COMSC-165 Lecture Topic 2, Basic C/C++ Control Structures

Reference
Deitel, chapter 3
Code samples

Algorithms
a procedure for solving a problem
  language independent
  worked out and tested in advance
  guides program writing
algorithm formats
  notes and diagrams
  logic flow charts
  pseudocode (recipe)

Control Structures
"sequential execution" and "transfer of control"
  the "GO TO" statement
  "structured programming" loses GO TO
two ways to transfer control
  selection statements (ifs)
  repetition statements (loops)
C and C++ "keywords", `if` and `while`

Our Basic Programming Tools (so far)
general tools
  `int` and `double` variables
  `string` variables
  assignment statements (with `=`)
  operators and keywords
    `if` statements (for bypass)
    `while` statements (for

Simple Selection With `if` and `else`
`if` statements
  simple `if`
    equals, less/greater than
  `if` code block -- syntax template
  `if`... else...
  `if`... `else if`... `else`...

a new tool: the conditional operator

```
cout << ((grade >= 60) ? "pass" : "fail") << endl;
```

Simple Repetition With `while`
"while-true" code blocks with "if-break"
  code blocks in `{` curly brace containers `}`
  `break;` // break out of loop
  `continue;` //skip to end of loop and go again
"while-condition" code blocks
  check for "break" before loop starts
count-controlled and event-controlled loops

Applications of Loops
boolean search loops
counting loops
validation loops
EOF loops

Tools For Debugging
syntax errors -- these prevent compilation
  code formatting
  alignment and indenting
  commenting out
  the "stare" method
  the "retype" method

logic errors -- compiles, but does not run right
  the `assert` function
  the `__LINE__` and `__FILE__` macros
  the `__TIME__` and `__DATE__` macros

debug line tracing:

```
cout << __FILE__ << ' ' << __LINE__ << ' ' <<  __TIME__ << endl;
```

reversing)
  console I/O
C-specific tools
  `atoi` and `atof` (cstdlib, with NO using std::)
C++-specific tools
  `cin` and `cout` (iostream: requires using std::)
  `setprecision` (iomanip: requires using std::)

- **C++11 `auto` Specification**

e.g. `auto i = 0;`
e.g. `auto s = string("Hello");`
e.g. `auto population = 7000000000L;`
lets compiler figure out the intended data type
