COMSC-165 Lecture Topic 3
Advanced C/C++ Control Structures

Reference
Deitel, chapter 4, appendix C
Code samples

for-Loops
special syntax to simplify while-loop
  built-in counters
  `for (int i = 0; i != 10; i++)` // adaptation of while-loop
  `for (int i = 0; i < 10; i++)` // common
  `for (int i = 0; i < 10; i += 2)`
  `int i; for (i = 0; i != 10; i++)`
  variations
  `for (;;)`
  `for (;( i < 10) && (!found);)`
  `for (int i = 0;; i++)`
  `for (int i = 0; i < 10; i += 2)`
  `for (;;);`
  code blocks in { curly braces }
...can be *any* data type, not just `int`
...not only for count-controlled loops!
  can be sentinel- (or event-) controlled
Nested loops
ANSI C
Code samples

C++11 Range for-Loops
no index specification, e.g.:
```
for (auto value: arrayName)
  cout << value << endl;
```

Currency Data Types
there are no currency data types
options:
  use `int` to track pennies
  use floating point (`float` or `double`)
    accommodate "round-off error"

```
if (total == 100.0) // may not work
...
if (99.999 < total && total < 100.001) // try this
```

Formatting Floating Point Values
applies to `float` and `double`
using `#include <iomanip>`
does *not* change the value
  just alters its appearance
avoid manipulator "fixed" -- hard to unset

```
// set to 2 digits after the decimal
cout.setf(ios::fixed|ios::showpoint);
cout << setprecision(2); // "sticky"
```

Formatting Floating Point Values *continued*

```
// 2-line code to unformat, so show a number "as is"
cout.unsetf(ios::fixed|ios::showpoint);
cout << setprecision(6); // resets to default

// round to the nearest to whole number
cout.setf(ios::fixed);
cout << setprecision(0);

// left- or right-justify in 10-character space
cout << setw(10); // not sticky
cout.setf(ios::left, ios::adjustfield); ...or... << left <<
cout.setf(ios::right, ios::adjustfield); ...or... << right <<
```

`ios` is in iostream
`setw`, `setprecision`, `left`, `right` are in iomanip
cannot easily format the "thousands comma"

EOF Loops
```
while (fin.good())
```

do-while Loops
special syntax to simplify while-loop
when if-break is *last* statement in loop
```
do {...} while (...);
```

switch Statements
special syntax to simplify if-else
when all conditions test the same variable
...and that variable is `int` or `char`
  broader definition in C11...
`case`, `break`, and `default` keywords
  all optional

Data Types
`int` (32 bit in VisualC++)
`long` (32 bit, +/- 2B range limit)
`long long` (64 bit)
`float` (32 bit)
`char` (8 bit)
`short int` (16 bit)
`unsigned char`, **`int`**, **`long`**
`double` (64 bit)
`long double` (80 bit)
climits and cfloat
variations among compilers (the `sizeof` operator)

Logical Operators
compound conditions
  AND (`&&`) and OR (`||`)
the unary "not" (or "negation") operator (`!`)

How to enter TWO values on the same console input line

```
double x;
double y;
char buf[100];
cin >> buf; x = atof(buf);
cin >> buf; y = atof(buf);
cin.ignore(1000, 10);
```

NOTE: no cin.ignore statement used after inputting x!