COMS 1001 Spring 2006

Introduction to Computers

Languages, Levels, and Machines
What's Ahead

- Levels and Languages
- COBVM, sh.c
- Software Lifecycle
- Hangman Hint
Software Life Cycle

- Problem Statement
- Requirements Specification
- Design Specification
- Construction or Implementation
- Testing
- Documentation
- Maintenance
Design Strategies

- Top-down (outline)
- Bottom-up (essential details first, building blocks)
- Object Oriented (OOP)
- eXtreme Programming (feedback from customer)
- Many others, which are both design strategies and programming styles or disciplines...
Taking stock: Tradeoffs

- CS has a number of “vs” relationships:
  - Big vs. Little endian
  - RISC vs. CISC
  - Interrupts vs. Polling
  - Translation vs. Interpretation
  - Open vs. Closed source
  - Combinational vs. Sequential circuits
  - Space vs. Time
Translation and Interpretation

- Source Code
  - Compiles to:

- Object Code (executed by hardware)
  - Or:

- Interpreted (executed by some other software)
  - Javascript is interpreted
  - Bash is interpreted
What Are Some Programming Languages?

● High-level vs. low-level languages

● Languages are defined by precise grammars: the rules of what alphabet symbols can appear where in a source code listing

● A language is a model of syntax and semantics

● Compiled vs. interpreted
Machine Instruction Set

• The “native language” of the processor
  – Set of directly executable commands
  – Binary string
  – Opcode + data/address

• programmers use an assembly language to represent the opcodes

• Operands
Low Level Languages

• Assembly Language
  – Mneumonics for machine instruction set
  – Opcode + address/operands
  – Function( x, y )
  – Add a, b
  – And a, b
  – Mul a, b

• Execution begins at top and usually falls straight through
Assembly Language

• Data Transfer
  – Mov, store, load, push, pop
• Control Flow
  – Jmp, jnz, cmp, nop
• Logical
  – And, or, xor, shift, not
• Math
  – Add, mul, sub, inc, div
High-Level Languages

- Resemble everyday speech, but more structured and precise
- Variables
- Types
- Structured Blocks
- Higher-level control flow constructs (loops)