Midterm

1,2,3 and 4.1-4.10

Chapter 1:    
    Isolated points described in the slide set.   Can omit the rest of Chapter 1 in the book (i.e., those parts not covered in the slide set).  
  
Chapter 2  
    All sections  
  
Chapter 3  
    Sections 3.1, 3.2, 3.3, 3.4, 3.5, 3.6, 3.7 (nothing beyond Section 3.7)  
  
Chapter 4  
    4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.14 (skip Sections 4.12 (Comparing Characters and Strings), 4.13 (Conditional Operator), and 4.15 (Blocks and Scope).

**CHAPTER 1**

Main Memory

* RAM, volatile, running programs
* Direct access to any element
* Bit smallest (0 or 1)
* Byte, 8 bits with address

Puesdo Code

Machine Language

* 0010010

Compiler

* Translates programming language into binary sequence
* Final executable produced by linker

IDE

* Integrated Design Environments
* Write, compile and debug
* C++, turbo C++, Codewarrior

High level to Executable

* Source code – entered in text editor
* **Preprocessor** – executes on preprocessor directive (#include)
* Modified source code – removes all preprocessor directives
* **Compiler** – converts to binary
* Object code – not executable, unresolved references
* **Linker** – goes into C++ library to find things like cout
* Executable

Elements of a Program

* Key Word – reserved (using namespace)
* Program defined ID’s – variables from programmer
* Operators - +,-,\*
* Punctuation - , ;
* Syntax – rules of grammar, controls use of key words

Variable Declaration

* <data type><variable name>

Programming Process

* Define objective
* Visualize program running
* Pseudo Code
* Type one piece of code
* Save compile debug
* Repeat 4-6
* Run and Test
* Correct Errors, repeat 5-8
* Validate

**CHAPTER 2**

Parts

* // comments
* # preprocessor directive
* Using namespace (which namespace)
* int main {} (main function)
* { (beginning)
* cout (output)
* return (send 0 to OS)
* } (end)

Function

* Named sub routine

Cout

* Console output, steam object
* Must have <#include iostream>
* cout << “ “ ;
* Can send more than one item
* endl stream manipulator
* escape sequences (‘\n’),\t,\a,\b,\r

Preprocessor Directive

* Not part of C++ language

Compiler

* Translates code, w/out preprocessor directives, into object code

Object code

* Unresolved references, “holes”

Linker

* Links missing object code w/ code for missing functions to produce executable image

Variable

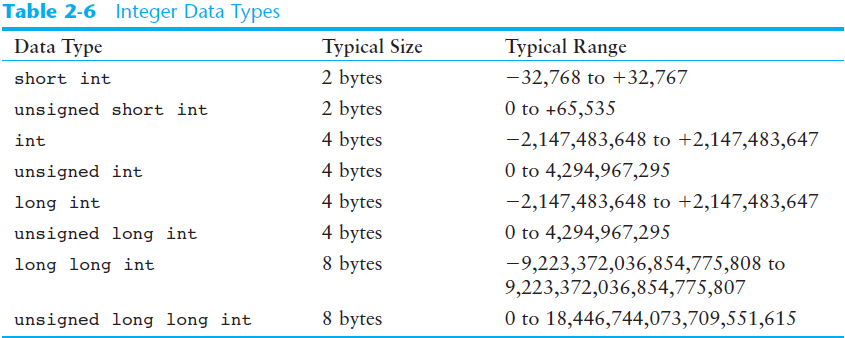
* Interface to RAM
* Declaration 🡪 int number
* Assignment 🡪number = 5
* Retrieve data 🡪 cout << number <<

Literal

* Value written into code
* “Hello, there” (string literal)
* 12 integer literal (int)
* 12.5 floating point literal (double)

Identifier

* Programmer defined name (variables, functions)



Integers

* Short int 2 bytes -32,768 to +32,767
* Unsigned short into 2 bytes 0 to +65,535
* int 4 bytes -2,147,483,648 to +2,147,483,647
* usigned int 4 bytes 0 to 4,294,967,295
* long int 4 bytes -2,147,483,648 to +2,147,483,647
* unsgd long int 8 bytes 0 to 4,294,967,295
* long long int 8 bytes -9,223,372,036,854,775,808 to

9,223,372,036,854,775,807

* unsgd long long into 8 bytes 0 to 18,446,744,073,709,551,615
* **SIGNED BY DEFAULT**
  + **-**2^(N-1) to 2^(N-1) – 1 \*\*\*where n is the number of bits
* **UNSIGNED**
  + 0 to 2^(N-1)

Integers

* Any numeric literal without a decimal point
* Type L at end to store in long member
* 0 at beginning is base 8
* 0x at beginning is base 16

Char

* Used for character or very small integers
* Usually 1 byte
* Numeric value stored in memory
* ‘C’ stored as 67 in memory

String Literals

* Series of consecutive characters stored in memory terminated by null
* \0

Char vs String

* ‘A’ is A
* “A” is A \0
* Recall a char can only hold one character
* **WE CAN’T STORE STRING LITERALS TO THEM BECAUSE OF THE NULL (\0)**

