Ch 1: Intro to Computers and Programming

CS 1428
Fall 2011

Jill Seaman

Lecture 1

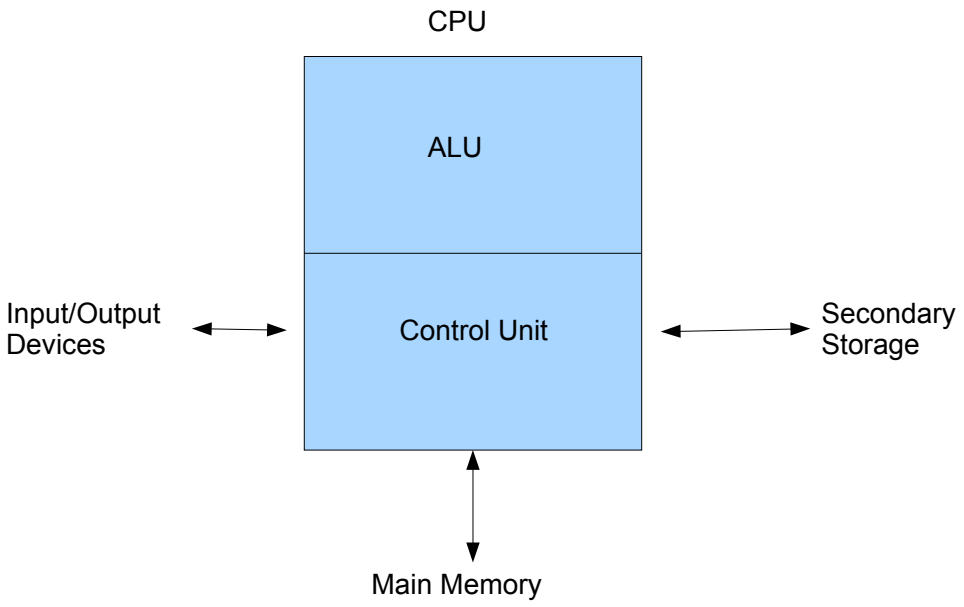
Computer Systems

- Hardware Devices
- Software Components

Hardware

- **Central Processing Unit** (CPU)
  - **Arithmetic Logic Unit** (math, comparisons, etc)
  - **Control Unit** (processes instructions)
- **Main Memory** (RAM): Fast, expensive, volatile
- **Secondary Storage**: Slow, cheap, long-lasting
- **Input Devices**: keyboard, mouse, camera
- **Output Devices**: screen, printer, speakers

Organization of Hardware



Software

- Programs that run on the hardware
- Operating Systems:
  - Let user operate hardware and run apps, manage environment
  - Unix, MS-DOS, Linux
  - Windows, Mac OS X
- Application Programs (Apps):
  - Solve specific problems for user
  - Word, Excel, iTunes, Firefox, Angry Birds, Outlook

What is a Program?

- Set of instructions to perform a specific task (an **Algorithm**)
- Runs on a computer

Example (algorithm)

1. Display on screen: “how many hours did you work?”
2. Wait for user to enter number, store in memory
3. Display on screen: “what is your pay rate (per hour)?”
4. Wait for user to enter rate, store in memory
5. Multiply hours by rate, store in memory
6. Display on screen: “you have earned $xx.xx” where xx.xx is result of previous step

Note: Computer does not speak English

Programming Languages

- Machine Language:
  - Instructions are Sequence of 1's and 0's
  - Machine specific
- Low Level Languages: Assembly Language
  - Letters and digits
  - Direct correspondence to Machine Language
- High Level Languages:
  - Words, symbols, numbers
  - Easier for humans to read and use
  - Must be translated to Machine Code

Translation Process

Source Code File → [Preprocessor] →

Modified Source Code → [Compiler] →

Object Code → [Linker] →

Executable Code File

Usually don't see intermediate files

Using an “Integrated Development Environment” (like Eclipse) you may only see the source, and result of running the executable file.

```
#include <iostream>
using namespace std;

int main()
{
   double hours, rate, pay;

   // Get the number of hours worked
   cout << “How many hours did you work? ” ;
   cin >> hours;

   // Get the hourly pay rate
   cout << “How much do you get paid per hour? ”;
   cin >> rate;

   // Calculate the pay
   pay = hours * rate;

   // Display the pay
   cout << “You have earned $” << pay << endl;

   return 0;
}
```

Language Elements
- Key Words
- Programmer Defined Identifiers
- Operators
- Punctuation
- Statement
- Variables
- Variable Definition

Language Elements

- Key Words: have special meaning (lowercase)
- Programmer Defined Identifiers: names made by programmer
- Operators: instruction to manipulate data
- Punctuation: special meaning to compiler
- Statement: complete instruction to computer
- Variables: named storage location
- Variable Definition:
  - instruction to set up variable
  - requires data type information (number, character)

Categories of Instructions

- Input
  - cin >> hours
  - gathers info from “outside world”
- Processing
  - pay = hours * rate;
  - computation
- Output
  - cout << “How many hours did you work? “;
  - sends info to “outside world”

Programming Process

1. Clearly define the problem
2. Visualize output of program
3. Make a model of the program
    - hierarchy chart
    - flowcharts
    - pseudocode
4. Translate to C++ code (type it into a file)
5. Compile, fix syntax errors, repeat
6. Test the program (execute it with data)
7. Correct errors, go to step 5. If no errors, quit.

What is Software Engineering?

Entire process of developing and maintaining computer software

- Designing
- Writing Code
- Testing
- Debugging
- Documenting
- Modifying (updating)
- Maintaining (fixing bugs reported by users)