



CS 161

Intro to CS I

What is CS all about?



Odds and Ends

- Go to Lab this week (laptop required)
- Assignment 1 posted and can upload to Peerceptiv
- Math study: Elise.Lockwood@oregonstate.edu
- Questions?



How to Be Successful

- Read and listen carefully
- Start assignments early
- Be proactive with absences and issues that arise in the term
- Get help when you need it



Help Hierarchy

- Reread assignment, lecture slides, labs, syllabus
- Google/Bing/Open a textbook
- Ask a friend
- Ask a TA
- Ask Jennifer

- All Emails Should Include:
 - What your problem is
 - What you have tried
 - What would help you most
 - Section number (if relating to a grade issue)



Computers Are Everywhere

- Examples:
 - homes, offices, rooms/servers, phones, pacemakers, cars, etc.
- What is the difference b/w these?
 - Complexity
 - Size



What is a computer?

- A Computational Device
 - It computes (input-> processing -> output)
 - Modern: device that can be programmed to carry out an algorithm.
- What is Computer Science?



What is an algorithm?

- Step-by-step description of how to accomplish a task, i.e. recipe
- Algorithmic thinking
- Expressed in any language
 - Natural
 - Programming



What is programming?

- Problem Statement
- Solve the Problem
- Specify Algorithm
- Algorithm -> Computer Language

- Why do we teach programming 1st?



Hardware vs. Software

- Computer: **machine** that manipulates data and carries out **set of instructions**
- Hardware
 - CPU
 - RAM
 - Hard Disk
- Software
 - Programs



Software/Programs

- Primary piece of software on computer?
- What is its purpose?
- What are applications?

Digital Realm



- Based on discrete #s
 - Specifically: Circuits
- Binary, i.e. base 2
 - 0 or 1



More Binary

- What is each digit called?
- What is a Byte?
- How many numbers can be expressed in a Byte?



What does this mean for us?

- Unsigned
 - What is the smallest number?
 - What is the largest number?

- Signed
 - What is the largest number?

 - What is the smallest number?



Programming

- Writing **code** that a computer can **execute**
 - Does that mean we have to write in binary?
- High-level language
 - Translated Continuously during runtime
 - Interpreted
 - Just in time compilation/caching
 - **Translated Prior/Ahead of time to runtime**
 - High-level -> machine language
 - High-level -> intermediate language

C++ Programming Environment



- Create a program: **vim hello.cpp**
- Compile program: **g++ hello.cpp -o hello**
- Run program: **./hello**
- Example: **hello.cpp**

```
#include <iostream>
int main() {
    std::cout << "Hello CS 161 Class!!!";
    return 0;
}
```

Our first C++ program!



```
#include <iostream>
int main() {
    std::cout << "Hello CS 161 Class!!!";
    return 0;
}
```

- Libraries
 - Example: `#include <iostream>`
- Functions
 - Perform particular action/computation
 - Requires special function: **main**
 - `int main() {....}`
- Statements
 - Ended by semicolon



Comments

- Ignored by compiler
- Comment a block of code: `/*.....*/`
- Comment one line of code: `//`
- Why use these?

What are you required to have now?



- Header at beginning of program and other appropriate comments

```
/******  
** Program: hello.cpp  
** Author: Jennifer Parham-Mocello  
** Description: This program prints hello world to the console  
** Input: none  
** Output: hello world text  
*****/
```

- Appropriate use of line comments



More C++

- Programming Style: please read your class style guide
 - Program Header/Description
 - Placement of {}
 - Indentation: spaces vs. tabs
- String Literal in quotations, ""
 - Not single quotes!
 - INCORRECT: `std::cout << 'Hello World';`
 - Do not span more than one line!
 - INCORRECT: `std::cout << "Hello
World";`



More C++

- Escape Sequences
 - Display special characters
 - Use backslash, \, before special character to print
- Examples:

```
std::cout << "\"Hello World\""\n";
```
- Refer online for common escape sequences:
<http://en.cppreference.com/w/cpp/language/escape>