UNIT 1 – Basics, Principles & Licensing

---

1. What is Free Software?

- Free Software means users have the *freedom to run, study, modify, and share* programs.

- Introduced by **Richard Stallman** in the 1980s.

- Managed by the **Free Software Foundation (FSF)**.

- Promotes collaboration and transparency.

- Not necessarily zero cost — "Free" means **freedom**, not "price."

- Users can customize programs for their own needs.

- Example: **GNU Compiler Collection (GCC)**, **GIMP**, **Linux Kernel**.

---

2. What is Open-Source Software?

- Software whose **source code is available** to everyone.

- Encourages community collaboration and innovation.

- Anyone can fix bugs, add features, or improve security.

- Maintained by a global developer community.

- Focused on **development efficiency** and **transparency**.

- Example: **Mozilla Firefox**, **LibreOffice**, **Python**, **MySQL**.

---

3. Difference between Free and Open-Source Software

Aspect	Free Software	Open-Source Software
Philosophy	Emphasizes user freedom.	Focuses on collaborative development.
Control	Users control software usage.	Community controls development process.
Focus	Ethical and social values.	Practical and technical benefits.
Promoted by	Free Software Foundation (FSF).	Open Source Initiative (OSI).

Aspect	Free Software	Open-Source Software
Licensing	Uses GPL, which enforces freedom.	Uses varied licenses (MIT, Apache).
Example	GNU/Linux, Emacs.	Firefox, VLC, Blender.

4. Difference between Proprietary and Open-Source Software

Aspect	Proprietary Software	Open-Source Software
Access to Source Code	Hidden; not shared.	Publicly available.
Cost	Usually paid license.	Often free or low cost.
Customization	Not allowed.	Freely modifiable.
Support	Provided by vendor.	Provided by community or paid support.
Examples	Microsoft Windows, Adobe Photoshop.	Linux, LibreOffice, GIMP.

5. Explain Software Freedom

- Defined by the **4 essential freedoms**:
    1. Run the program for any purpose.
    2. Study how the program works.
    3. Modify and improve the program.
    4. Share copies and distribute improvements.
- Ensures **transparency and innovation**.
- Protects user rights against restrictive licensing.
- Example: **GNU/Linux** upholds all freedoms.

6. What are Open-Source Standards?

- Publicly available technical guidelines ensuring interoperability.

- Prevent **vendor lock-in** by promoting openness.

- Allow multiple systems to communicate smoothly.

- Encourage fair competition and innovation.

- Examples: **HTML, TCP/IP, XML, ODF (Open Document Format).**

---

7. Key Requirements of Open-Source Software

- Source code must be open.

- Free redistribution allowed.

- Modifications and derivative works permitted.

- No discrimination against users or fields.

- License must be neutral to technology.

- Example: **Apache License**, **MIT License**.

---

8. Principles of Open Source

- **Transparency:** Open visibility of source code.

- **Collaboration:** Anyone can contribute.

- **Community-driven:** Maintained by global users.

- **Peer review:** Quality improves continuously.

- **Freedom of choice:** No restrictions on use.

- Example: **Linux Kernel development model**.

---

9. Shared Software

- Software made public for free or educational use.

- May have restrictions on editing or distribution.

- Allows limited collaboration, not fully open-source.

- Often used in research or academic settings.

- Example: **MATLAB Student Version**, **Trial Software**.

---

10. FOSS as a Business Strategy

- Businesses earn via support and services.

- Promotes brand reputation and community trust.

- Encourages adoption by lowering entry barriers.

- Companies offer free software but paid upgrades.

- Examples: **Red Hat Enterprise Linux**, **WordPress hosting**.

---

11. History of Free Software

- 1983 – Richard Stallman starts the **GNU Project**.

- 1985 – **Free Software Foundation (FSF)** established.

- Created free Unix-like utilities and compilers.

- Released under the **GNU General Public License (GPL)**.

- Combined with Linux kernel → **GNU/Linux**.

---

12. Role of FSF and GNU Project

- FSF promotes freedom and ethical software use.

- GNU Project developed free OS components.

- Produced compilers (GCC), editors (Emacs).

- Encouraged the **Copyleft principle**.

- Example: **GNU/Linux system**.

---

13. Pros and Cons of FOSS

Advantages	Disadvantages
Free to use and modify.	May lack official support.
Secure and transparent.	Interface not always user-friendly.
Encourages innovation.	Compatibility issues with proprietary systems.
Reduces cost for users.	Some software lacks polish or updates.

Advantages   **Disadvantages**

Example: **LibreOffice vs MS Office**

---

14. Open-Source Hardware

- Hardware designs released publicly.

- Anyone can study, modify, and manufacture.

- Encourages learning and local production.

- Promotes low-cost innovation.

- Examples: **Arduino**, **Raspberry Pi**, **Open Compute Project**.

