COMP 1006 — Introduction to Tools and Techniques in Computer Science 2

Course Description

Calendar entry

This is a lab-based course. Every Computer Scientist needs to make use of an expansive set of modern computing tools and techniques. This course continues the hands-on experiential introduction to working with the tools and techniques we use every day to design, develop, analyze, and maintain software.

Prerequisite: COMP 1002.

General Course Description

COMP 1002 introduces lots of new (to you!) ideas about computing that aren't always covered in other courses, but are important for your success, both in courses and in the workplace.

We're going to keep going with this idea in COMP 1006, starting with thinking about some strategies for debugging, using build and dependency management software, continuing to improve our skills and experiences on the command line, and then ultimately moving toward some more advanced topics that include administering a local Linux system (one you install on your own laptop in either a "virtual" or physical environment) and administering remote Linux systems (ones that are either virtual or physical but are located somewhere else and owned by someone else).

Detailed Prerequisites

Before entering this course, a student should be able to:

- Install software on their computer and write structured text.
- Connect to other computers and transfer data between computers.
- Use command-line tools to manage files.
- Compile and run programs written in different programming languages.
- Use version control software.

Course Goals

By the end of this course will:

- Apply strategies for debugging software and systems.
- Describe and use build and dependency management software.

- Use the command line effectively.
- Install and administer Linux systems, both locally and remotely.

Learning Outcomes

Debugging (strategies and usage)

Students should be able to:

1. Describe what debugging allows you to observe about a running program.
2. Use log-based debugging to identify a problem with code.
3. Use a visual debugger to set breakpoints, inspect values, step through a running program, and make observations about a program when it crashes.
4. Use a command-line debugger to set breakpoints, inspect values, step through a running program, and make observations about a program when it crashes.

Command-line environment and back ups

Students should be able to:

1. Redirect the output from a program to a file.
2. Redirect a file to be the input for a program.
3. Send data between programs using a pipe.
4. Choose an appropriate back up strategy for protecting your files.
5. Install and configure software to back up your files to a remote server.
6. Restore and verify your back up from the remote server.

Shell scripting

Students should be able to:

1. Write and run shell scripts to repeat complex tasks.
2. Use command-line arguments in a shell script.
3. Use loops, conditional statements, and variables in a shell script.
4. Use "globs" in a shell script to apply operations to files.
5. Apply strategies to debug shell scripts.

Build systems and dependency management

Students should be able to:

1. Download and run a program that requires you to install dependencies.
2. Create a `Makefile` to accomplish a task (building software, running commands).
3. Create a new empty software project that includes a build and dependency management tool.
4. Add and use a new dependency in a software project.

Containers, virtual machines, and remote systems

Students should be able to:

1. Install and configure virtual machine software.
2. Download and use a pre-installed operating system in a virtual machine software.
3. Run a command within a container.
4. Initialize a server in a publicly available cloud computing environment (e.g., AWS).

Running your own Linux system

Students should be able to:

1. Choose a Linux distribution.
2. Install the Linux distribution either on a personal laptop or in a virtual machine.
3. Install packages using a package manager (command-line and GUI).
4. Configure and run a web server.