CS 180: Intro to C++

Announcements

- Syllabus
- Lab on Friday this week
- Code: 80386
Resources for this class

- Text book
- Transition guide (look for pdf on webpage)
- cplusplus.com
- Tutoring & office hours
This course: data structures in C++

First, C++. (More on that next.)

But — what is a data structure?

store data : container

Ex: - tree
   - list
   - stack

\[\text{Diagram of a tree}\]
C++ versus Python

High level versus low level.

Interpreter versus compiler.
On the fly
2 steps: compile, run

Dynamic versus static typing

\[
\begin{align*}
\text{C++} & : \\
\text{Python} & :
\end{align*}
\]

\[
\begin{align*}
\text{C++} & : \\
\text{Python} & : \\
\text{C++} & :\\
\text{Python} & :
\end{align*}
\]
Why learn C++?

- faster
- ubiquitous everywhere!
- understand low level details
- control
**Comparison**

**Python**

```python
def gcd(u, v):
    # we will use Euclid's algorithm
    # for computing the GCD
    while v != 0:
        r = u % v  # compute remainder
        u = v
        v = r
    return u

if __name__ == '__main__':
    a = int(input('First value: '))
    b = int(input('Second value: '))
    print('gcd:', gcd(a, b)
```

**C++**

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

int gcd(int u, int v) {
    /* We will use Euclid's algorithm
    for computing the GCD */
    int r;
    while (v != 0) {
        r = u % v;  // compute remainder
        u = v;
        v = r;
    }
    return u;
}

int main() {
    int a, b;
    cout << "First value: ";
    cin >> a;
    cout << "Second value: ";
    cin >> b;
    cout << "gcd: " << gcd(a, b) << endl;
    return 0;
}
```