

**4002-XXX**  
**Software Development on Linux Systems**  
**Course Syllabus**

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

Instructor: TBA  
Office: 70 – XXXX  
XXX-XXXX (Office)

Office Hours:  
TBA  
or by appointment

Email address: TBA

Course Web Page: MyCourses.rit.edu

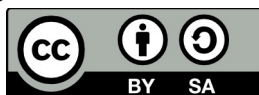
**Course Text and Materials**

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](http://fedoraforum.org)
- [ubuntuforums.org](http://ubuntuforums.org)
- [linuxquestions.org](http://linuxquestions.org)
- [ubuntuguide.org](http://ubuntuguide.org)
- [launchpad.net](http://launchpad.net)
- [bugzilla.redhat.com](http://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, lab 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 December 0, 20xx.

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

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

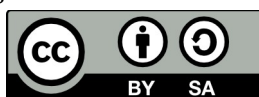
## Prerequisites

4002-219 and 4002-360 . These prerequisites are necessary and will be strictly enforced.

## Role of course in curriculum for

IT: This is an elective in Information Technology

Applied Networking and System Administration: 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 (ideally Launchpad)
- Examine the bug tracking process, maintenance and patching
- Explore future development and project forking

## Course Organization

### Lab practical exams

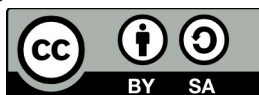
A midterm practicum will be scheduled during lab class time.

### Project

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.

### Labs

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



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

A lab check-off sheet will be provided to keep track of you completed labs. The official record will be on MyCourses and updated by your instructor or TA. You must be in the lab to get lab completion signoffs.

Points assigned for each lab are shown on the lab check-off sheet and MyCourses. Labs 1, 2 and 3 must be completed before the midterm practical. Points for these labs will not be awarded after the midterm. A lab schedule is located below and will normally be on the second class day of each week, including the first week.

## Grading

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

Component	Weight
Lab Exercises	25%
Lab Practical	25%
Homeworks	10%
Project	40%
Peer Evaluation (if applicable)	10%

Range	Grade
$\geq 90.0\%$	A
$\geq 80.0\% \ \& \ < 90\%$	B
$\geq 70.0\% \ \& \ < 80\%$	C
$\geq 60.0\% \ \& \ < 70\%$	D
$< 60.0\%$	F

## Academic Honesty Policy

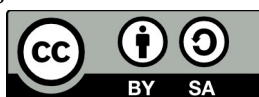
Academic dishonesty is misrepresenting anyone else's work as your own. This includes the past or current work of any other person – student, friend, colleague, relative, faculty member or absolute stranger, either with or without his/her knowledge. Academic dishonesty can be plagiarism, collusion, or outright cheating. All students involved will receive "F" for this course. These are "single instance" policies – i.e. there are no second chances.

A detailed reference to clarify what is and is not plagiarism can be found at <http://www.wpacouncil.org/node/9>. Plagiarism and other academic actions contrary to RIT's code of conduct and the attached

[http://www.rit.edu/studentaffairs/studentconduct/rr\\_academicdishonesty.php](http://www.rit.edu/studentaffairs/studentconduct/rr_academicdishonesty.php) will be addressed in accordance with these policies.

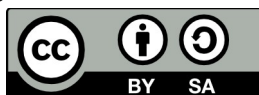
Except for assignments that are specifically designated as being "group efforts," all work submitted (assignments, labs, participation activities, bonus opportunities, examinations, etc.) under your name is assumed to be your own individual effort and will be graded as such under the IST Department's 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	Lab 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	Lab 2: Open Source Community involvement and data gathering	Community involvement lab and project planning
3	Open source relationship with business and proprietary software.  Open source communities, societies and politics	Lab 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	Lab 4: Using development tools and frameworks; exploring IDEs	Work on Project
5	Command line discussion, scripting, intro to configuration files	Lab 5: Scripting for development	Work on Project
6	Version control software and project planning	Lab 6: Configuration files and installing/using version control software	Work on Project
7	Code Documentation, release documentation and generating man pages	Lab 7: Documentation and Code Review	Study for lab practical
8	Practical Review	Lab Practical	Work on Project
9	Linux distribution differences, specialty distributions, building RPMs and DEBs	Lab 8: Building and installing RPMs and DEBs	Work on Project



10	Integrating with services; databases, couchdb, d-bus	Lab 9: Couchdb, mysql, SQLite and d-bus	Work on Project
11	Integrating with services; Apache and Django	Lab 10: Apache, Django and MySQL; 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	Lab 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

