Basic Terms & Concepts

- hardware
- software
- program
- run, execute, invoke, launch
Basic Terms & Concepts

- hardware  physical components of a computer
- software
- program
- run, execute, invoke, launch
Basic Terms & Concepts

- **hardware**: physical components of a computer
- **software**: programs
- **program**: specified pattern of instructions for a computer to perform
- **run, execute, invoke, launch**
Basic Terms & Concepts

- **hardware**: physical components of a computer
- **software**: programs
- **program**: specified pattern of instructions for a computer to perform
- **run, execute, invoke, launch**: make the program go (start performing the instructions)
Hardware Terms (1)

- CPU
- ISA
Hardware Terms (1)

• CPU
  – Central Processing Unit
  – Part of microprocessor (e.g., G4, Pentium, etc.)
  – the "brain" of the computer

• ISA
**Hardware Terms (1)**

- **CPU**
  - Central Processing Unit
  - Part of microprocessor (e.g., G4, Pentium, etc.)
  - the "brain" of the computer

- **ISA**
  - Instruction Set Architecture
  - what specific instructions the CPU can perform
Hardware Terms (2)

• Primary (Main) Memory

• Secondary Memory
• Primary (Main) Memory
  – RAM = Random Access Memory
  – “fast” storage, “close” to CPU
  – stores both an executing program and the data it works on
  – e.g., Microsoft Word + a current English paper

• Secondary Memory
  – slower storage, “outside” main unit
  – hard disks
  – “Tertiary”: floppies, tapes etc.
Flow of Computation (abstract)

Data

Program

(computer)

Results
A Simplistic View of a Computer – Flow of Computation

Storage

Input

Output

CPU

Input

Output

Storage
How do we tell a computer what to do?

- Design an **Algorithm**
  - An algorithm is a sequence of instructions used to accomplish a goal

- Write a **Program**
  - A program is a translatable implementation of an Algorithm
How do we tell a computer what to do?

- Is 2008 a leap year?
How do we tell a computer what to do?

- Is 2008 a leap year?
- Is 2007 a leap year?
How do we tell a computer what to do?

• Is 2008 a leap year?

• Is 2007 a leap year?

• Is 2000 a leap year?
How do we tell a computer what to do?

• Is 2008 a leap year?
• Is 2007 a leap year?
• Is 2000 a leap year?
• Is 1900 a leap year?
How do we tell a computer what to do?

• Is 2008 a leap year?

• Is 2007 a leap year?

• Is 2000 a leap year?

• Is 1900 a leap year?

• Write an algorithm to compute is a given year is a leap year.
How do we tell a computer what to do?

- Program: Specified pattern of instructions for a computer to perform

- Need a language to describe (specify) this instruction pattern
  - An intermediary between humans and the computer

- Language should be:
  - Convenient for expressing programs
  - Convenient for computers to “understand”

- Why not “Natural Languages”?
  - (like English)
Programming Languages

• Levels of Computer Languages
  – Machine language
  – Assembly language
  – High-level language
Machine Language

0001110010000110 1101110110000010 0101001100110111

- Sequence of instructions in binary code
- Code corresponds directly to instruction set of computer
- Direct control of machine
- Difficult to program, read, debug
- Not portable to other platforms
Assembly Language

LDA 5
ADC 6
STA sum

- These instructions typically correspond 1-1 with machine language instructions
- Mnemonic codes
- Symbolic names (like sum)
- Direct control
- Still hard to use and not portable
High-Level Languages

sum = 5 + 6;

- Examples: C++, Fortran, Lisp, Java, …
- Closer to natural language
- Easier to read, write, debug & modify
- Employ powerful data and control primitives, which allow relatively concise specification of complex computation patterns
Why C++?

- Popularity
- Portability
- Speed
- Functionality
- Features
Questions?
Questions?

- 1) What is the “official” name of the “brain” of the computer?
Questions?

• 1) What is the “official” name of the “brain” of the computer?
• 2) list 2 input devices
• 3) list 2 output devices
Questions?

• 1) What is the “official” name of the “brain” of the computer?
• 2) list 2 input devices
• 3) list 2 output devices
• 4) the following code is an example of
  – assembly language
  – machine code
  – high level language
    ```
    int x = 2, y = 5;
    int sum;
    sum = x + y;
    cout << sum;
    ```