

**Subject Code: 01CE0618**
**Subject Name: Open Source Technologies**
**B.Tech. Year – III**

**Objective:** This course enables students to analyze and apply open-source technologies through the study of software licensing models, version control systems, package managers, and deployment platforms. Students will also evaluate and use open-source tools for application development, automation, and testing in real-world scenarios. The course emphasizes hands-on exploration and understanding of contributions within open source communities.

**Credits Earned:** 04 Credits

**Course Outcomes:** After completion of this course, student will be able to

- Understand the principles of Open Source Software (OSS) and explore various open-source licenses
- Apply version control tools for effective team collaboration and code management.
- Analyze the architecture and deployment strategies of full-stack open-source applications to determine their suitability for various industry use cases.
- Evaluate software reliability using open-source testing frameworks and tools.
- Apply and Analyse existing open-source software projects and customize open source applications.

**Prerequisite of course:** Basic understanding of operating systems, programming and web technologies.

**Teaching and Examination Scheme**

Teaching Scheme (Hours)			Credits	Theory Marks			Tutorial/ Practical Marks		Total Marks
Theory	Tutorial	Practical		ESE (E)	Mid Sem (M)	Inter nal (I)	Viva (V)	Termwo rk (TW)	
3	0	2	4	50	30	20	25	25	150

**Contents:**

<b>Unit Topics</b>	<b>Contact Hours</b>
--------------------	----------------------

1	<b>Introduction to Open Source:</b> Definition, History, OSS vs Proprietary vs Freeware, Characteristics of OSS, Benefits and Challenges	4
2	<b>Open Source Licensing and Legal Aspects:</b>	4



**Syllabus for Bachelor of Technology  
Computer Engineering**

GPL, MIT, BSD, Apache, LGPL Copyleft vs Permissive, Copyright, Patent, Trademark, Compliance Considerations		
3	<b>Open Source Communities and Contribution Models:</b> Structure of OSS communities, How to contribute, Communication platforms (IRC, forums), Issue tracking, Pull requests, Code of Conduct	4
4	<b>Version Control Systems:</b> Git Basics, Git workflows (feature branches, pull requests), GitHub/GitLab/Bitbucket usage	4
5	<b>Package Managers :</b> Introduction to npm, pip, composer, yarn, Semantic versioning, Managing project dependencies	4
6	<b>Web App Development using OSS Frameworks :</b> Introduction to Laravel, Django, Node.js, MVC architecture overview, Setting up development environments	5
7	<b>Deployment Platforms and Web Servers :</b> Web servers (Apache, Nginx), Reverse proxy and load balancing, Introduction to CI/CD in open source deployments	4
8	<b>Open Source Databases :</b> Relational - MySQL and PostgreSQL - Basic setup, CRUD operations, common use cases, Non-Relational - MongoDB - Document-based model, CRUD operations, basic comparison with SQL, MySQL workbench and MongoDB compass - GUI for interaction	3
9	<b>Open Source Testing and Automation Tools :</b> Unit testing (PyTest, PHPUnit, JUnit), Web testing (Selenium, JMeter), API testing (Postman), Security testing (OWASP ZAP)	5

10	<b>Case Studies and Real-world OSS Applications :</b> Desktop (GIMP, LibreOffice, Thunderbird), Web (WordPress, Wikipedia, Odoo), Using and customizing open source projects	5
	<b>Total Hours</b>	<b>42</b>

#### References:

1 Karl Fogel, **Producing Open Source Software**, O'Reilly Media.

2 Scott Chacon & Ben Straub, **Pro Git**, Apress.

3 Automate the Boring Stuff with Python Al Sweigart , No Starch Press. 4 Selenium Testing Tools Cookbook , Unmesh Gundecha , Packt Publishing

#### Suggested Theory distribution:



#### Syllabus for Bachelor of Technology Computer Engineering

Distribution of Theory for course delivery and evaluation					
Remember	Understand	Apply	Analyse	Evaluate	Create
0%	10%	40%	40%	10%	0%

#### Suggested List of Experiments:

Sr. No.	Topics	Contact Hours
1	Explore github/gitlab for open-source projects with different licenses	2
2	Setup git and explore commands related to VCS	2
3	Create a GitHub/GitLab repository and upload sample code	2
4	Use npm/yarn/pip to install and manage packages	2
5	Deploy a simple app using nginx/apache	2
6	Setup nginx to handle proxy request and load balancing	2
7	Develop a Laravel/Django based web application	2
8	Use pytest to test python application	2
9	Use selenium to create browser based test	2
10	Use Postman to test sample API	2

11	Use OWASP ZAP to check security integrity	2
12	Modify any of the open source desktop application	2
13	Contribute to any of the web based open source projects	2
14	Use source code of open source web application and deploy on your machine	2

**Instructional Method:**

- a) The course delivery method will depend upon the requirement of content and the needs of students. The teacher, in addition to conventional teaching methods by black board, may also use any tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.
- b) The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.
- c) Practical examination will be conducted at the end of semester for evaluation of performance of students in the laboratory.

**Supplementary Resources and Documents**

1. <https://www.edx.org/certificates/professional-certificate/linuxfoundationx-open-source-software-development-linux-and-git>
2. <https://www.coursera.org/specializations/oss-development-linux-git>
3. <https://docs.github.com/>
4. <https://www.selenium.dev/documentation/>