

today: functions

- more on functions (textbook chapter 5, sections 5-6)
- a few new things
- review for the midterm (chapters 1-5)

more on functions: reference parameters

- last class we talked about how functions can pass parameters
 - and how the value of those parameters might change inside the function
 - but in the *calling function*, the value of the parameters does not change
- we also talked about *scope*
 - and how variables are defined within either a *global* or a *local* scope
 - and how *local* variables, e.g., those that are defined within a function, “go away” when the function exits
- in C++, there is a feature of functions called *reference parameters* that lets you pass what is called the “address” of a variable to a function so that when the function exits, if the value of the variable has changed inside the function, then the new value can be retained outside the function

reference parameters: classic example

- the classic example of using reference parameters is a function called `swap()`

```
void swap( int &a, int &b ) {  
    int tmp;  
    tmp = a;  
    a = b;  
    b = tmp;  
    return;  
} // end of swap()
```

a few new miscellaneous things

- terminology: *associativity*, *escape sequences*
- compound assignment operators
- `#include <cmath>`
- formatted output (`cout.setf()`, `cout.precision()`, `cout.width()`)
- conditional operator (`?:`)

midterm review: chapter 1

- algorithm
- assignment statements
- arithmetic operations (+, -, *, /, %)
- output to the screen (cout)
- variables
- identifiers and keywords
- for loops
- increment and decrement shorthand (++ and --)
- comments
- *tracing* program behavior
- software development cycle: design, write pseudocode; write/save, compile, run; test, debug; re-write/save, compile, run; etc.

midterm review: chapter 2

- if statements
- real number data types: double, float
- character data type: char and ASCII
- arithmetic precedence ("my dear modern aunt sally") and "associativity" (left to right: * / % + -, right to left: +x, -x, ++, --)
- formatting output, endl
- "escape sequences" (\n, \t)
- relational operators (<, >, <=, >=, ==, !=)
- increment and decrement operators (++ , --)
- compound assignment operators (+ =, - =, * =, / =, % =)
- math library (math.h or cmath.h)

midterm review: chapter 3

- interactive data entry, keyboard input (cin)
- "prompting" the user
- while loops
- formatted output: cout.setf() and cout.precision() and cout.width()
- if-else statements
- if statements that contain multiple instructions
- conditional operator ? :
- file I/O

midterm review: chapter 4

- declaring a loop index
- defining a constant

midterm review: chapter 5

- library functions
- programmer-defined functions
- function *prototypes* and *headers*
- variable *scope*: local versus global
- `return` statements
- *arguments* or “parameters”
- formal, or “dummy” parameters
- void functions
- “parameterless” functions (a function that has no arguments)
- reference parameters