

Chapter 1

CompCycle: Source, preproc, pp'd
src, compile, objcode, link in libs, exe

Chapter 2

Random

```
<cstdlib>
y=rand() pseudorandom
srand (seed-int) prime seed
dice = 1 + rand() % 6
```

File

```
<ifstream>
Declare: (if|of)stream f
f.open("file.dat")
a : \\dir\\file
```

Example	Base
0xF4	hex
0434	oct
0b111010	bin

Scientific Notation

1.98e30 : 1.98x10³⁰
1.98e-4 : 1.98x10⁻⁴

Char v. String

'A'	65
"A"	65 \ 0

Sizeof

```
sizeof(int)
sizeof(anInt)
```

Var. Rules

First char: a-zA-Z_
After first: a-zA-Z0-9_
Case-sensitive

Chapter 3

Casting And Constants

<cmath> to get C functions: sqrt(x),
pow(x,2.0)
cast: val = static_cast<type>(variable)
constants: const type CONST
preproc-constant: #define PI
3.14159

string objects can be concatenated
with '+'
See Tables below

Chapter 4

- Truth is nonzero, false is zero
 - if (some condition) activity;
 - Beware == float compares because of round off issues
 - if () /else if () / else
 - LogOp Precedence: !, &&, ||
 - Use arithmetic operations to compare char and string; ordered by ASCIIbetical order
 - switch (integer expression){
 case constant:
 break;
 ...
}
 - Be aware of fall thru in switches, think of the top as premium, it inherits at the top and should catch all of the attributes down to default or "base" model
 - enum Roster Tom, Sharon, Bill; Roster x; x=Bill;
- "parameters" are the formal arguments local to the function
 - Prototypes need only define types: "void x (int, float, float)"
 - 'static int x' would create a variable x that would persist between function calls
 - Pass arguments with defined parameters last
 - Pass a reference as such: prototype int f(int &)
function int f (int & q)
 - Pass a filehandle as such: prototype void readfile (ifstream &)
function void readfile (ifstream &file)
 - Functions can be *overloaded* provided they have differing parameter lists.
 - 'exit' comes with <cstdlib>
 - stubs display the data that was passed in: " I was given and I ran..."
 - drivers display the results that a function returned: "I got back this after I sent in ..."

Chapter 5

- Postfix uses the value of the – or ++'d number **first** and *then* increments / decrements
- While: while (expr) | *Pre-test loop*
- Do.: dowhile(expr); | *Post-test loop*
- break: aborts a loop | continue
Re-Loops
- For file read in and test, see tables

Scoping

- When a global variable exists by the same name as a function local variable name, the global variable cannot be reached, it is occulted by the local definition

Chapter 6: Functions

- Functions have prototypes and definitions
- The function must be defined before it is called OR its prototype must have been declared
- "args" are what's passed in when a function is called

Variables And Memory [CH. 3]

type	Size	Value Range
long double	16 bytes	<i>long double floating</i>
double	8 bytes	<i>double floating</i>
float	4 bytes	<i>floating</i>
unsigned long	4 bytes	[0,4294967295]
long	4 bytes	[-2147483648, 2147483647]
unsigned int	4 bytes	[0,4294967295]
int	4 bytes	[-2147483648, 2147483647]
unsigned short	2 bytes	[0, 65535]
short	2 bytes	[-32768,32767]

CStringManip (<iostream><iomanip>) [CH. 3]

assume	const int SIZE12;
initalize	char word[SIZE]"Hello"
set	strcpy(word, "Monkey")
get	cin » setw(4) » word
get	cin.width(SIZE); cin » word
get	cin.getline(Cstring, maxLength)

File test operations

```
#include <iostream>
#include <fstream>
using namespace std;
ifstream inputFile;
inputFile.open("datafile.dat");
if ( !inputFile )
    // Error, etc.
int number;
inputFile >> number;
while ( !inputFile.eof()){
    cout << number;
    inputFile >> number
}
```

String Reading <iostream> [CH. 3]

cin » var	Reads line. Wspace terms
getline(cin,var)	Reads entire line, w/s indiff.
cin.get(char)	Reads in a char, leaves \n in buffer
cin.ignore(num,'\n')	Ignore next <i>num</i> char or until <i>char</i>

Formatting <iomanip> [CH. 3]

Formatter	Description
setw(3)	sets width to be used in print, is right or l. justified
setprecision(3)	sets significant digits; stays in effect
fixed	show decimal, no sci. not.; stays in effect
left right	justifies; stays in effect