Basics of Programming in C++
Computation

input(s) → Code → output(s)
What you are trying to compute

sub computation 1

\ldots

sub computation n

while
variables
functions
conditionals
arithmetic operations
\ldots
Structure of a C++ Program

int main()
{
    return 0;
}

Structure of a C++ Program

• Literals
  – Value written in source code
    26
    true
    “Hello, world!”

• Types
  – int – (e.g., 3, -5)
  – float – (e.g., 3.0, 5.3, -25.4)
  – bool – true/false

• Variables
  – Data storage
    <type> <identifier> [= <value>];
  – Declare, initialize
    int weight = 10;
  – Operators
  – Type safety
  – Constants
    const <type> <identifier> [= <value>];
Structure of a C++ Program

• Operators
  − Arithmetic: +, -, *, /, %

• Operands

• Expressions
  − Sequence of operators and operands that specifies a computation
  − Order of operations (precedence)

• Statements
Structure of a C++ Program

- **Syntax**
  - Legal expressions

- **Semantics**
  - Meaningful program
  - Program has intended meaning
    ```cpp
    int msg = "hello, world" + 5
    ```

- **Errors**
  - Compile time
  - Run time
Iteration

• Repeat a block of statements

• Examples
  – Do this until you are done
  – While the sun shines make hay

• while (expression) {}
Selection

• “Select” an action amongst a number of possibilities

• Example
  – If this is true, do something, else do something else

• Syntax
  
  ```
  if (expression) {
      ...
  } else {
      ...
  }
  ```
Function

<return type> <identifier> ( parameters )

• Example
  int main()
  {
    return 0;
  }

• Named set of statements used to accomplish a specific task
• Why?
  – Code used in multiple instances of our program
  – Allows a program to divided
    • Organization
    • Distributing work
    • Eases testing
Example

- Compute the maximum of two integers
Building and Running a C++ Program

Source Code (.cpp, .h) → preprocessor/compiler → Object code (.o)

Executable Program → linker → Libraries
Building and Running a C++ Program

• Pre-processing
  – The #include directive is an example of a pre-processor directive (anything starting with #).
  – #include <iostream> tells the preprocessor to copy the standard I/O stream library header file into the program

• Compiling
  – Syntax checking, translation of source code into object code (i.e. machine language). Not yet an executable program

• Linking
  – Puts together any object code files that make up a program, as well as attaching pre-compiled library implementation code (like the standard I/O library implementation, in this example)
  – End result is a final target -- like an executable program

• Run it!
Programming Strategies

- How do I go about writing a program?
- Top-down programming
  - Start with description and divide it into sufficiently small units corresponding to available components
- Bottom-up programming
  - Start with small components and build from them
What you are trying to compute

sub computation 1

…

sub computation n

while
variables
functions
conditionals
arithmetic operations

…
Next Class

- Quiz
- Debugging