180 Introduction

Class 1

Administrative

- class web site
- syllabus
- grades
- textbook Gaddis
- classroom electronics policy

Academic Honesty

- you may
  - discuss the mechanics of editing, compiling, and running a program
  - discuss the mechanics of file names, uploading
  - discuss the mechanics of using the shell, printing
  - discuss the *general* strategy for completing an assignment
  - use code from me or Gaddis, original or modified, without attribution
- you may not
  - look at any portion of another student's code or writeup
  - show any portion of your code or writeup to another person
  - discuss the details of any assignment, in person or electronically
  - copy code from any source other than me and Gaddis
- you must
  - cite any source of ideas other than me or Gaddis
  - this means if you looked something up on the web, you have to tell me the exact URL

Software

- C++11
- clang and llvm version 4.0 or later
- Linux server *ice*
- Code::Blocks IDE

Keys to Success

- computer science is so much more than programming
- but you cannot be a computer scientist without programming
- this class is largely about developing programming skills
- program every day
  - type in, run, and experiment with my code examples
  - implement the programs in the text
- this class moves fast and is cumulative
- if you fall behind, you're toast
  - time management!
  - get help early!

Tools

- humans build and use tools
- a computer is a tool
- most tools are designed to do a task

The Computer Tool

- the computer is a tool
- uniquely, it is designed not to do a task
- it is designed to be told how to do a task
- a computer can be used to do a task that didn't exist when the computer was built

- this is accomplished through software
- software development, or software engineering, is the process of designing, building, testing, and using software
- software engineering is a major area of computer science
- a main aspect of software development, and thus of computer science, is programming

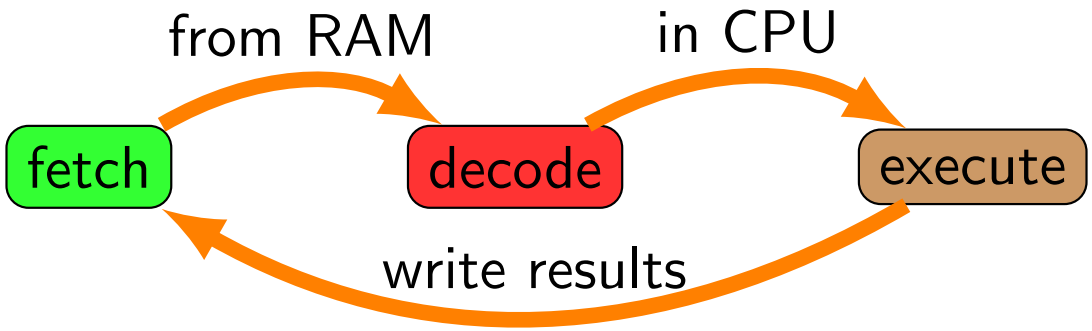
Computer Systems

computer systems are made of

- hardware
  - input devices
  - output devices
  - primary and secondary storage
  - CPU
- software
  - system software
  - application software (apps)

this course is about writing apps

The CPU Cycle



Memory aka RAM

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16 **149**	17	18	19
20	21	22	23 **72**	24	25	26	27	28	29

- the most important concept to understand for programming
- volatile: contents vanish when program terminates or computer is turned off
- composed of bits, with values 0 (off, false) and 1 (on, true)
- 8 consecutive bits is a byte
- bytes and larger chunks have addresses
- above, the value 149 is stored in the location with the address 16, and the value 72 is stored at address 23

Algorithm

- Muhammad ibn Musa al-Khwarizmi was a Persian scholar
- lived in Baghdad (now Iraq) in the 800's
- around 820 he wrote a "book" giving precise, unambiguous, mechanical, efficient, and correct instructions for adding and multiplying numbers, and for calculating square roots, using decimal numbers
- the title of his book included the word *al-jabr*, from which our word algebra comes
- the word algorithm is named for al-Khwarizmi

Algorithm

A set of precise, unambiguous, mechanical, efficient, and correct instructions for accomplishing a task.

