CS246 Review Notes Midterm Edition

Preamble

Welcome to the Midterm addition of the CS246 course notes that I will be creating and updating through the term. These notes are not aimed to replace lectures, tutorials, labs, office hours with your professor or TA, Piazza, or any other source directly provided by the course and the University of Waterloo. These notes are for the sole purpose of reviewing and as a resource for students interested in the course.

The course material that has influenced the contents of these course notes are as follows:

- Lectures run by Brad Lushman, PhD
- "Absolute C++", textbook for the course.
- Posts on the CS246 Piazza.
- Student perspectives.

To preserve the chronological order that the material was presented during lecture, these review notes will follow a similar structure. However, some material in these review notes may be introduced earlier or later. This will only occur if we feel as if material flows better in a separate section than it was originally introduced in.

Practice exercises can be found scattered throughout each section of the review notes. There is a special section at the end dedicated to sample problems based on emphasized course material, struggling assignment questions and other factors.

Disclaimer

I ask that you respect all of the guidelines I set forth with respect to these review notes. If you are unable to respect any of the guidelines mentioned in the proceeding few paragraphs, please delete this file or close the hard copy of the review notes.

1. I alone reserve the right to alter and distribute these review notes. If you see a mistake or feel as if important content is missing, contact me (information provided below) so that I can make the appropriate correction(s).
2. I will be supporting the accuracy of these review notes for the remainder of the W14 semester. After such a time these review notes should be considered obsolete.
3. These review notes are provided absolutely free of charge. If you paid for a hard copy or e-copy of these review notes then you are obligated to contact us so that we can take the appropriate action(s) towards the distributor.

Keep in mind that these notes are in no way guaranteed to be accurate. There may be mistakes, outdated information or tangents that will not be directly related to some of the material presented in the course. These notes have been developed by students during our undergraduate year taking CS246. No professor, TA or anyone involved in the administration at the University of Waterloo has endorsed these notes.

If you feel a need to contact me at any time for clarification, issues with the notes whatever, you can reach me at my contact information below:
Jacob Willemsma, author (**wjwillem@uwaterloo.ca**).

Acknowledgements

Some further acknowledgments to the CS246 material and other influencing figures.

- Professors and Teaching Assistants for the summer semester of 2014: Brad Lushman and Kirsten Bradley.
- Design inspired by "Linear Algebra Course Notes" by Dan Wolczuk (with permission).
- CS246 Course Notes F13 written by Liam Horne.

Table of Contents

1 Introduction

The official catalog entry for CS246 at the University of Waterloo is as follows:

DEFINITION
CS246

This course introduces students to basic UNIX software development tools and object-oriented programming in C++ to facilitate designing, coding, debugging, testing, and documenting medium-sized programs. Students learn to read a specification and design software to implement it. Important skills are selecting appropriate data structures and control structures, writing reusable code, reusing existing code, understanding basic performance issues, developing debugging skills, and learning to test a program.

In the prerequisite course – CS136, the focus was programming in a functional and imperative paradigm. However in CS246, the focus has shifted to programming in an object oriented paradigm in C++.

DEFINITION
Functional Paradigm

The process of evaluating code as a sequence of Mathematical functions, avoiding mutation and the manipulation of state. It has an emphasis on function definition, function application and recursion.

DEFINITION
Imperative Paradigm

The process of evaluating statements that change a programs state. This is done through mutation and control flow.

DEFINITION
Object Oriented Programming (OPP)

The process of representing concepts as "objects" that have data fields and associated procedures (or functions) known as methods.

1.1 Syntax

We now are told to quickly change languages to C++, oh no! What about my precious syntax knowledge. Not to worry, the C++ gods had this all covered. Nearly all C programs you've ever written have the capability to run in C++, so all that precious knowledge we picked up in CS136 has not gone to waste. In fact, it is in most cases the same! The following code is in "C++", but you'll notice it's no different than C.

EXAMPLE 1

The following function, f consumes two integers, x and y and produces the sum of x and y.

```
int addTogether(int x, int y) {
  return x + y;
}
```

Not bad eh?

