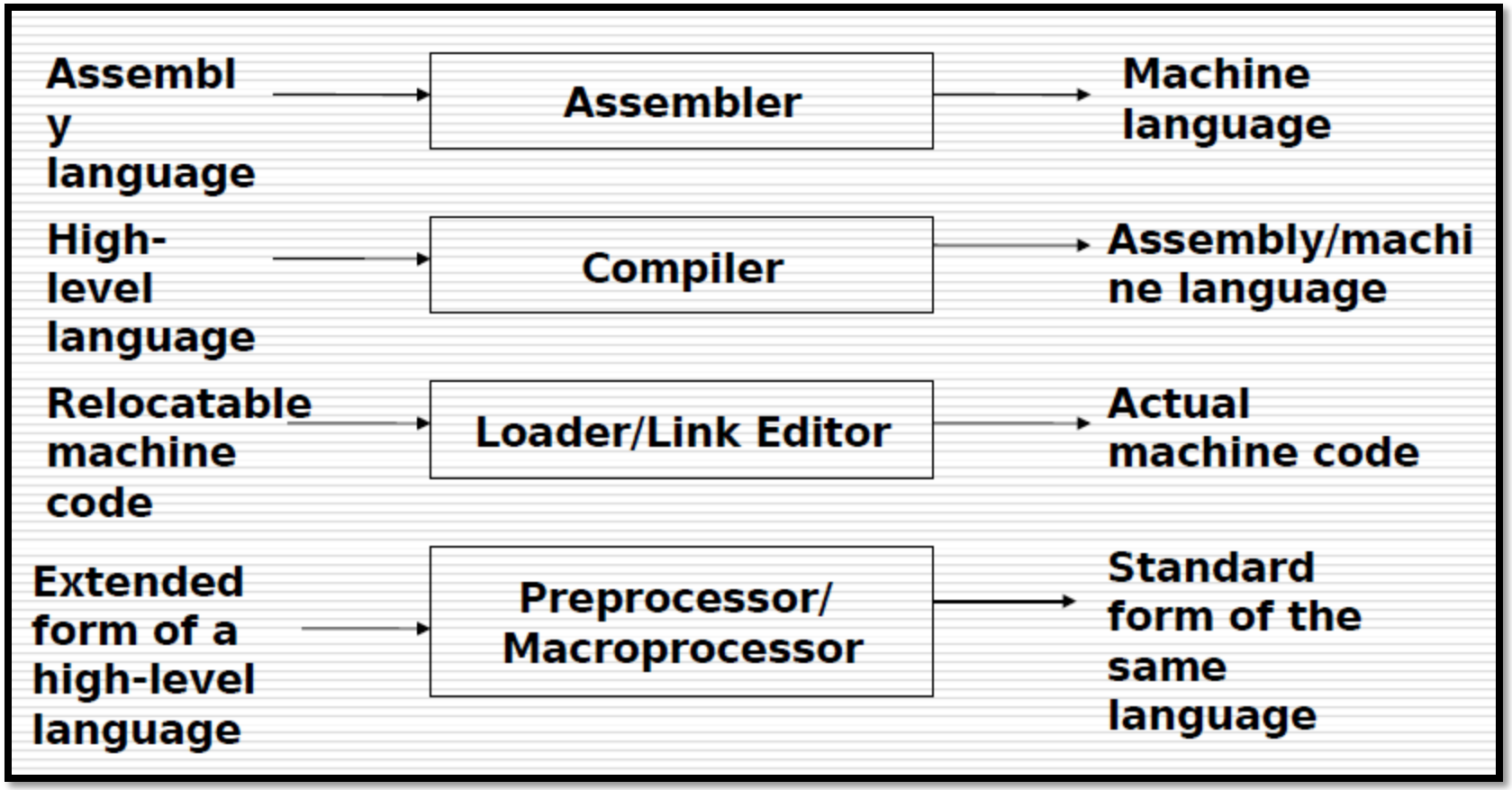
Translators



Why do we need to use a Translator?

