

1. Course Code

2202

2. Course Title

Fundamentals of Open Source Software

3. Teacher

MARKON, Sandor

4. Term

Fall 2

5. Course Overview and Objectives

Open Source Software is becoming an important resource for development, especially in developing countries. A working understanding of the economical and technical background of the Free / Open Source Software movement (FOSS) is essential for its effective use. ICT innovators need an active knowledge of the available FOSS resources, including hosting sites, projects, communities, legal frameworks, organizations etc.

The course takes students through the history and current status of the FOSS world, and starts them exploring it, by connecting their personal experiences with corresponding FOSS projects. Students will experience finding and using Open Source Software projects. They also learn how to initiate a new project, or join and extend an existing project. By completing the course, students will be ready to apply OSS methods and resources in their work.

6. Course Goals (Attainment Targets)

- (1) Can identify the licensing of open source systems and make decisions on their use, based on an understanding of the legal, economical and technical issues.
- (2) Can find open source projects related to a given development problem.
- (3) Can install from source code an open source project and start using it.
- (4) Can choose the correct license, development model, and development community for open source projects, and can initiate a new project or join an existing project.
- (5) Can use the major development platforms and tools that are common for open source projects.
- (6) Experience participating in a public open source project.

7. Correspondence relationship between Educational goals and Course goals

Educational goals of the school			Course Goals
High level ICT skills	Basic academic skills		(1) (5) (6)
	Specialized knowledge and literacy		(2) (3) (4)
Human skill (Tankyu skill)	Ability to continually improve own strengths		(2) (3) (4)
	Ability to discover and resolve the problem in society	Problem setting	(2) (4)
		Hypothesis planning	(2)
		Hypothesis testing	
		Practice	
	Fundamental Competencies for Working Persons	Ability to step forward	(2) (4)
		Ability to think through	(4)
		Ability to work in a team	(6)
Professional ethics			(1) (4)

8. Course Requirements (Courses / Knowledge prerequisite for this course)

9. Textbooks (Required Books for this course)

None. This course uses only material freely available on the Internet (links will be provided on Moodle).

10. Reference Books (optional books for further study)

11. Evaluation

Goals	Evaluation method & point allocation					
	examination	Quiz	Reports	Presentation	Deliverables	Other
(1)			○			
(2)				○	○	
(3)				○	○	
(4)				○	○	
(5)			○			
(6)				○	○	
Allocation			40	30	30	

12. Notes

13. Course plan

(Notice) This plan is tentative and might be changed at the time of delivery

Lesson 1: Orientation, introduction, motivation (discussion and demonstration, 90 min)

Students will be engaged in a discussion about their perception and understanding of Open Source Software issues. We review the areas that are to be covered in the course, and identify common issues of importance. A sample Open Source Software project is reviewed, we install and test it together as a starting reference point for the course.

Lesson 2: History, background, current status	(lecture, discussion, 90 min)
---	-------------------------------

We discuss the emergence of Free Software and Open Source Software movements, using examples and practical issues, some of them from personal experience. We then review how OSS is being used now, discuss areas of success and failure, trends, outlook.

Lesson 3: Where are the projects?	(discussion and demonstration, 90 min)
-----------------------------------	--

We review OSS repositories and development organizations, with emphasis on finding projects and resources relevant to a given development task. Students choose together a topic of common interest and we explore OSS repositories to find resources for it.

Lesson 4: Legal and social issues	(lecture, discussion, 90 min)
-----------------------------------	-------------------------------

Starting with the concepts of copyrights and patents for software, we discuss issues like protecting creative work, building up common foundations of culture, freedom of creation and expression, the dilemma of sharing or exclusivity, motivations for contributing etc. Common licensing models for OSS projects will be reviewed and discussed from the point of view of practical consequences of choices for licensing, both for the developer and the user.

Lesson 5: Business issues	(lecture, discussion, 90 min)
---------------------------	-------------------------------

How to make money out of free software? We discuss business models for companies, developers and users, including OSS-oriented IT companies, manufacturers developing OSS components for their physical products, service providers, independent consultants and others.

Lesson 6: Technical foundations of OSS 1.	(lecture, exercises, 90 min)
---	------------------------------

Some of the important coding conventions and development methodologies of OSS development are explored using examples from Apache and the Linux kernel. We cover the use of version control systems, bug reporting, release control, project organization, developer and user forums.

Lesson 7: Technical foundations of OSS 2.	(lecture, exercises, 90 min)
---	------------------------------

We review some major development languages, libraries and frameworks that are commonly used in OSS projects. We look at the source code of Apache as an example, discussing issues like naming, commenting, structuring.

Lesson 8: Using OSS 1.	(group work, 90 min)
------------------------	----------------------

Students form working groups, each of which selects a development topic to be solved using OSS. Each group creates a Tankyu chart, setting up a hypothesis for solving the issue, and a plan for finding OSS resources.

Lesson 9: Using OSS 2.	(group work and discussion, 90 min)
------------------------	-------------------------------------

Each group presents their plan and findings, to be reviewed by the class. Points to be checked include relevance of the resources to the problem, maturity of the OSS projects, practical usability of the current release, and the feasibility of joining and adapting the projects.

Lesson 10: Everyday OSS	(demonstration and discussion, 90 min)
-------------------------	--

We discuss living with Android, Linux, LibreOffice, Chrome, Thunderbird, and other OSS components, without requiring any non-free software. Who can and who cannot leave the commercial software world? Can a company, a school, a government office, a grandmother standardize on OSS? What are the missing pieces? What are the advantages and pitfalls? What is coming in the near future? How can we contribute?

Lesson 11: Using OSS 3.	(group work, 90 min)
-------------------------	----------------------

Final presentation of the “Using OSS” project.

Lesson 12: Developing OSS 1.	(group work, 90 min)
------------------------------	----------------------

The groups select a development topic (might be the same as the “Using OSS” project) that requires a new OSS project for solution. Each group sets up a new OSS project at SourceForge, GoogleCode or another hub and starts development.

Lesson 13: Developing OSS 2.	(group work, 90 min)
------------------------------	----------------------

The groups present their status, discussing progress and issues that they have encountered.

Lesson 14: Developing OSS 3.

(group work, 90 min)

Final presentation of the “Developing OSS” projects.

Lesson 15: (enter lesson title)

(discussion, 90 min)

Based on what we have learned in the course, we discuss what each student will do using OSS during their studies and after graduating. We also review the impact of OSS on our countries and the society, and discuss what we can do to contribute.
