**Recent Open Source Project Lab**

**Assignment No. 1**

# **Q1. Basic Concepts of Open Source Software.**

Open source software is software with source code that anyone can inspect, modify, and enhance. Source code is the part of software that most computer users don't ever see, it's the code computer programmers can manipulate to change how a piece of software, a program or application works. Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't always work correctly.

People prefer open source software to proprietary software for a number of reasons, including:

1. **Control**: Many people prefer open source software because they have more control over that kind of software. They can examine the code to make sure it's not doing anything they don't want it to do, and they can change parts of it they don't like.
2. **Training**: Other people like open source software because it helps them become better programmers. Because open source code is publicly accessible, students can easily study it as they learn to make better software. Students can also share their work with others, inviting comment and critique, as they develop their skills.
3. **Security**: Some people prefer open source software because they consider it more secure and stable than proprietary software. Because anyone can view and modify open source software, someone might spot and correct errors or omissions that a program's original authors might have missed.
4. **Stability**: Many users prefer open source software to proprietary software for important, long-term projects. Because programmers publicly distribute the source code for open source software, users relying on that software for critical tasks can be sure their tools won't disappear or fall into disrepair if their original creators stop working on them.
5. **Community**: Open source software often inspires a community of users and developers to form around it. That's not unique to open source; many popular applications are the subject of meetups and user groups.

# **Q2. Difference between Free and Open source software.**

| **Free Source Software** | **Open Source Software** |
| --- | --- |
| It was coined by the Free Software Foundation. | In response to the restrictions of free software, the phrase open source. |
| Free Software does not allow you to work with other proprietary software. | Open Source software can work with other proprietary software. |
| Free Software allows the coder to have least control over the program. | Open Source allows the coder more control over his program compared to Free Software. |
| Every free software is open source. | Every open-source software is not free software. |
| There is no such issue that exists in free software. | There are many different open-source software licenses, and some of them are quite restricted, resulting in open-source software that is not free. |
| No restrictions are imposed on free software. | Open-source software occasionally imposes some constraints on users. |
| Examples: The Free Software Directory maintains a large database of free software packages. Some of the best-known examples include the Linux kernel, the BSD and Linux operating systems. | Examples: Prime examples of open-source products are the Apache HTTP Server, the e-commerce platform Open Source Commerce, internet browsers Mozilla Firefox, and Chromium. |

# 

# **Q3. What is GPL (General Public License)?**

The GNU General Public License is a free, copyleft license for software and other kinds of works. The licenses for most software and other practical works are designed to take away the freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee the freedom to share and change all versions of a program to make sure it remains free software for all its users.

When we speak of free software, we are referring to freedom, not price. The General Public Licenses are designed to make sure that we have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if we want it, that we can change the software or use pieces of it in new free programs, and that we know we can do these things.

Developers that use the GNU GPL protect your rights with two steps:

1. Assert copyright on the software
2. Offer the License giving you legal permission to copy, distribute and/or modify it

To protect your rights, we need to prevent others from denying you these rights or asking to surrender the rights. Therefore, there are certain responsibilities if we distribute copies of the software, or if we modify it responsibilities to respect the freedom of others. For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

# **Q4. Different ways to contribute to an Open Source Project?**

Let's see the most common ways to contribute to Open Source projects:

**Create your own open source project**: Every project should start with an identified need. If you feel that existing projects on GitHub don't offer the functionality you would like to build, then create your own open source solution. Besides an initial project draft, you should consider the following set of questions:

1. What skills do you need for your project?
2. How much time are you willing to spend on your project?
3. What problem(s) does your software solve?
4. How many potential users are there for your product?

**Create open source alternatives to commercial software**: Today's commercial projects actively engage open source solutions. Many companies base their projects on free tools. When there's a huge selection of software, you don't need to reinvent the wheel. This is why it's useful to play around with free software that can replace similar proprietary software, or that fixes an issue you've recently faced.

Another reason for replacing commercial solutions with open source software is eagerness for real innovation and growth. Commercial software claims to be innovative, but its final goal is turning a profit. Open source software unites best practices, great quality of code and passionate developers willing to code just because they like to.

**Contribute to existing open source projects**: You can find many projects you are free to participate in on GitHub, a developer-oriented platform with a simple but essential set of functionality. GitHub attracts developers with public APIs, a sleek and frequently updated UI, gists (Git repositories) that allow you to share pieces of code or even whole applications, and much more. You can contribute to free software in many ways. Developers can fork projects, make changes to code, and send pull requests. And quality assurance is always appreciated. Sometimes developers are too busy or too lazy to check the quality of their code. So go ahead and report a bug or try to fix it.

You can reach the hottest GitHub projects by following the Trending link. And in order to make your search more relevant, use advanced search, select the language you would like to code in and choose the best match criteria. Best match ranks projects according to relevance, taking into account the number of forks (which represents how actively the project is updated) and stars. Most projects have known issues with labels like bug, discussion, security, or refactor, or other labels according to the level of difficulty like easy, medium, hard.