Course: BTech **Semester:** 1

Prerequisite:

Rationale: Open Source has acquired a prominent place in software industry. Having knowledge of Open Source and its related technologies is an essential for Computer Science student. This course introduces Open Source methodologies and ecosystem to students.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	T	CE	P	T	P	
2	0	2	-	3.00	20	20	20	60	30	150

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**Introduction to Open-Source:** Open Source, Need and Principles of OSS, Open-Source Standards, Requirements for Software, OSS success, Free Software, Examples, Licensing, Free Vs. Proprietary Software, Free Software Vs. Open-Source Software, Public Domain. History of free software, Proprietary Vs Open-Source Licensing Model, use of Open- Source Software, FOSS does not mean no cost. History: BSD, The Free Software Foundation and the GNU Project.	15	3
2	**Open-Source Principles and Methodology:** Open-Source History, OpenSource Initiatives, Open Standards Principles, Methodologies, Philosophy, Software freedom, Open-Source Software Development, Licenses, Copyright vs. Copy left, Patents, Zero marginal cost, Income-generation Opportunities, Internationalization. **Licensing**: What Is A License, How to create your own Licenses, Important FOSS Licenses (Apache, BSD, PL, LGPL), copyrights and copy lefts, Patent	20	5
3	**Open-Source projects:** Starting and maintaining own Open- Source Project, Open-Source Hardware, Open-Source Design, Open-source Teaching, Opensource media. **Collaboration:** Community and Communication, Contributing to Opensource Projects Introduction to GitHub, interacting with the community on GitHub, Communication and etiquette, testing open-source code, reporting issues, contributing code. Introduction to Wikipedia, contributing to Wikipedia or contributing to any prominent open-source project of student's choice. Open-Source Ethics and Social Impact: Open source vs. closed source, Open-source Government, Ethics of Open- source, Social and Financial impacts of open-source technology, Shared software, Shared source, Open Source as a Business Strategy	20	7
4	**Understanding Open-Source Ecosystem:** Open-Source Operating Systems: GNU/Linux, Android, Free BSD, Open Solaris. Open-Source Hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, Debuggers, Programming languages, LAMP, Open-Source Database technologies	20	7
5	**Case Studies: Example Projects** Apache Web server, BSD, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, WordPress, Git, GCC, GDB, GitHub, Open Office, LibreOffice Study: Understanding the developmental models, licensing, mode of funding, commercial/non-commercial use.	25	8

Reference Books	
1.	**Open-Source Technology", Kailash Vadera & Bhavyesh Gandhi, University Science Press, Laxmi Publications, 2009, Software Engineering, Sommerville (TextBook)**
2.	**Open-Source Technology and Policy", Fadi P. Deek and James A. M. McHugh, Cambridge University Press, 2008Software Engineering Pankaj Jalote; Wiley India**
3.	**Perspectives on Free and Open-Source Software", Clay Shirky and Michael Cusumano, MIT press**

Course Outcome

After Learning the Course the students shall be able to:
After learning the course, the students shall be able to: 1. Differentiate between Open Source and Proprietary software and Licensing. 2. Recognize the applications, benefits, and features of Open-Source Technologies 3. Gain knowledge to start, manage open-source projects. 4. Worked with Open-Source ecosystem, its use, impact, and importance. 5. learn Open-Source methodologies, case studies with real life examples.

List of Practical

1.	Demonstration of Basic Linux commands.
2.	Execute C Program using *gcc* compiler.
3.	Demonstration of *gprof* command using Linux.
4.	Create and Edit documents using Google Docs.
5.	Create Presentation using Google Slides.
6.	Demonstration of different Arithmetic and Logical Formulas using OpenOffice Calc.
7.	Use of HTML to create simple web page.
8.	Demonstration of MathML – a markup language for describing mathematical notation.
9.	Demonstration of virtualization using Docker Container.
10.	Demonstration GitHub Facility.