

# Week 0: Intro to Computers and Programming

Gaddis: Sections 1.1-3 and 2.1

CS 1428  
Fall 2015

Jill Seaman

1

## 1.1 Why Program?

Computer – programmable machine designed to follow instructions

Program – instructions in computer memory to make the computer do something

Programmer – person who writes instructions (programs) to make computer perform a task

SO, without programmers, no programs; without programs, a computer cannot do anything

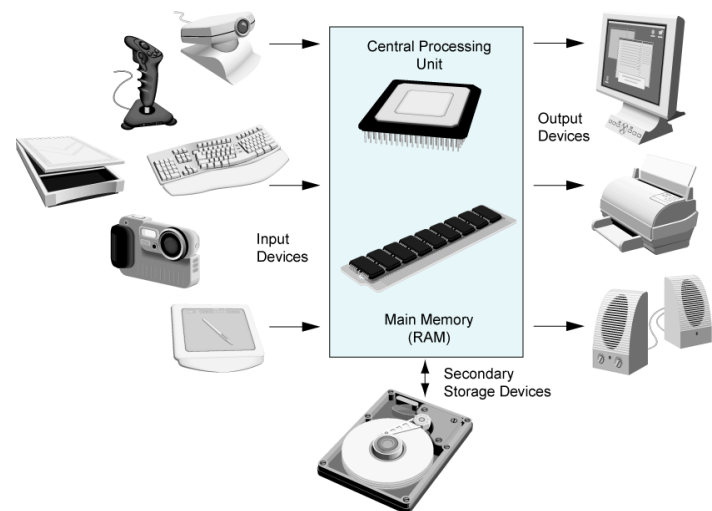
2

## 1.2 Computer Systems: Hardware and Software

- Hardware: the physical components that a computer is made of.
- Software: the programs that run on a computer

3

## Hardware Components Illustrated



4

## Hardware Components

- **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

5

## Software

- Programs that run on the hardware
- **Operating Systems (System software):**
  - programs that manage the computer hardware and the programs that run on them.
  - Unix, MS-DOS, Linux, Windows, Mac OS X
  - Time machine, printer drivers, compilers
- **Application Programs (Apps):**
  - Solve specific problems and provide services to the user
  - Word, Excel, iTunes, Firefox, Angry Birds, Photoshop

6

## 1.3 Programs and Programming Languages

- A program is a set of instructions that the computer follows to perform a task
- An algorithm:
  - ▶ A set of well-defined steps for performing a task or solving a problem.
  - ▶ A step by step ordered procedure that solves a problem in a finite number of precise steps.
- An algorithm can be in any language (English, C++, machine code, etc).

7

## Example (algorithm)

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

**Note:** Computer does not speak English, it only understands its own "machine language"

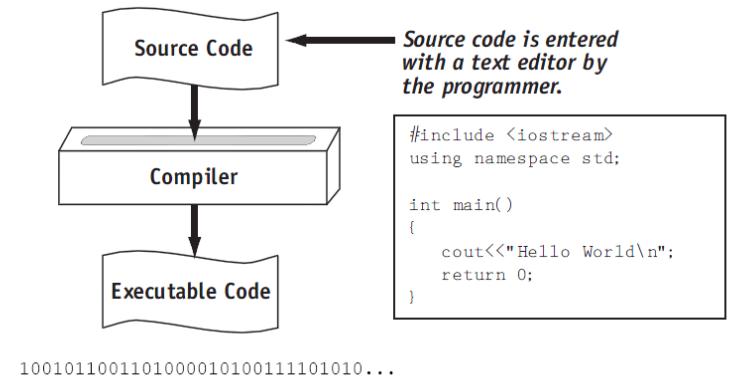
8

# Programming Languages

- Machine Language:
  - ▶ Instructions are encoded as a sequence of 1's and 0's
  - ▶ Machine specific
- Low Level Languages: Assembly Language
  - ▶ Letters and digits (codes)
  - ▶ Direct correspondence to Machine Language
- High Level Languages (like C++):
  - ▶ Words, symbols, numbers
  - ▶ Easier for humans to read and use
  - ▶ Must be translated to Machine Code

9

# Translation Process



Tony Gaddis, Starting out with C++: From Control Structures Through Objects 7th ed.

10

## 2.1 The Parts of a C++ Program

```
// sample C++ program
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello, world!";
    return 0;
}
```

11

## Parts of a C++ Program

- Comment: `//...`
  - ▶ ignored by compiler
  - ▶ notes to human reader
- Preprocessor Directive: `#include <iostream>`
  - ▶ compiler inserts contents of file `iostream` here
  - ▶ required because `cout` is defined in `iostream`
- `using namespace std;`
  - ▶ allows us to write `cout` instead of `std::cout`

12

# Parts of a C++ Program

- `int main ( )`
  - start of function (group of statements) named `main`
  - the starting point of the program
- `{ }`
  - contains the body of the function
- `cout << "Hello, world!";`
  - statement to display message on screen
- `return 0;`
  - quit and send value 0 to OS (means success!)