EECS 268: Spring 2009
Laboratory 0: C++ Review and Introduction to Linux

Due: 11:59:59 p.m. the day before your lab that meets the week of February 2

Lab Topics: Linux Development Environment. Javadoc Comments. Submission Requirements.
Review Topics: Pointers. Dynamic Memory Allocation. Syntax, Runtime, and Logic Errors.

1. Introduction

In this lab, you will learn how to develop C++ programs in a Linux environment. You will be using g++, make, and gdb to compile and debug C++ programs. This lab will also introduce the Javadoc method of documenting files, functions, and classes.

2. Linux Development Environment

This lab introduces the Linux development environment for editing, compiling, and running C++ programs. We recommend using the standard text editor (Gedit) for editing your C++ source code. By default, Gedit provides several useful features, including syntax highlighting. Additional information on configuring Gedit can be found here: http://people.eecs.ku.edu/~jvalland/notes/gedit.html. If you are already familiar with other Linux editors, such as vi or Emacs, you may use those as well.

Throughout the semester, you will be using **g++** to compile and debug your C++ programs. You will learn how to use **g++** to display extra warning information (-Wall) and to include the debugging information that is necessary for gdb (-g). You will also learn how to use **make** to simplify the compilation of large projects.

The Linux Graphical Debugger (**gdb**) will be used to step through the execution of your program. You will learn how to find and correct runtime and logic errors by tracking the execution and variables of your program as you process input, call and return from functions, process loops, etc.

The Linux archive utility **tar** will be also be introduced. All projects are required to be submitted in the form of a tar archive.

More information on these tools can be found here: http://people.eecs.ku.edu/~jvalland/labs/intro.html.

3. Javadoc Comments

This class will require a fairly significant amount of documentation. Most of this documentation should be provided in the Javadoc format, which is introduced on pages 44-45 of your textbook. A complete listing of the Javadoc format is provided in Appendex F. Full Javadoc comments will be expected at the top of each **file** and before all **functions** and **classes**.

Javadoc comments will usually amount to between 15% and 35% of the total lab grade. This reflects the importance that this class places on documentation, and it should encourage you to spend extra time documenting your programs.

4. Submission Requirements

All labs are to be submitted to your lab instructor by email. Labs should be submitted as a tar archive containing only your source files (.cpp, .h) and make files. The KU email system frequently drops attachments it thinks may be dangerous, so take care that all executables and object files are removed from your lab prior to archiving and submitting it. We recommend including a “clean” target in each of your make files that you can easily run before submitting your labs:

```
clean :
    rm -f *.o main
```

This can be run from the command line as:

```
$ make clean
```

5. Grading Criteria

Grades will be assigned according to the following criteria:

20	**Correctness**
	- Demonstrated compilation using g++ and make.
	- Fixed existing syntax, runtime, and logic errors using g++ and gdb.
15	**Documentation**
	- Reviewed Javadoc syntax (pp. 44 – 45, Appendix F)
	- Completed Javadoc syntax for all files and functions in the provided source code.
15	**Submission**
	- Created tar archive containing ***only source code*** (.cpp, .h) and ***make files***. **Note:** Executables and object files (.exe, .o, etc.) should not be included.
	- Tar archive emailed to mcalnon@ku.edu or vlandham@ku.edu.
50	**Total**