CS 149D, slide set 4a

M. Overstreet
Old Dominion University
Spring 2006

From C++ text

- Read chapter 1

Grand Problems in Computing

- Set of problems
  - That seem like they could be solved (at least helped) by computers
  - People have worked on them for a long time without success
- Their solution would be very helpful

In biology: lots of challenges

- If we have supercomputers, what problems in biology could we solve that were impossible before?
- Biogeochemistry & membrane biology
- Bioinformatics and computational biology
- Computational cell biology

Programming Languages

- Lots of them
  - Different languages for different tasks
- Some are visual, some are textual
  - Many make heave use of algebraic expressions
  - Since lots of the early uses were mathematical
  - And the early users were comfortable with algebra
- They're often hard to use
  - For most people
  - Particularly at the beginning
  - Require too many details in an artificial language
- They're designed to be easy to use
  - But it's hard to give detailed instructions that cover all of the things that could happen

Focus on C++

- Widely used in the sciences
- Many of the computing challenges in the sciences are computationally intense
  - Need a language that can generate very time-efficient code
- Lots of people prefer Java to C++
  - More widely used
  - The two languages are similar in many ways
  - But usually C++ will be faster than Java

To code in C++

- You must know a solution in detail
- So you must understand the problem you are attempting to solve in detail

An insultingly simple (maybe) example

- Find the distance between two points
- Assume each point is represented as and (x,y) pair
- Then the distance can be computed as:

$$d = \sqrt{a^2 + b^2}$$

where a is $|x_1-x_2|$ and b is $|y_1-y_2|$

Note:

- To do this is C++, we only have
  - equations – but not really
    - assignment statements
  - if statements
  - while statements
- In this programming language everything must be expressed at this leval
- No knowledge of how to compute distances (or much of anything else)

C++ algebraic notation

- The first typing devices used with computers for writing programs required that everything be typed on individual lines
  - No subscripts
  - No superscripts (so nothing like x^2)
  - No special algebraic symbols (like√)

So some code pieces that do the computations

- From text, pg. 17

```
side_1 = x2 – x1;
side_2 = y2 – y1;
distance = sqrt(side_1*side_1+side_2*side_2);
```


Some C++ rules - 1

- The names of memory locations in which I can store the results of computations are called
  - **variables**
- There are rules about how you make up variables names
  - Mostly they must be letters and numbers (see text)
  - Many computer programs have 100's of variables so
    - use of one letter (like algebra) doesn't work
    - hard to remember what a variable is used for so it's a good idea to use a name that I can relate to
      - Doesn't matter to the c++ compiler, but people are often easily confused

More rules - 2

- Must tell the compiler:
  - The variables that will be used in the program
  - What kinds of things (integers, floats, letters, strings) will be stored in each variable name

More rules - 3

- Use comments
  - Again, like helpful variable names, not for compiler but for people who may need to understand the program

More rules - 4

- C++ has the facility to use program parts that others have written
  - Saves time; I don't have to write them
  - Lots of people need to do the same things so if written once, everyone can use them
- Examples:
  - How to send text to the computer screen, how to read what's typed on a keyboard
  - How to do some common mathematical computations like square roots

Let's do an example

Observations

- The C++ program doesn't look much like English. This is both good and bad.
  - English is much too ambiguous for scientific purposes.
- The C++ program doesn't look much like algebra. Sorry
  - Algebra used for equations and expressing mathematical relationships
  - C++ is a set of commands telling the CPU to perform some simple tasks.
    - Reserve some memory space for variables
      - The variable type tells how much space is needed
    - Include some libraries (that is, libraries of computer code previously written and stored on the computer's disk)
    - Do some computations and store the results in a variable
    - Test a condition and if true do something