Instructor: Keith Hellman `khellman@mines.edu`

http://inside.mines.edu/~khellman/teaching/csci261

Your TODO list:

1. Make sure you are able to login to `ADIT`.
2. Double-check your enrollment for this course on **Blackboard**
   http://blackboard.mines.edu
3. Make sure your **Official Correspondence** Email address is setup correctly in Trailhead.
   http://inside.mines.edu/~khellman/official_corres.html
4. Read and understand the **syllabus** and **collaboration policy**:
   https://csci261.mines.edu/csci261/

Introduction to Computing

November 15, 2010

Why Learn Programming?

- As an engineer, you will have conversations with programmers; *and you will need to communicate effectively with them.*
- Top-quality software used by professionals can often be scripted to increase the productivity of the user. *Knowing a programming language makes scripting easier to learn.*
- Programming skills are a “plus” with many employers.
- If you want to solve new problems, you might have to write new programs.
- It can be a lot of ***FUN***.

The Book Definitions

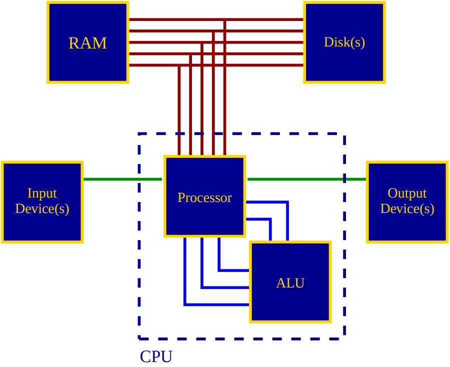
Computer A machine designed to perform operations or tasks through a sequence of instructions (the **program**).

Hardware Refers to the computer's components, much of it with embedded software.

Software The programs executed by the CPU. Resides on disk, loaded to RAM, and executed by the CPU.

The Nuts and Bolts

Processor	Controlling Circuitry of Computer
ALU	Arithmetic and Logic Unit.
RAM	Random Access Memory, volatile but fast storage.
Disk	Non-volatile but slow storage (IDE, SATA, Flash, ...)



CPU = Processor + ALU

Software Languages

Machine Language Binary Instructions

`010101011...`

Assembly Language Specific to a CPU, manipulates the internal components of a CPU.

`PUSH/POP stack, CMP registers, LOAD registers, ...` ⟹

High Level Languages C, C++, Ada, Fortran, Python, Java, ...