Chapter 2b: Language Design Issues

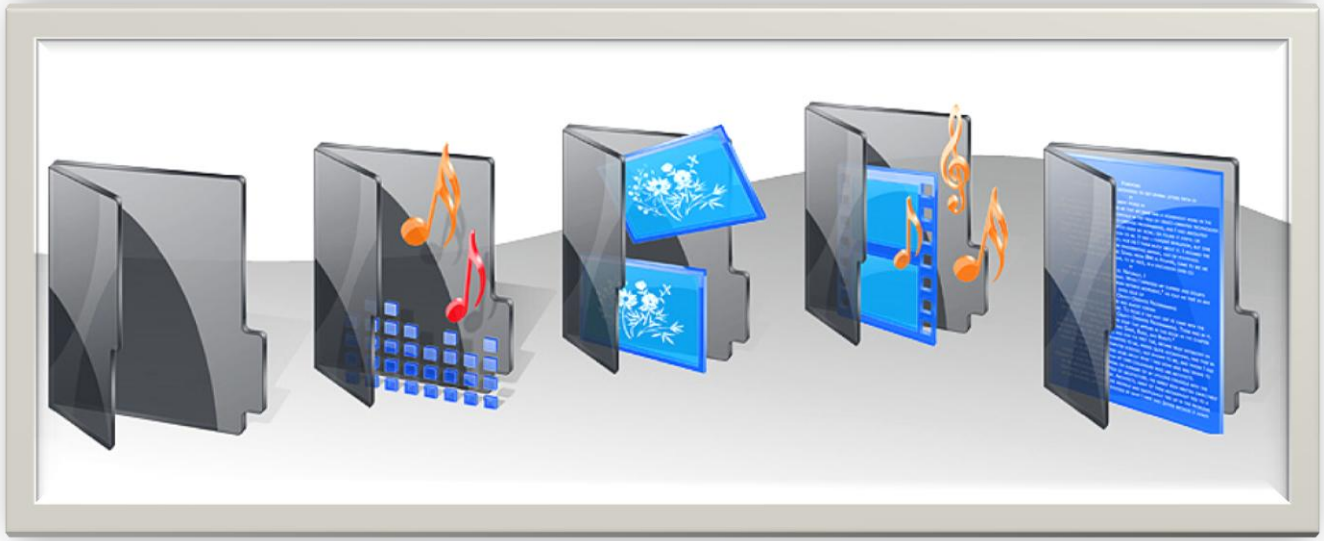
Translators



Chapter 2b: Language Design Issues

Methods of Implementation

1. **Compilation**
   - The high-level language is translated into another language usually assembly or machine
   - The translator is called a compiler.



Chapter 2b: Language Design Issues

Compilation Approaches

1. **High-level -> Machine**
   - Compiler written to translate a high-level to machine.

2. **High-level -> Assembly -> Machine**
   - Compiler written to translate high-level to assembly.
   - The assembly language further translated to machine language using the assembler provided in every computer.

3. **High-level -> Intermediate -> Assembly or Machine**
   - 'Half-compiler' written to translate high-level to intermediate.
   - Then, another half-compiler written to translate from intermediate to assembly or machine.

Chapter 2b: Language Design Issues

Notes on Compilation

- The object of compilation is to produce the machine language equivalent of the high-level language.
- It is the _____ which is ultimately executed.
- Total work is classified:
  - Compile-time
  - Run-time

Chapter 2b: Language Design Issues

Methods of Implementation

2. Interpretation

- Method that simulates, through a program running on another host computer, a computer whose machine language is the high-level language.
- It is as if the instructions of the high-level language is executed directly.
- What do you call the 'one' which carries the interpretation?

Something to Ponder

Compilation is a preferred method of implementation than interpretation.

Something to Ponder

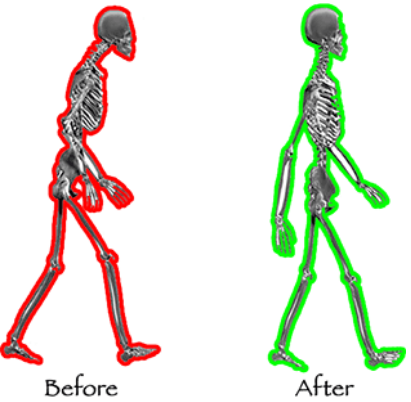
Chapter 2b: Language Design Issues

Virtual Machines/Computers

Computers can be constructed either by hardware realization, firmware realization, software simulation or by a combination.

