

**Software functions:**

* Patent (om det är en teknisk lösning)
* Copyright (protect source and code)
* Trademark?

**Software design, structure:**

* Trademark
* Designskydd
* Copyright (inte i Europa, baserat på vissa kriterier)

**Software design, user interface:**

* Design (mönsterskydd)

**Software code:**

* Secrecy

**What is the main idea behind the patenting system, i.e. what is it intended to protect?**

**(•**Provides health and safety for creators

• Drives economic growth and competitiveness and encourages innovation.**)**

This is a form of intellectual property (IP) protection that gives the creator of an invention the exclusive legal right to market, sell, manufacture, and profit from that invention.

**Do you think the patenting system works as intended for software patents? Why? Why not?**

As per Sec. 3 (k) of the Patent Act, 2002, computer program cannot be patented per se. However, **a software can be patented if it is attached to an invention** and that it is a component of such invention.

**Task 3: Open Source**

**1. Which IPR aspect is Open Source Software policy aiming to protect and why?**

Open source software by its **nature includes protectable intellectual property rights**, most notably copyright. Therefore, if a person wishes to use any open source software, that person will be obliged to comply with the associated licence terms. So no one can benefit from the innovation commercially.

**2. Gatto (2019) argues that OSS can ”taint” proprietary software. Identify and discuss at least two types of OSS licenses that can have a tainting effect on proprietary software.**

* A GPL-license might be considered as “tainting” proprietary software since you are “forced” by the license to share your code. Say that you use oss with a GPL license.
* LGPL, although being more permissive than strict GPL, can still “taint” proprietary software in similar ways to GPL. You can escape this in some ways, if you write code that only uses the oss binaries and they are compiled separately for example.

*“Perhaps the biggest risk in using OSS is that it may impact proprietary software, including the potential requirement to make the source code for that software available to others. This is often referred to as OSS “tainting” of proprietary software. Some OSS licenses (e.g., the GPL license) require that if any software contains or is derived from any GPL-licensed code, then that software must be licensed under the terms of the GPL license. Two significant ramifications of this are that: i) the source code for that software must be made available to recipients of the software; and ii) recipients must have the right to copy, modify and redistribute that software at no charge. This can be devastating if that software is intended to be proprietary software.”[[1]](#footnote-0)*

* **Funny :-)**

“The LGPL does not require you to distribute the source code of your proprietary software that is linked dynamically with the LGPL code. On the other hand, if you static.ally llnk any proprietary code with the LGPL code, this does trigger a certain copyleft requirement. Under the LGPL, you must allow, with respect to the proprietary code, "modification for the customer's own use and reverse engineering for debugging such modifications. The Free Software Foundation's stated position Is that this requirement obligates you to allow customers to reverse engineer and modify your proprietary software for limited purposes.”[[2]](#footnote-1)

* **Fönny**

**3. Distributing OSS under has gained popularity among the large software producers. What is the intent of releasing OSS under dual licenses? Identify two popular software products released under the dual license**

Dual licensing provides a third option for distributing software. Using dual licensing, licensors can distribute software to licensees under a **proprietary(ownership)** model as well as an open source model, allowing the licensor to simultaneously leverage the advantages of both types of licenses. That is, some companies use a dual licensing model to distribute the same software under two different license forms: (1) a version subject to a proprietary license (which may come with the right to further develop and commercially distribute that software and with licensor technical support and added features), and (2) a version licensed under, and subject to, the restrictions and obligations of an open source license, such as the GPL.

One well known example of dual licensing is Oracle’s MySQL database management system, another is The Qt Project.

* Sell one part, keep another for further development!

1. <https://www.natlawreview.com/article/open-source-software-policies-why-you-need-them-and-what-they-should-include> [↑](#footnote-ref-0)
2. <https://www.legal.io/articles/5170736/Open-Source-Software-a-legal-guide> [↑](#footnote-ref-1)