CS180 知识点梳理

Chapter 1 - 3

- Elements of a program
- Data types
  - The data types is a set of value with a set of operations that define value
- Operators and scoop, const variable
- White space & keyboard buffer
- cin with the stream extraction operator
- Type casting and the formatting
  - setw() function, setprecision() function and the “fix” manipulator.
  - The static_cast<> casting method
  - srand(seed) and random number generator
    - rand() % (MAX - MIN + 1) + MIN

Chapter 2 - 4

- The if statement
  - if structure and the if-else structure and the if-elseif-else structure
- Relational and boolean topics
  - Truth tables

a	b	a && b
f	f	f
f	t	f
t	f	f
t	t	t

a	b	a || b
f	f	f
f	t	t
t	f	t
t	t	t

a	!a
f	t
t	f

  - Logical operators
  - String comparing
  - Boolean flag

Chapter 5 - 7

- While and do-while loop
  - Increment and decrement
  - Prefix and the postfix
  - While loop structure and the do-while loop structure
  - Counters and Accumulators
  - Control a loop with a boolean flag
- The for loop

    - for loop structure
    - Things to avoid in the for loop
        - Declare the loop control before the loop header
        - Missing parts of the loop header
        - Multiple statements in the loop header
        - Modifying the loop control variable in the loop body
        - Using a floating point to control a for loop
    - Nested Loops
    - Do not use any break, continue or return in the for loop
- Text file I/O
    - If stream & ofstream
    - File.open() & file.close()
    - getline() & cin.getline()
    - dummy variables
    - File open errors
- Functions
    - Function definition
    - Function prototype
    - Javadoc
    - Function call
    - Parameter
        - Formal parameter and actual parameter
        - Scope and local variable
    - return function and void function
    - global constants
    - Pass by value & pass by reference (pointer)
- Arrays
    - Array variable
    - Initialization and use it in the function
    - 2-D array
        - row & column
    - Copy array & read array
    - Output array
    - Const array in the function
- Vector
    - Vector initializations
    - Vector pushback
    - Vector in the array

Chapter 8 - 10

- Searching
    - Linear search
    - Binary search
- Sorting
    - Selection sort

- Bubble sort
- Pointers
  - Relation between pointers and the pass by reference parameters
  - Pointer variables
  - Draw a picture about how this work
  - The pointers and the arrays
    - Dynamically allocated arrays
      - Use the operator new to allocate a piece of memory at runtime
      - Remember to delete that after use
      - Diagramming dynamic memory
      - De-allocating memory
- Strings
  - Character functions

Function	Return Value
isalpha	true if the argument is alphabetic
isalnum	true if the argument is alphabetic or a digit
isdigit	true if the argument is a digit
islower	true if the argument is lowercase alphabetic
isprint	true if the argument is a printable character
ispunct	true if the argument is a punctuation character (includes braces, dollar sign, less-than, etc.)
isupper	true if the argument is uppercase alphabetic
isspace	true if the argument is whitespace

  - C-strings
    - Null character
    - “char” type
    - C-string library functions
      - strlen: the number of characters before the \0
      - strncat: concatenate two strings together
      - strncpy: overwrite one string with another
    - strcmp: a function used to compare two strings are identical
    - strstr: find the location of a substring within a string
  - Converting from c-strings to numbers
    - Use cstdlib library
      - atoi() stands for “ASCII to integer”
      - atof() stands for “ASCII to float”
  - Arguments to main function
    - Initial in the terminal before the program run
    - e.g.
      - Int main(int argc, char* argv[])
    - argc: an integer that is the count of arguments
    - argv: an array of strings, stands for “vector of arguments”
  - From number to string
    - Use to_string() function to let the number transfer to the string

- Use .length() can get the length of the string
- Use .substr() can extracts a substring
- String :: npos stands for “no position”

Chapter 11 -12

- Structs
  - Abstraction is a model defines the common characteristics of some things
  - Abstract data type is defined by programmer
  - Basically is defined a series of new data types and variable that can be use in the program
  - Construct a structs and initializing a struct

```
struct Part
{
  string description;
  unsigned current_quantity;
  unsigned min_quantity;
  unsigned max_quantity;
};
```

    - Use of the struct
    - Passing structs to functions
    - Copy struct variables
      - Just use the. “=“ is fine
    - Comparing struct variables
      - Well, you cannot compare like movie1 < movie2
      - But you can compare like movie1.length < movie2.length
    - Pointer to the structure variables
    - Dynamically allocating structures
    - Overloading
      - Signature
- Binary, decimal and hexadecimal
  - Binary to decimal algorithm
  - Decimal to binary algorithm
  - Binary to hex, decimal to hex, hex to binary, hex to decimal algorithm
- Binary Files
  - ASCII Encoding Method

Binary	Decimal	Category
0000 0000 – 0001 1111	0 – 31	control characters
0010 0000	32	space
0010 0001 – 0010 1111	33 – 47	punctuation
0011 0000 – 0011 1001	48 – 57	digits
0011 1010 – 0100 0000	58 – 64	punctuation
0100 0001 – 0101 1011	65 – 90	uppercase
0101 1100 – 0110 0000	91 – 96	punctuation
0110 0001 – 0111 1001	97 – 122	lowercase
0111 1010 – 0111 1110	123 – 126	punctuation
0111 1111	127	delete

- Read the binary files from the disk
  - Use fstream library

```
fstream file;
file.open("bindata.dat", ios::out | ios::in | ios::binary);
```

  - Everything must be converted to or from char, use reinterpret_cast to accomplish that
  - Remember to close the file before the program ends.
- Converting strings
  - From C-String convert to the C++ String use cppstr:
    - e.g. string cppstr = cstr;
  - Extract C-string from C++ string using cppstr.c_str();
- Seeking
  - a binary file has a read marker and a write marker, the read marker is positioned with the seekg (for get) function and the write marker is positioned with the seekp (for put) function

```
file.seekg(2UL * sizeof record);
file.read(reinterpret_cast<char*>(&record), sizeof record);
```