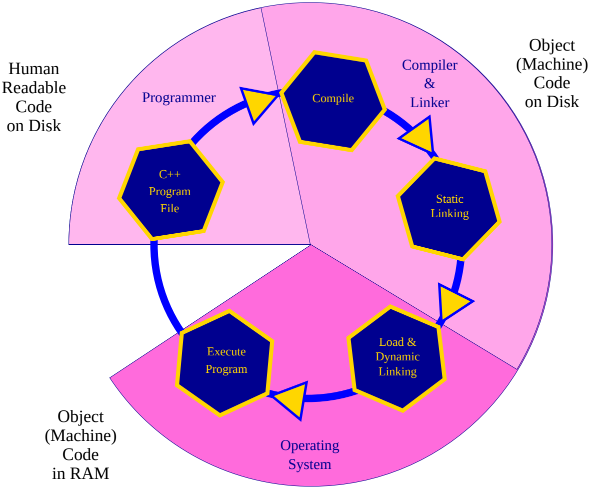
`print "Hello World"`

x86 Assembler

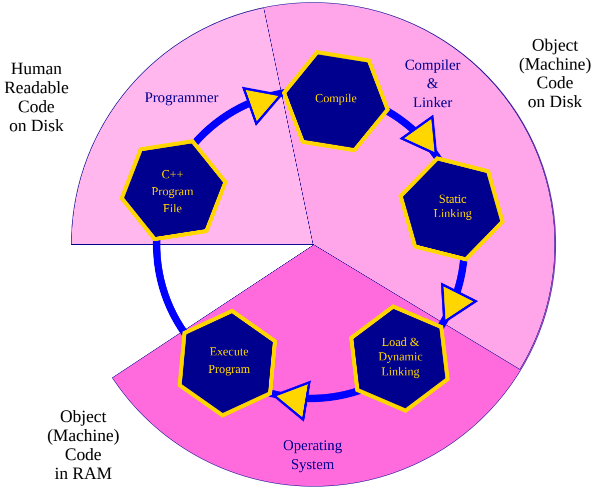
```
:
:
fldl    -0x28(%ebp)
fmull   -0x20(%ebp)
fmull   -0x18(%ebp)
fstpl   -0x10(%ebp)
movl    $0x0,0x4(%esp)
movl    $0x0,(%esp)
call    ba <main+0x48>
fldl    -0x10(%ebp)
fstpl   0x4(%esp)
mov     %eax,(%esp)
call    c9 <main+0x57>
movl    $0x10,0x4(%esp)
movl    $0x0,(%esp)
call    dd <main+0x6b>
mov     $0x0,%eax
add     $0x34,%esp
pop     %ecx
pop     %ebp
lea     -0x4(%ecx),%esp
ret
```

The “Toolchain” Software States

How does software go from human readable form to a running program?