Floating Point

* Double
* Long Double
* Float
* All can accept real number (decimal) values
* Stored similar to sci notation (e notation)
* All are signed

Floating Point Literals

* Double by default
* Represented in fixed point (decimal) or e notation
* Can be forced float or long with F or L at end respectively
* Double to int results in truncation

Bool

* Condition that evaluates to T or F
* There is not Boolean data type
* So Represented by small ints
* Nonzero is T
* Zero is F

Size

* Sizeof()
* Returns size
* Sizeof(int) returns 4 on most
* Sizeof(double) returns 8 on most

Assignment

* Operand = operand
* lvalue = rvalue
* Something capable of receiving = something evaluated to single value

Scope

* Part of program in which variable can be accessed
* Not before initialization

Integer division

* Int / int the decimal portion is discarded
* Float / int or int / float decimal will be retained if it exists. No dec if no remainder

Modulus

* Requires integers
* Can we use to determine odd or even?

Named constants

* Cannot be changed during execution
* Integer literals – ints
* Floating point literals – default is double
* String Literals
* const <data type> <variable name> = rvalue
* name = ALLUPPER\_WITHUNDERSCORES
* **Must be initialized when declared**
* **#define <identifier> <replacement text>**

**CHAPTER 3**

cin

* Stream extractor
* Converts data being read to data type of variable holder
* User 🡪key board🡪 key board buffer🡪variable
* User enters sp sp sp 45 sp sp 5 . 1
* Buffer == sp sp sp 4 5 sp sp 5 . 1 \n
* Cin reads off first whitespace
  + Starts to parse on first token
    - Continues parsing until
      * More whitespace
      * Un par sable token
* Multiple cin lines will pick up where last left off
* \n always remains in the buffer

Whitespace

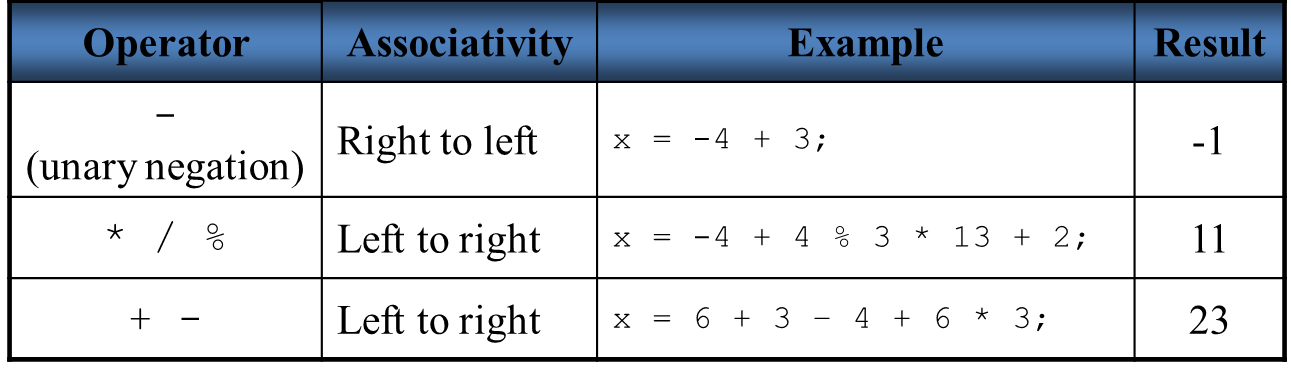
* Spaces, tabs and newlines

Tokens

* Stuff between the whitespace

Float cannot accept character literals

Mathematical operator precedence



No exponential operator

* Cmath header and pow function
* Pow(4.0,2.0) = 4.0^(2.0)

Type conversion

* Operations can be preformed on operands of differing types
* C++ promotes lower value
* Long double
* Double
* Float
* Unsigned long
* Long
* Unsigned int
* Int
* Char, short, unsigned short are automatically ints

Methods

* Coercion (automatic conversion of operand) 🡪char,short, unsigned short
* Promotion (lower to higher)🡪operation with double and float (promotes float to double)
* Demotion (higher to lower)🡪 assignment operator converts rvalue to match lvalue

Integer division

* Integer/integer (truncates any decimal remainder)(if stored to double, will tack on .0)
* Double / int (doesn’t truncate)
* Int / double (doesn’t truncate)

Over flow underflow

Typecasting

* Static\_cast <data type> (what casting)

Combined assignment operators

* +=,-=,\*=,/=.%=
* x += 1 is x = x + 1

**Formatting Output last subject**

Stream manipulators

Set precision 🡪 number of significant figures

Fixed 🡪displays x number of decimals, won’t force them

Showpoint 🡪 displays x number of decimals, will force them

Together 🡪decimal mode

**CHAPTER 4**

Precedence

Syntactically valid Identifier (no dollar signs, can’t start with number, no keywords)

Constant variable initialized when declared