---

15. Copyright and Copyleft

Aspect	Copyright	Copyleft
Definition	Legal right restricting copying.	Uses copyright to keep software free.
Control	Maintains exclusive ownership.	Allows modification with same rights.
Purpose	Protects creator's control.	Ensures user freedom continues.
Example	Microsoft Word.	GNU GPL software.

---

16. Free vs Proprietary Licensing

Free Licensing	Proprietary Licensing
Source code open.	Source code closed.
Free modification and distribution.	Restricted redistribution.
User retains rights.	Vendor retains all rights.
Example: GPL, MIT, Apache.	Example: Adobe Photoshop, MS Office.

---

17. Importance of Licensing in FOSS

- Protects developer and user rights.

- Defines how code can be used or shared.

- Prevents misuse and enforces openness.

- Supports global collaboration.

- Examples: **GPL**, **MIT**, **BSD**, **Apache License**.

---

18. Income Generation in FOSS

- Paid technical support and training.

- Customization for organizations.

- Cloud hosting services.

- Donations and sponsorships.

- Example: **Mozilla** gets funds from Google; **Red Hat** from support services.

---

19. Benefits of Open Standards

- Encourages cross-platform compatibility.

- Ensures free access to specifications.

- Reduces dependency on one vendor.

- Promotes global collaboration.

- Example: **ODF**, **PDF**, **HTML5**.

---

20. Role of Wikipedia in Open Source

- Free online encyclopedia built collaboratively.

- Open editing and content sharing under **Creative Commons License**.

- Uses **MediaWiki**, an open-source platform.

- Symbol of open knowledge sharing.

- Nonprofit, transparent, and community-driven.

---

21. WordPress and Start-ups

- Free, open-source website CMS.

- Simple interface; no coding required.

- Thousands of free plugins and themes.

- Used by small businesses and bloggers.

- Example: **WooCommerce** for e-commerce.

---

22. Example of Open-Source Software

- **Linux:** Free operating system kernel.

- **LibreOffice:** Office suite alternative to MS Office.

- **VLC:** Free media player for all platforms.

- **Python:** Programming language for AI and data science.

UNIT 2 – Collaboration, Ecosystem & Ethics

---

23. Importance of Community

- Community ensures sustainability of projects.

- Volunteers contribute to updates and security.

- Encourages peer support and knowledge sharing.

- Creates trust among developers.

- Example: **Linux community**, **Apache Foundation**.

---

24. How to Contribute to Open Source

- Identify project and read contribution guidelines.

- Fork the repository (copy project).

- Make code or documentation changes.

- Submit a **pull request** for review.

- Participate in issue discussions.

- Example: Contributing on **GitHub** to Mozilla or Django.

---

25. What is GitHub

- Web platform for code hosting and collaboration.
- Uses **Git** version control system.
- Developers can store, track, and share code.
- Supports **issues**, **branches**, and **pull requests**.
- Example: Projects like TensorFlow and React are hosted on GitHub.

---

26. Steps to Contribute using GitHub

1. Create a GitHub account.
2. Fork the repository (copy project).
3. Clone repo to your computer.
4. Make changes and commit them.
5. Push changes and submit pull request.
6. Wait for project maintainer to review.

---

27. Open-Source Government

- Governments using open software and data.
- Improves transparency and cost efficiency.
- Enables public participation in tech policy.
- Example: **BOSS Linux** (India), **GOV.UK** platform (UK).

---

28. Role of Open Source in Government

- Reduces cost by avoiding expensive licenses.
- Enhances local skill development.
- Enables customization to regional needs.
- Promotes transparency and data accessibility.
- Example: **Kerala's IT@School Linux initiative**.

---

29. Social Impact of Open Source

- Empowers education and innovation.

- Promotes digital inclusion worldwide.

- Builds local software communities.

- Reduces dependency on foreign vendors.

- Example: **Linux in public schools** and universities.

---

30. Financial Impact of Open Source

- Reduces operational software costs.

- Promotes open digital economy.

- Encourages local entrepreneurship.

- Allows SMEs to use advanced tech for free.

- Example: Banks and startups using Linux instead of paid OS.

---

31. Ethical Issues in Open Source

- Giving proper credit to contributors.

- Avoiding plagiarism and code theft.

- Respecting software license conditions.

- Avoiding fake commits or malicious code.

- Promoting inclusivity in communities.

---

32. Open-Source Operating Systems

- Provide free and customizable OS environment.

- Highly secure with regular updates.

- Supported by global developer networks.

- Examples: **Linux**, **Android**, **Ubuntu**, **FreeBSD**.

---

33. Role of Open Source in Office Tools

- Free alternatives to commercial suites.

- Include word processor, spreadsheet, presentation tools.

- Compatible with MS Office formats.

- Example: **LibreOffice**, **Apache OpenOffice**.

- Used by schools, NGOs, and government agencies.

---

34. Case Study – Any Open-Source Project

Example – Mozilla Firefox:

- Free web browser developed by Mozilla Foundation.

- Built under **Mozilla Public License (MPL)**.

- Funded by donations and partnerships.

- Community-driven updates and privacy-focused.

- Symbol of open innovation and transparency.