**Recent Open Source Project Lab**

**Assignment No. 2**

# **Q1. Can Open Source software be used for commercial purposes?**

All Open Source software can be used for commercial purposes, the Open Source Definition guarantees this. You can even sell Open Source software. However, note that commercial is not the same as proprietary. If you receive software under an Open Source license, you can always use that software for commercial purposes, but that doesn't always mean you can place further restrictions on people who receive the software from you. In particular, copyleft-style Open Source licenses require that, in at least some cases, when you distribute the software, you must do so under the same license you received it under.

# **Q2. What is copy left? Is it the same as open source?**

Copyleft is a subset of open source. Contrary to what the term might imply, copyleft is not the opposite of copyright. In fact, copyleft is grounded in the concept of copyright, without which copyleft couldn't exist. Before someone can license software under a copyleft license, they must first own the copyright to that software, thus giving them the right to distribute it.

Both open source and copyleft allow for source code to be modified and distributed. However, the difference is that with copyleft, the modified product must be distributed with the same copyleft license attached to the original software. This allows for creation of derivative work based on the source code covered by the copyleft license while protecting the original creator's interests.

For example, Bob owns the copyright to original software, which he decides to distribute under a copyleft license. Gemma downloads Bob's software, modifies it, and then distributes her modified version. Gemma must distribute her modified version under the same copyleft license that Bob used. Anyone who then modifies Gemma's version must also distribute it using Bob's same copyleft license.

Both open source and copyleft are complex terms that deal with a user's right to access, use, modify, and distribute a work that is based on the source code of another piece of software. Understanding the basics of both concepts is crucial when modifying or distributing any form of software or computer code.

# 

# **Q3. Which Open Source license should be chosen to release software under?**

Open source software is often distributed widely and used in other software programs. So it’s important to tell others how to use it. Using a specific license also gives your software legal protection. You can use it to take legal action against those who violate the terms of the license.

There are two basic types of open source licenses:

1. **Permissive license**: These live up to their name by allowing users to do just about anything they want with the code. Users don’t have to republish any changes they make and typically only have to give attribution to the original authors in the form of a comment within the code. Permissive licenses include the MIT license, Apache License 2.0, and the BSD License.
2. **Copyleft license**: These require that anyone who changes your code for their software also has to release their code. For example, if you use an open source library to build your software, you’ll need to open up your code for others to operate under the same license. Copyleft licenses include the GPL license, the AGPL License, the LGPL License, and the Mozilla Public License 2.0.

The first question is if your code will be part of a broader open source ecosystem many open source communities have a few favorite licenses they use. If it will be, consider using the licenses vetted and accepted by your community. If you’re on your own, first ask yourself how you want others to use your software. Are you building an open source library others will use to create their software?

If you choose a permissive license, users can take your code, change it, and use it in their proprietary software without releasing the code. If you choose a copyleft license, anyone who uses your code will have to share their code in the same way.

Many open source licenses dictate how or if the software can be used in commercial applications. You can choose if companies can use your libraries in commercial applications and the requirements and restrictions under which they can do so.

# **Q4. Which Open Source license is best?**

Unlike bilateral copyright licenses, which are negotiated between two parties and embody a truce between them for business purposes, multilateral copyright licenses of which open source licenses are a kind are community agreements. They express the consensus of how a community chooses to collaborate. They also embody its ethical assumptions, even if they are not explicitly enumerated.

When that consensus includes giving permission to all to use, study improve and share the code without prejudice, the license is an open source license. The Open Source Definition provides an objective test of evaluating that such a license is indeed an open source license and delivers the software freedom we all expect. The following is a list of the most popular open source licenses used by developers, their risk classifications, and whether the license has been approved by the Open Source Initiative (OSI).

| **Rank** | **License** | **Risk** | **OSI Approved** |
| --- | --- | --- | --- |
| 1 | MIT License | Low | Yes |
| 2 | Apache License 2.0 | Low | Yes |
| 3 | ISC License | Low | Yes |
| 4 | BSD 3 - Clause New or Revised License | Low | Yes |
| 5 | BSD 2 - Clause Simplified License | Low | Yes |
| 6 | Creative Commons Zero v1.0 Universal | Varies by usage | No |
| 7 | Generic Public Domain | Varies by usage | N/A |
| 8 | GNU Lesser General Public License v2.1 | High | Yes |
| 9 | Common Development and Distribution License 1.1 | Medium | Only 1.0 |
| 10 | Eclipse Public License 1.0 | Medium | Superseded by version 2.0 |

# **Q5. What are contributor agreements? Are they like open source licenses?**

Many open source projects will only accept patches (code contributions or documentation contributions) from people who have submitted a legal document known as a contributor agreement. Contributor agreements are not open source licenses but rather, they are a way for the contributor to tell the project that it has the right to distribute the new contributions under the project's existing open source license. Some contributor agreements also allow for the project to distribute the contributions under other open source licenses too, which enables projects to change their license in the future, and some agreements even allow the project to distribute the contributions under any license the project wants.

There are two kinds of contributor agreements:

1. **Contributor License Agreement (CLA)**: The original contributor retains copyright ownership of their contributions, but grants the project a broad set of rights such that the project can incorporate and distribute the contributions as it needs to.
2. **Copyright Assignment Agreement (CAA)**: Here the contributor actually transfers copyright ownership of the contributions to the project, who can then license it however they want since they own it but a CAA typically grants very broad non-exclusive rights back to the contributor so that they too can use, distribute, sublicense etc their contribution freely.

With both CLAs and CAAs, it is of course necessary that the project be some kind of legal entity able to enter into agreements. Sometimes the project is incorporated itself, usually as a non-profit entity or sometimes it is represented by an umbrella non-profit organization or sometimes a for-profit corporation considers itself the main sponsor of the project and requests contributor agreements in order to manage the development community and maintain a public distribution of the software in question.