Syllabus - COMP 312
Open Source Practicum
Dr. William L. Honig

Course Description
This course is a seminar for those who want to learn about Free and Open Source Software and the kind of apps, projects, and tools your can build with them. This course is a mix of philosophy, system configuration/installation, software development/hacking, and computer science.

The course is run as a seminar, so there is much self-directed effort required in this course. There will be lectures; however, the lectures are designed to be general and motivate student project work, which is a required component of the course (40% of your grade). The general plan is for there to be lectures, guest speakers, and project / group work most days. Teamwork is required as part of this work. FOSS is all about community and working with others to get stuff done. If you don’t enjoy teamwork and collaborating with others, this course is not recommended for you.

Prerequisites and Key Information
COMP 271 Object Oriented Programming; COMP 313 Intermediate Object Oriented Development recommended: Students should have an advanced knowledge of software development, ideally using Java or similar languages, and proven skills in writing, compiling, and debugging software. Students must also have the style and communications skills to work effectively with a group of other students.

Textbooks
None required. Web materials will be used.

Academic Honesty
Students are expected to have read the statement on academic integrity available http://www.luc.edu/academics/catalog/undergrad/reg_academicintegrity.shtml. This policy applies to the course. The minimum penalty for academic dishonesty is a grade of F for that assignment. Multiple instances or a single severe instance on a major exam or assignment may result in a grade of F for the course. All cases of academic dishonesty will be reported
Academic dishonesty includes, but is not limited to, working together on assignments that are not group assignments, copying or sharing assignments or exam information with other students except in group assignments, submitting as your own information from current or former students of this course, copying information from anywhere on the web and submitting it as your own work, and submitting anything as your own work which you have not personally created for this course. If you do wish to use materials that are not your own, please check with me ahead of time and cite your source clearly.

Course Grading
Your grade will consist of three components with relative weights as follows (I reserve the right to adjust the percentages in your favor if circumstances warrant):

Custom Open Source Project (40%)
You will define, complete, document, and make available to others a software or systems project as a key part of the course. You may work with others inside or outside the course; multiple person projects will need to be larger and more complex than single person projects. All projects will be based on and use open source approaches and philosophy.

Personal Participation (30%)
The participation grade will be based upon attendance and contributions to discussions in class, and other individual course assignments. As a seminar course, learning depends on your active participation. You will make presentations in class, report on your investigations and study of open source, and present your custom project in class.

Disseminations (30%)
Your project information on an appropriate site, the quality of the documentation in your code, and your final report and presentation of your project. Your contributions to course materials, class notes, class web.

The course will be graded on a curve based on performance of all students.

Participation and contributions in class are an important part of learning. All students are expected to attend all classes in person. Attendance and valuable participation in class discussions (in the classroom and in the discussion boards) are a factor in your course grade.

If for any reason you do miss a class session, it is your responsibility to determine what you missed, locate any handouts, determine any changes in assignments, course plans, or schedules, etc. It is not my obligation to help you make up for missing class. A key part of the class will be your participation and understanding of class room work. You will be at a great disadvantage if you miss a class.
Please do not ask for “extra credit” to improve your grade as this is neither practical in the course nor fair to your fellow students. I will be happy to discuss your performance in the course with you at any time, including discussing possible grade based on performance to date and ways to improve your performance during the remainder of the course.

Grades on each assignment are posted in Blackboard. Blackboard also allows you to see your current total and an estimate of your current total percentage. Points on assignments and grading comments are to be reviewed promptly by the student. To encourage your use of the grading comments to improve learning, requests for further explanation or adjustments to your score must be made no later than two weeks after the points are posted; after that time the grading is final and cannot be changed.

**Timely Completion**

Late work will not be accepted. Exceptions only by prior approval and will rarely be given. “Time is of the essence of this course”.

**Continuous Improvement**

I believe in a personal quality process of continuous improvement. Anything can be improved by applying the quality process of “Plan, Do, Check, Act” (PDCA). To improve the course and the learning of these computer science concepts, I welcome your feedback, comments, suggestions, and complaints at any time.

In support of this PDCA process, I may ask you to participate in surveys during the course. These surveys will measure student impressions of the course; when time permits I will share the results with the class. Your inputs on these surveys are anonymous and in no way affect your grade.

Version 1 (original: 20 Aug 2012)