Admin

- Section signups available on web, now until Sun 5pm
- CS and the Honor Code
- Alternate final exam
  - I relented. Will offer final Th Mar 20 12:15-3:15pm. Absolutely NO other alternates.
- Cafe hangout today after class in Terman — join us!
- Today's topics
  - C++ syntax and structure, procedural paradigm
  - User-defined types, parameter passing
- Reading
  - Handout 4, Reader Ch. 1, 2.1, 2.6 (today)
  - Ch. 3(next)

C++ vs Java: what's the same?

- General syntax
  - comment sequence
  - use of braces, parentheses, commas, semi-colons
  - variable/parameter declarations, function call
- Primitive variable types
  - char, int, double, but note Java boolean is C++ bool
- Operators
  - arithmetic, relational, logical
- Control structures
  - for, while, if/else, switch, return

Dissecting a C++ program

```
/*
 * average.cpp
 * -----------
 * This program adds scores and prints their average.
 */

#include "genlib.h"
#include "simpio.h"
#include <iostream>

const int NumScores = 4;

double GetScoresAndAverage(int numScores);

int main()
{
   cout << "This program averages " << NumScores << " scores." << endl;
   double average = GetScoresAndAverage(NumScores);
   cout << "The average is " << average << "." << endl;
   return 0;
}
```

average.cpp (cont'd)

```
/* Function: GetScoresAndAverage
 * Usage: avg = GetScoresAndAverage(10);
 * -------------------------------------
 * This function prompts the user for a set of values and returns
 * the average.
 */
double GetScoresAndAverage(int numScores)
{
    int sum = 0;
    for (int i = 0; i < numScores; i++) {
        cout << "Next score? ";
        int nextScore = GetInteger();
        sum += nextScore;
    }
    return double(sum)/numScores;
}
```

C++ user-defined types

- Enumerations
  - Define new type with set of constrained options

```
enum directionT {North, South, East, West};

directionT dir = East;
if (dir == West) ...
```

- Records
  - Define new type which aggregates a set of fields

```
struct pointT {
  double x;
  double y;
};

pointT p, q;
p.x = 0;
p = q;
```

C++ parameter passing

- Default is *pass-by-value*
  - Parameter copies value, changes affect local copy

```
void Binky(int x, int y)
{
  x *= 2;
  y = 0;
}
```

```
int main()
{
   int a = 4, b = 20;
   Binky(a, b);
   ...
```

- Add & to declaration for *pass-by-reference*
  - Parameter is now reference to original variable, which can change

```
void Binky(int &x, int y)
{
  x *= 2;
  y = 0;
}
```

```
int main()
{
   int a = 4, b = 20;
   Binky(a, b);
   ...
```

  - Ref param also used for efficiency to avoid copying large data

C++ libraries

- Groups related operations
  - Header file provides function prototypes and usage comments
  - Compiled library contains implementation
- C++ standard libraries
  - e.g. string, iostream, fstream
  - `#include <iostream>`
  - Terse, lowercase names: `cout getline substr`
- CS106 libraries
  - e.g. simpio, random, graphics
  - `#include "random.h"`
  - Capitalized verbose names: `GetInteger RandomChange DrawLine`