However, in this course we will not begin with C++, instead we will familiarize ourselves with the Linux environment first. What is Linux you may ask? When you hear Windows you may think of a bunch of different operating systems under the same "umbrella": Vista, 7, 8, XP, 98 (if you're really old). Linux is almost exactly like that, however all the different distributions (distros) of linux are not owned by the same entities and in fact, are generally free and open source software!

DEFINITION
Linux

Is a UNIX like computer operating system under the model of free open source software. It was first pioneered by our lord and saviour Richard Stallman during the founding of the GNU Project.

DEFINITION
Free and Open Source (FOSS) Software

Software that any is freely licensed to use, copy, study and change in any way and where the source code is openly shared and improved.

DEFINITION
Distributions

A Linux distribution is a version of Linux. Since it is FOSS, many people have created their own versions in order to specialise specific tasks, whether those be: system usability, security, server hosting, et cetera. All these different versions are classified as distributions.

1.2 Getting Started

For me, this is always the most daunting task at the beginning of a CS course and I think it may be like this for a number of students. As a result, I've put together a setup guide to get the proper environment running on your computer!

Getting a Terminal

If you are using a Macintosh computer, you're already mostly finished! Look at you and your shiny Starbucks/Facebook machine go! The only step you need to do at this moment is to go to Utilities inside the Applications folder and locate the "Terminal".

Now, if your machine didn't cost you as much as a used car, you're probably on a PC. In this case, there is a little more work to be done! You have a few options, you can install and dual boot a Linux instance on your machine, or use a program to connect to the university's Linux servers. Personally, I prefer the first option, but I will set you up on the second option as well.

The first step to dual booting a Linux distribution is to pick which distribution you'd like to run. Personally, I think for beginners Ubuntu is the most familiar and simple to use. Head on over to `http://www.ubuntu.com/desktop` to download the latest version.

The install will guide you through the process of partitioning your hard drive. When allocating space, you shouldn't need any more than 10GB, unless you wish to use it as a primary operating system.

Once installed, when booting up your computer you will have the choice of either Windows or Ubuntu. When in Linux, at any time holding Control + Alt + T will open your terminal!

An alternative to dual booting is to install Cygwin or Putty. I'll leave the installation up to you!

2 Linux and the Shell

You're finally in the world of Linux. It's ok if you're a little lost and scared. That's normal.

For the purposes of this course, we will be spending most of our time in something called the shell.

DEFINITION
Shell

The Shell is an interface for interacting in the Linux operating system. It is used for everything from creating files, to connecting to servers to writing code. There's are two types of interfaces, graphical and command-line. The shell is a command-line interface.

2.1 File Systems

The file system present in Linux is very similar to what you are used to in other operating systems. The file system is set up in a tree-like hierarchy with **directories** and **files**. It is also possible to have directories inside of directories.

DEFINITION
Path

A path is just a way of illustrating where you are in the file system. This is represented in the shell of a single line of directories and subdirectories separated by a forward slash. There are two types of paths, absolute and relative. The absolute path is the list of directories from a special directory called the **root** note. The relative path is the path relative to the working directory (not necessarily the root). Typing *pwd* at any time in the shell with output the absolute path.

Getting around in the shell is another task that requires a little bit of learning. The command *cd* which stands for change directory does exactly that.

1. cd "directory name" - changes the current directory to the specified directory.
2. cd .. - changes the current directory to the parent directory.
3. cd - changes the current directory to the home directory.

The next handy command is *ls* which stands for list files. When in your shell, writing *ls* at any time will list all the directories and files in the currently directory. All the commands for *cd* also work as expected for ls.

Another quick tip for Linux is that calling *man "command name"* will open the manual for any given command. This manual will have a user guide for the command including how to properly use it and all the different special add-ons with the command (there are many).

EXAMPLE 1 Moving through a file directory in Linux

```
cd cs246/1141/a1        Changes the directory to assignment 1 of cs246.
ls                      Lists all the files in the present directory
a1q1.cc a1q2.cc test1.in test1.out
cd ..                   Changes the directory to the parent directory.
pwd                     Lists the present working directory.
/u/wjwillem/cs246/1141/
```