REMINDER: The information presented in this syllabus is subject to expansion, change, or modification during the semester.

Instructor and Class Details

Professor --
Office: --
Email: --
Phone: --
Office hours: --

Course Text and Materials
Possible Texts:

- The Success of Open Source – Steven Weber – ISBN: 978-0674018587
- Open Source 2.0: Continuing the Evolution – Chris DiBona, Danese Cooper, Mark Stone – ISBN:978-1171648161
- The Cathedral and the Bazaar – Eric. S. Raymond – ISBN: 978-0596001087
- Producing Open Source Software – Karl Fogel – ISBN: 978-0596007591
- Rapid GUI Programming with Python and QT – Mark Summerfield – ISBN: 978-0132354189
- The Linux Programming Interface – Michael Kerrisk – ISBN: 978-1593272203

Suggested support:

- Mailing lists
- IRC Channels
- Blogs
- Planets
- fedoraforum.org
- ubuntuforums.org
- linuxquestions.org
- ubuntuguide.org
- github.com
- bugzilla.redhat.com

Course Description

Students will be introduced to open source software development concepts and Linux platforms. Students will learn to use the Linux command line, edit configuration files and setup systems for development. Students will learn how to design, package, release and maintain open source software. Topics such as open source communities, Linux packages, package managers, version control systems, software development tools, licensing, releasing, bug tracking, maintenance, patching and future development, will be covered in lectures, exercises and a project. Students will demonstrate competency with the Linux command line, Linux development and software development.

Important RIT Deadlines

Last day of add/drop week is January 0, 20xx.

Last day to withdraw with a grade of "W" is January 0, 20xx

NOTE: IGM department policy states that a student has one semester to **challenge** any **grade**. After that, grades cannot be challenged.

Prerequisites

These prerequisites are necessary and will be strictly enforced.

Interact Des & Alg Prob Solving Sequence

- IGME-101 & IGME-102

or

Game Dev & Algo Prob Solving

- IGME-105 & IGME-106

or

Equivalent programming sequence.

Role of course in curriculum for

Game Design & Development: This is an elective.

New Media Interactive Development: This is an elective

Course Goals and Objectives

General Course Goals
The student will:

- Be introduced to the basic concepts of Linux and Unix
- Be familiar with the history of open source
- Understand the fundamentals of licensing, trademarks, software ownership and rights
- Understand the basics of open source economics and the relationship between open source and businesses.
- Understand open source communities, societies and politics
- Examine how to use Linux for development and services
- Explore how to get involved in the open source community (IRC, Blogs, etc)
- Explore high level process of open source software development
- Examine Linux package types and package managers
- Explore how to use open source version control software and compare various packages
- Explore open source software development tools and potential benefits of each (such as portability)
- Examine the publishing process and release cycle using hosting facilities
- Examine the bug tracking process, maintenance and patching
- Explore future development and project forking

Course Organization

Midterm exam
A midterm exam will be scheduled during class time.

Project(s)
A significant project will be assigned that will require students to design and develop an open source software package leveraging other open source software, open source libraries or, with needed more stringent requirements, from scratch. Every project will need approval from the instructor.

Exercises
Some class periods may be reserved for the project. Plan on attending the class periods in order to complete the exercises. Some exercises may take longer than one class session to complete. You may be required to plan time outside of class to complete them. All exercises will be collected and graded. You will need to be familiar with the exercises in order to do well on the exam.

The bonus questions ARE NOT OPTIONAL for graduate students. To get full credit, graduate students must complete the bonus questions.

Grading

The grading scale used along with the grading criteria is as follows:

Component	Weight
Exercises	20%
Exam	25%
Homeworks	20%
Project(s)	35%
Peer Evaluation (if applicable)	10%

Range	Grade
>=90.0%	A
>=80.0% & <90%	B
>=70.0% & <80%	C
>=60.0% & <70%	D
<60.0%	F

Academic Honesty Policy

My policy on academic dishonesty is simple: If you cheat/plagiarize, you get an "F" as a grade for the course, a letter detailing the incident goes into your folder, and you are immediately removed from the class. (If this is a second occurrence during your career at RIT, the penalties are harsher.)

Note that if you get accused of cheating, I have already documented the offense and had the evidence reviewed by other faculty members to verify it will withstand an appeal.

Please review RIT policy on academic dishonesty:
http://www.rit.edu/studentaffairs/studentconduct/rr_academicdishonesty.php

Except for assignments that are specifically designated as being "group efforts," all work submitted (assignments, projects, participation activities, bonus opportunities, examinations, etc.) under your name is assumed to be your own individual effort and will be graded as such under the IGM's Academic Dishonesty Policy.

Submission of coursework under your name to your professor indicates that you understand and agree to abide by all relevant dishonesty policies.

Course Schedule (Subject to change)

Week	Day 1	Day 2	Activity
1	Intro to course, intro to the project, language overview, Brainstorming process, getting involved in open source community	Exercise 1: Brainstorming project ideas	Project Idea, team formation and Open Source Software research assignment.
2	History of open source, how open source works, open source licensing, trademarks, software ownerships, rights, Intro to open source software development and software development cycles	Exercise 2: Open Source Community involvement and data gathering	Community involvement Exercise and project planning
3	Open source relationship with business and proprietary software. Open source communities, societies and politics	Exercise 3: Setting up Linux environment, intro to command line, package installations	Plan for development environment; language, database, etc
4	Software development tools, frameworks, portability and IDEs	Exercise 4: Using development tools and frameworks; exploring IDEs	Work on Project
5	Command line discussion, scripting, intro to configuration files	Exercise 5: Scripting for development	Work on Project
6	Version control software and project planning	Exercise 6: Configuration files and installing/using version control software	Work on Project
7	Code Documentation, release documentation and generating man pages	Exercise 7: Documentation and Code Review	Study for Exam
8	Exam Review	Exam	Work on Project
9	Linux distribution differences, specialty distributions, building RPMs and DEBs	Exercise 8: Building and installing RPMs and DEBs	Work on Project

10	Integrating with services; databases, mongodb, d-bus	Exam 9: mongodb, mysql, SQLite and d-bus	Work on Project
11	Integrating with services; Apache and Django	Exam 10: Apache, Django and mongoDB; simple project web page	Work on Project
12	Publishing process, code hosting, bug tracking process, project forking, project merging	Packaging and pre-publishing	Work on Project, submit package to mycourses
13	Future development, maintenance, patching	Exam 11: Setup bug tracking, patching, forking and merging projects	Work on project
14	Code review, documentation review	Code review, documentation review	Work on project
15	Open Source development issues and current themes	Publish Alpha code (if applicable & meets coding standards)	End of Course