COMSC-165 Lecture Topic 4
Basics of C and C++ Functions

Reference
Deitel, chapter 5.1-5.11
Code samples
Tutorial

Functions
subprograms, a.k.a. methods, procedures,...
a way to do "temporary" transfer of control
purpose/advantages
- code reuse
- modularization

sources
- C/C++ Library
- your own functions
- 3rd party add-ons (e.g., SQL)

Built-in Common Functions
the `cstdlib` library -- a C-library
- ...so no `using std::` statements!

replaces `stdlib.h` from the C language
`atoi(const char*)` and `atof(const char*)`
`itoa(int, char*, 10)` integer to string *Microsoft only*

Built-in Math Functions In C++
the `cmath` library -- a C-library
- ...so no `using std::` statements!

replaces `math.h` from the C language
`sqrt(double)`
`pow(double, double)`
`sin(double)`, cos, and tan
`log(double)`, and log10

Value-Returning Functions
```
int fun()
{
  int result = a default value;
  ...
  return result;
}
```

Function Prototypes
not required -- but we'll use them in Comsc-165 anyway
useful in multi-CPP projects

```
int fun(); // the prototype

int main()
{
  ...
  cout << fun();
  ...
}

int fun() // the function definition
{
```

Local Variables
defined inside a function
have function "scope" only

Function Parameters
a.k.a. "arguments"
used to initialize local variables
- declarations are inside the ()'s

```
int avg(int a, int b) // function definition
{
  int result = 0; // a default value
  result = (a + b) / 2; // truncated fraction
  return result;
}
```

function "call": `avg(10, 20)`
- "a" gets initialized to 10
- "b" gets initialized to 20

coersion and promotion

parameters do not need names in prototypes
```
int avg(int, int, int);
```

C++ Standard Library
new with standard C++ (version 99)
iostream, iomanip, fstream, etc
the `std` namespace

Game Programming
using the "random number generator"
in `cstdlib`, `rand()` returns an int
...between 0 and 2 billion (RAND_MAX)
to limit the range (e.g., 1 to 6) -- "scaling"
`1 + rand() % 6` // 6 values, starting with 1
scaling a pair of dice
`2 + rand() % 6 + rand() % 6`

"seeding" the sequence
use `srand(time(0)); rand();` // requires `ctime` library
- execute just *once* in any program

`enum` Constants In C/C++
`enum Status{UP, DOWN, IDLE};` // can be global
`Status elevatorDirection = IDLE;`
- use "Status" as a data type

Scope And Lifetime
local variables in functions, loops, and ifs
- end at closing curly brace

curly-brace containers as local scopes

`static` variables in functions

```
  ...
}
```

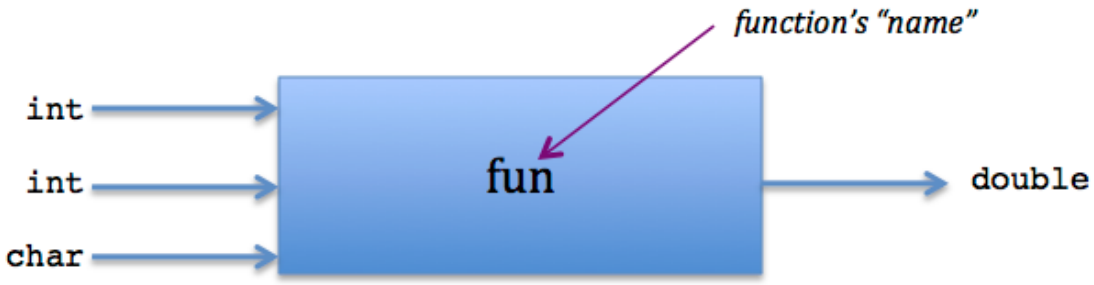
It's a *promise* that if I ever *call* this function,
...it will be defined somewhere
  either below...
  ...or in another CPP

```
int getNextInt()
{
  static i = 0;
  return i++;
}
```

returns sequence: 0, 1, 2,...

avoid "global" variables
  not a good way to avoid parameters

Intro To Functions



A "value-returning" function with 3 parameters (for input)

```
// "prototype"
double fun(int, int, char);

// function "definition"
double fun(int x, int y, char c)
{
  double result;
  ...
  return result;
}
```

A "void" function with 3 parameters (for input)

```
// "prototype"
void fun(int, int, char);

// function "definition"
void fun(int x, int y, char c)
{
   ...
}
```