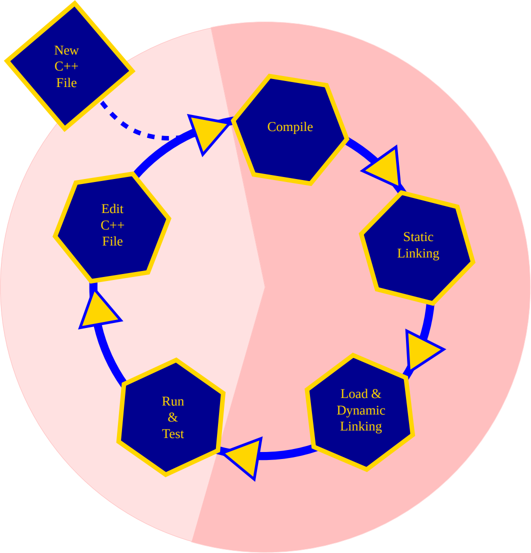


“Toolchain” Demonstrations



Ice→Water→Steam Demo

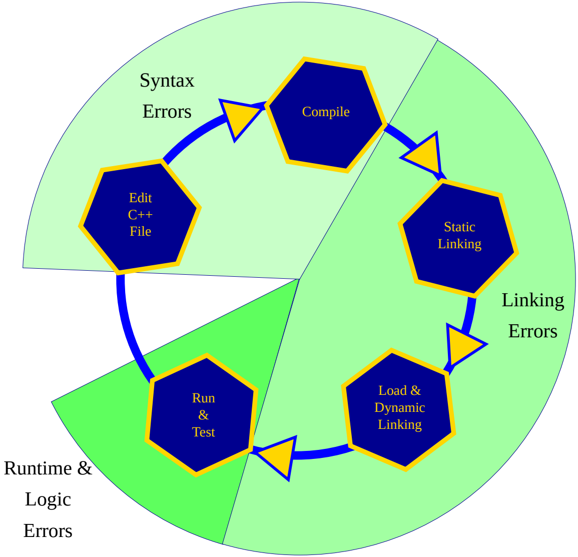
C++ Development Cycle



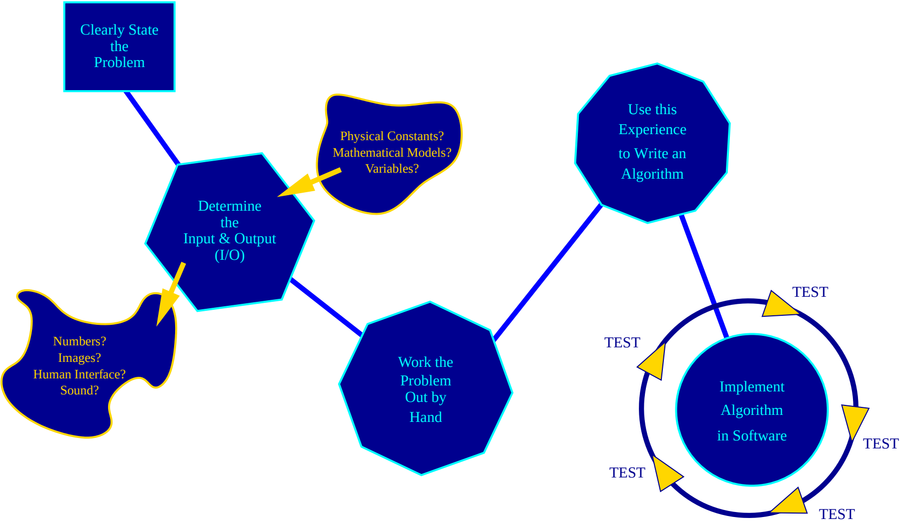
The “Art
of
Programming”

The
Programmer’s
“Toolchain”

Types of Errors



Designing a Computer Program



Clearly State the Problem

Determine the Input & Output (I/O)

Physical Constants? Mathematical Models? Variables?

Numbers? Images? Human Interface? Sound?

Work the Problem Out by Hand

Use this Experience to Write an Algorithm

Implement Algorithm in Software

TEST

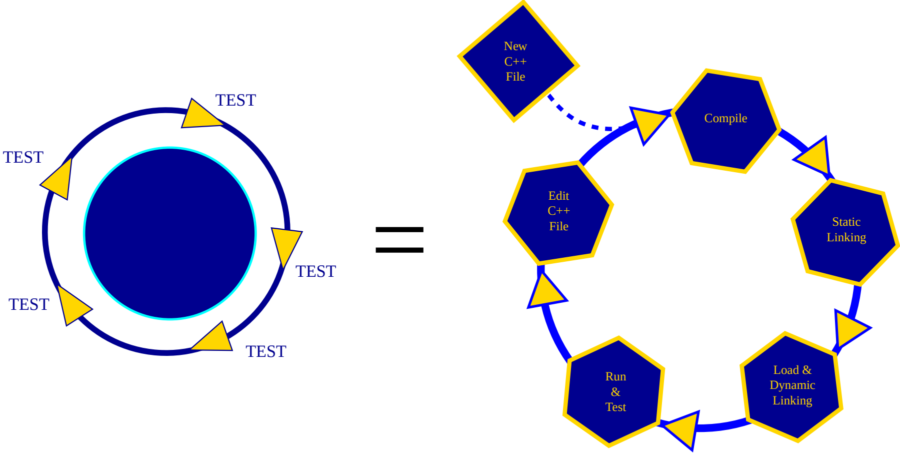
TEST

TEST

TEST

TEST

Designing a Computer Program



finis

The Volume of a Box

A. Write a program to compute the volume of box.

B. All inputs and outputs are real numbers ($\Re$)
Input: length, width, height
Output: volume

C. If the dimensions of the box are $20.75 \times 11.5 \times 9.5$, then

$$Volume = 20.75 \cdot 11.5 \cdot 9.5 = 2266.9375\text{un}^3$$

D. Algorithm:
1. initialize length, width and height
2. compute volume
3. output volume

E. Implement algorithm in C++ and test.

Implement Algorithm in C++

```
//
//  This program computes the volume of a box
//
#include <cstdlib>
#include <iostream>
using namespace std;

int main()
{
    /*  Declare and initialize objects */
    double length(20.75), width(11.5);
    double height = 9.5;
    double volume;

    /*  Calculate volume. */
    volume = length * width * height;
    /*  Print the volume. */
    cout << "The␣volume␣is␣␣" << volume ;
    cout << "␣units␣cubed." << endl;

    system("PAUSE");
    //  Exit program.
    return 0;
}
```