BEFORE...

- Build program to use hardware efficiently.
- Often use of machine language for efficiency

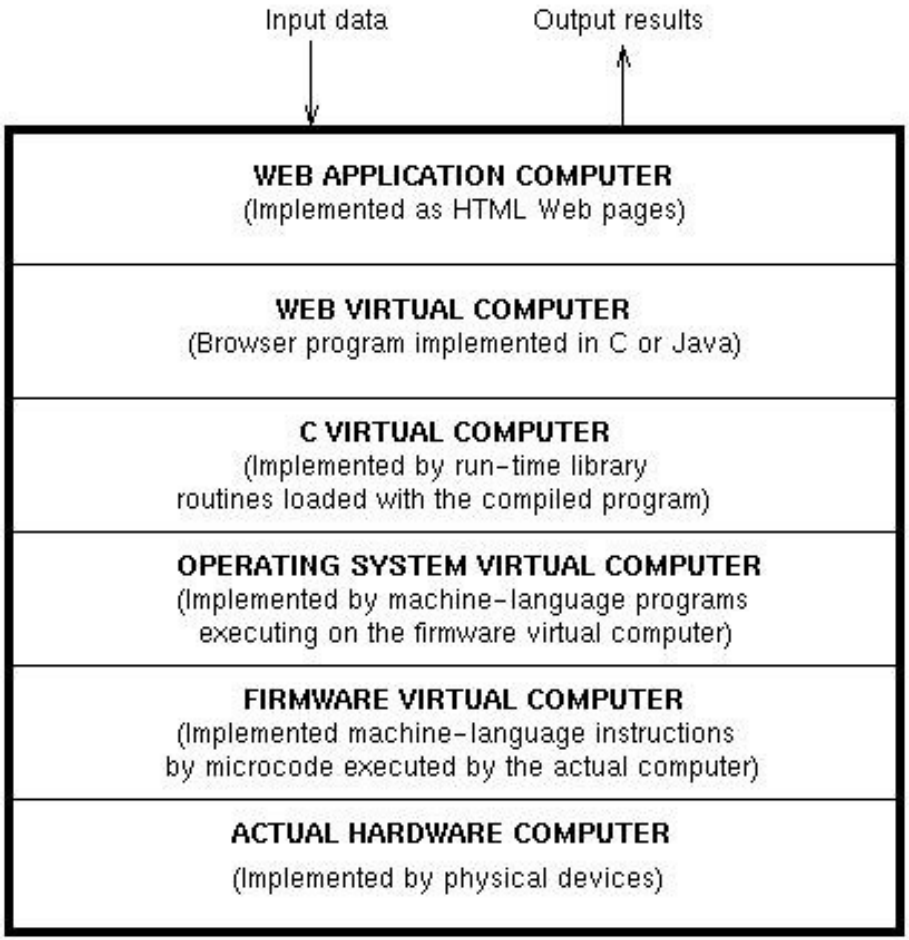


NOW...

- No longer write directly in machine language.
- Use of layers of softwares.
- **Concept of Virtual Machines.** Each layer is a machine that provides functions for the next layer

Chapter 2b: Language Design Issues

Layers of Virtual Machines



Chapter 2b: Language Design Issues

Virtual Computers and Language Implementation

To implement a Programming Language:

- The implementor determines the underlying virtual computer.
- Implementor must also determine precisely what is to be done during translation of a program and what during execution.

Chapter 2b: Language Design Issues

Factors Affecting Language Implementation

- Differences in each implementor's conception of the virtual machine.
- Differences in the facilities provided by the host computer.
- Differences in the choices made by each implementor as to how to simulate the virtual computer elements and how to construct the translation.

Reminders

Again, Reminders

....