CS316 9-03-09

1. Modern Programming Language Paradigms
    - i. Procedural
    - ii. Object-Oriented
    - iii. Functional
    - iv. Logic
    - (b) Assembly → Procedural → Object-Oriented → Functional Logic
      Lower Level --> Higher-Level (efficient execution becomes more difficult)
    - (c) Hybrid Languages combine two or more of the above paradigms
        - i. C++ = C part (procedural) & Object-Oriented part
        - ii. Lisp = Procedural (assignment/loops) & Functional
        - iii. Programmers have choices of paradigms in a single language
    - (d) Script Languages
        - i. Used for specialized purposes where execution efficiency is not the most significant factor
        - ii. Web applications
        - iii. GUI/Interactive applications
        - iv. Small to medium business applications
        - v. Interpreter execution, no compilation
        - vi. JavaScript, Python, HTML PHP
2. Overview of Programming Language Implementations
    - (a) Key Concepts:
        - i. Hardware machine (hardware processor)
        - ii. Virtual machine (software simulator, software interpreter)
        - iii. High-level languages → (compilation) → intermediate languages → machine language
          (C++, Java, lisp, prolog) JVM(Java byte code) (Intel CPU, AMD CPU, etc)
            - A. High level languages are executed by
                - virtual machines ("VH")(pure interpreter, implementation, BASIC) and
                - hardware machines ("HH")
            - B. Intermediate languages are executed by
                - virtual machines ("VI") and
                - hardware machines ("HI")
            - C. Machine languages are executed by
                - virtual machines ("VM")and
                - hardware machines ("HM")
        - iv. How does the execution compare with the above?
            - A. Slowest of 6: VH
            - B. HH is much faster than VH
            - C. HI is much faster than VI
            - D. HM is much faster than VM
            - E. VI is much faster than VH
            - F. VM is faster or equal to VI
            - G. HI is much faster than HH
            - H. HM is faster or equal to HI
            - I. HM is the fastest of all 6
            - J. What about HH to VI and HI to VM?

        - They are incomparable

(b) Intermediate Language as a bridge between multiple high level languages and multiple machine languages

   i. m high-level languages
      - A. High-level language 1 ….. High-level language m
      - B. High level languages 1 through m compile into a Common Intermediate Language
      - C. Common Intermediate Language compiles into Machine Languages 1 through n
      - D. High Level Languages 1 through m are the front ends to Machine Languages 1 through n
      - E. Machine Languages 1 through n are the back ends to High Level Languages 1 through m
      - F. Common Intermediate Language can be executed by a Virtual Machine (“portability” = hardware/OS are platform independent)

   ii. Only need to construct m front ends + n back ends.
      - A. Without the use of a Common Intermediate Languages, we would need to construct $m \times n$ (front + back) ends.
        - Example: m=10, n=10, 10 fronts + 10 backs
          ١٠×١٠=١٠٠ front + back
      - B. Other Benefits of Intermediate Languages
        - Better optimization (removal of redundant instructions)
          - Thanks to cleaner structure, the compiler can recognize redundant instructions more effectively
        - Provides 2 execution options:
          - by a virtual machine
          - by further compilation to specific hardware machine code