COMP2113 Programming Technologies / ENGG1340 Computer Programming II

Summary

Note: The purpose of this summary is to highlight concepts covered in this course. You should refer to the learning materials concerned if you cannot recall the concept and syntax.

Module 1: Linux Environment

- Basic Linux commands (e.g., `cd`, `ls`, `cp`, `mv`, `mkdir`, `rmdir`, `rm`, `cat`, `chmod`, `man`)
- Advanced Linux commands (e.g., `grep`, `wc`, `cut`, `diff`, `vi`)
- File redirection using < and >

Module 2: Shell Script & Version Control

- Variables
- Input and output
- Quoting (single quote vs. double quote vs. back quote)
- Command substitution
- String manipulation
- Flow of control using if statements and loops (recall how to check file existence, successful compilation etc.)
- Version control using Git

Module 3: C++ Basics

- Variables, constants, operators, data types, input and output
- Flow of control using if statements and loops

Module 4: Makefile

- Define dependency (e.g., `abc.o: abc.cpp abc.h` → `abc.o` depends on `abc.cpp` and `abc.h`)
- Teach Linux how to get the target (write down the command below the dependency line with a tab at the beginning)
- Variables (e.g., `$(FLAGS)`) and special variables (e.g., `$@`, `$^`, `$<`)
- Phony targets: not actually a target but can use `make` to call it (e.g., `make clean`)

Module 5: C++ Functions

- How to define a function and how to call a function
- `void` function vs. function that has return value (note how to define and how to call them)
- Scope of variables (e.g., local variables and parameters of a function)
- Pass-by-value (function cannot update the value of a parameter) vs. pass-by-reference (function can update the value of a parameter)

Module 6: C++ Arrays & Strings

- Array of any data type (how to define and how to use)
- `char` array (also known as C-string)
- C++ string (functions available such as `length()`, `substr()`, `find()`, `rfind()`, `replace()`)

Module 7: Structs, File I/O & Recursion

- `struct`: customized compound data type containing 2 or more member variables. Member function can also be defined.
- `class`: fundamental unit of object-oriented programming. A class can contain class/instance variables/functions.
- File input and output (`ifstream`, `ofstream`, `open()`, `fail()`, `close()`, `fin >>`, `fout <<`)
- `istringstream`: lets you extract strings from a long string using >> in which words are separated by spaces
- Output formatting (e.g., `showpoint`, `fixed` / `scientific`, `setprecision`, `setw`, `setfill`, `left` / `right`)

Module 8: Pointers, Dynamic Memory & Linked Lists

- Pointer: holding memory address of a variable
- Use `*` to declare and use & to dereference
- Use `->` to access member fields of struct pointer variable
- Can use pointer variable and `new` keyword to declare dynamic array (size can be specified at runtime)
- Linked list: basic unit is a `struct`, which contains `next` pointer pointing to next node
- Recall some basic operations: list traversal, head insertion, tail insertion, node deletion, release memory

Module 9: C++ Standard Template Library (STL)

- 3 containers:
  - vector = dynamic array
  - list = doubly linked list
  - map = balanced binary search tree
- Access items using iterators (similar to pointer and with `*` notation)
  - Useful functions: `push_back()`, `begin()`, `end()`

Module 10: C programming & Debugging

- C is like a subset of C++ language
- Basic input / output using `scanf()` and `printf()`
- No `string` class in C and can only use `char` array
- Flow of control using if statements and loops
- Function and array
- Dynamic memory allocation using `malloc()` which returns a pointer. Note the use of `sizeof()` and `free()` functions.
- C `struct`, `typedef` keyword
- GDB debugger