An Algorithm

an algorithm for calculating gross pay

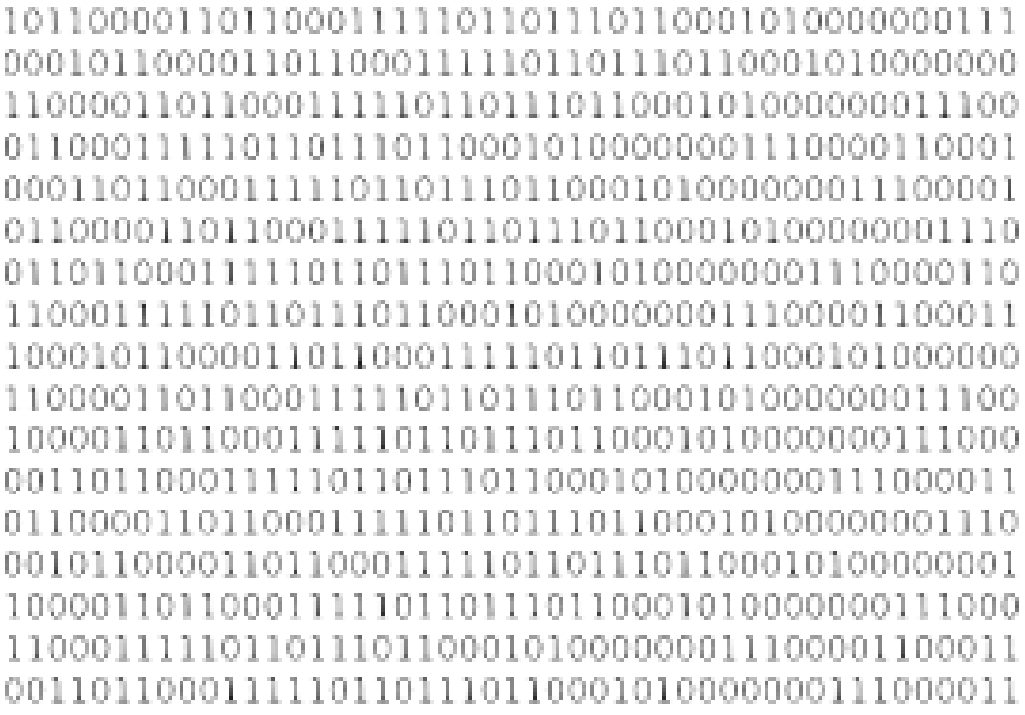
1. Display the message, "How many hours did you work?"
2. Wait for the user to enter a value
3. Store the entered value in a memory location
4. Display the message, "How much do you get paid per hour?"
5. Wait for the user to enter a value
6. Store the entered value in a memory location
7. Retrieve the number of hours and the amount paid per hour from their memory locations, multiply them together, and store the product in a memory location
8. Retrieve the product from its memory location and display it along with an explanatory message

Understanding

- if you don't understand the process, you can't write an algorithm
- if you can't write the algorithm, you can't write a program

Machine Language

- an algorithm is written in English
- computers can't understand English
- a computer can only execute machine language instructions
- machine language instructions are coded numbers

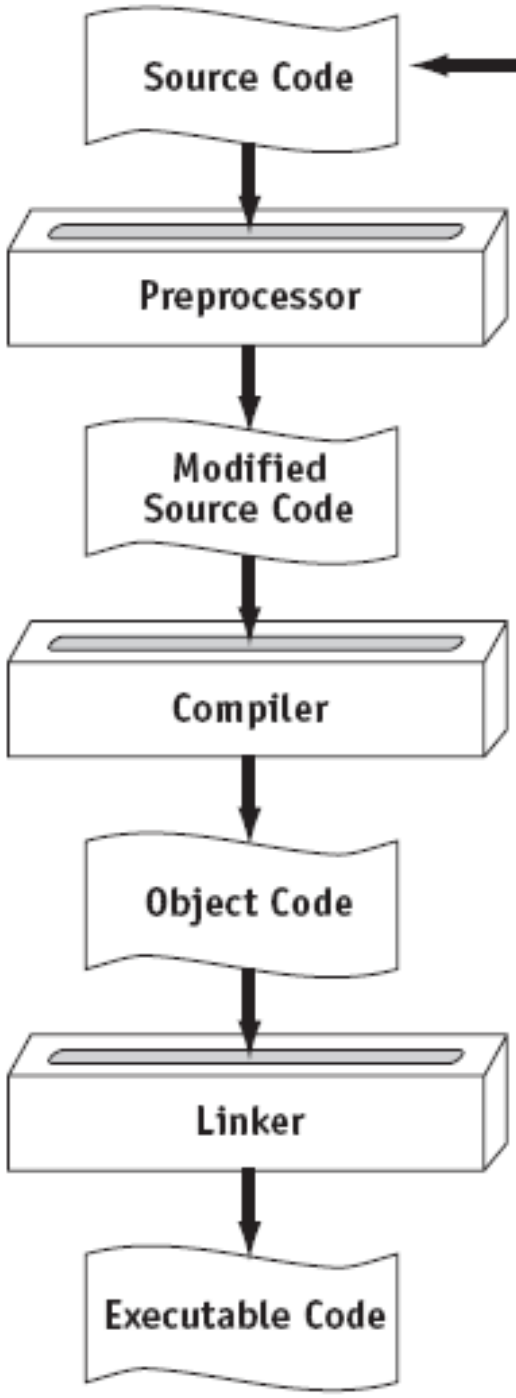


Some Programming Languages

C++ to Executable

1. create C++ source code file using a text editor
2. preprocessor: convert source file directives to source code program statements
3. compiler: convert source code statements into machine instructions
4. linker: connect hardware-specific code to machine instructions, producing a file of machine language statements

- steps 2–4 are often performed by a single command or button click
- an error at any step will prevent the following steps



Source code is entered with a text editor by the programmer.

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

int main()
{
   cout<<"Hello World\n";
   return 0;
}
```

Compilers

- for C++, these three are most used:
  - Microsoft Visual C++ (VC++) only works for Windows
  - the gnu C++ compiler (gcc) the classic free software, for all platforms
  - clang + llvm now the default for Macintosh, rapidly taking over from gcc, also for all platforms

Integrated Development Environment

- you could use separate editor, compiler, linker, debugger
- in fact you will do this in CS250

- but an IDE combines all the tools needed to write, compile, and debug a program in one package and is easier to learn
- examples: Microsoft Visual C++, Turbo C++ Explorer, CodeWarrior
- we will use Code::Blocks with llvm and clang

Tomorrow

- tomorrow we will be in the lab
- go straight to VH1232
- please do not bring your own laptop to lab this week
- we will all use ice
- you can use your laptop later if you wish