7907ICT

Workshop Document

**This document is where you write-up the TEN weekly workshop tasks, each one of which is worth 10% of your total marks.**

# How to Approach these workshops (1 - 10)

This document lays out the ten workshop exercises to be completed each week. Either during the workshop session or at another time at your convenience. It contains detail of the task, plus a read-made template to be used when answering the questions.

This is the document that will be submitted for marking in two stages; Part A in week 6 to include workshops 1 through 5. Part B submitted in Week 11 to include weeks 6 to 10.

Key points to note:

* The output of each workshop is a **600-word written report**.
* Write your 600-word report into this workbook, accumulating them until you have completed all ten, then submit it via the Turnitin portal at the bottom of the assignment page of the course website.
* Don’t be tempted to leave doing the workshop write-up until the week the submission. It is a fact that we usually under-estimate the amount of work needed.
* As per university policy, extensions to the allowed time to submit can be granted with the necessary documentation. But please bear in mind that the IT industry is a very deadline driven profession.
* The workshops follow a similar format. Once you become familiar with the process, you should be able to work through the ten workshops over the duration of the course.
* The workshops can be completed individually or in discussions with groups of 2-4 students. Your submission will be an individual one, not a group submission.
* Ensure your report has clear headings for each.
* Try to do one workshop write-up per week.
* Avoid directly copying and pasting information from online sources, including generative language models like ChatGPT or other.

# Module 6: The Ethics of Open-Source Software Licensing

**<Workshop 6>**

### Introduction

Open-Source Software is one of the greatest inventions in the software industry, accelerating the improvement of software and benefiting everyone. You probably have heard of the Linux system, a prime example of how open-source software can create powerful, reliable software that powers everything from servers to smartphones. By allowing developers to freely access, modify, and distribute the code, open-source fosters innovation and gives users more control over the software they use. However, with this also comes the responsibility to ensure that the software is used ethically, raising important questions about how to balance freedom with accountability.

#### <ethical considerations and risks>

Though this data analysis tool can be used for beneficial purposes like research and business optimization, it also has the potential for abuse in activities like targeted advertising or invasive surveillance, which could violate on individuals' privacy. As the developer of this software, I have responsibilities to prevent this tool from being used in ways that could harm society or individuals, such as enabling unethical data collection or profiling. Additionally, releasing the tool under an open-source license may allow malicious users to exploit it, raising the risk of unintended consequences and highlighting the need for careful consideration of how to control its use without sacrificing its positive potential.

#### <Choose a suitable open-source license >

When choosing an appropriate open-source license for this scenario, it is essential to balance the need for openness with the responsibility to prevent malicious use. A permissive license, such as the MIT or Apache License, offers flexibility and broad freedom to use, modify, and distribute the software. However, it does not address potential ethical concerns, such as the misuse of the tool for harmful purposes like surveillance or privacy violations. Alternatively, a copyleft license, such as the GNU General Public License (GPL), ensures that any derivative works remain open-sourced, which promotes transparency, but it also fails to directly tackle ethical issues. A more suitable option might be the Hippocratic License, which is a young licence created by Coraline Ada Ehmke in 2019. This licence explicitly prohibits the use of the software in ways that violate human rights. Although not officially recognized by the Open-Source Initiative (OSI), it aligns with the ethical considerations of the developer and could serve as a safeguard against misuse, while still encouraging collaboration and innovation

#### <solution for mitigating risks>

To mitigate the ethical risks associated with this open-sourced release data analysis tool, several proactive steps can be taken. First, the developer can include a clear ethical guideline in the documentation, specifying what is acceptable and what is unacceptable uses of the tool, and encouraging users to report misuse. Additionally, implementing a user agreement or terms of service that explicitly prohibits harmful applications, such as surveillance or unethical data collection, can provide an advanced protection. Engaging with the user community and promoting a culture of responsible use is another effective approach, as a strong community can help monitor and address potential misuse. Lastly, integrating features within the tool that limit its capacity for unethical use, such as restrictions on data access or enhanced privacy protections, can help minimize the risk of harm while still allowing the tool’s beneficial applications to flourish.

### Conclusion

Open-source software plays a vital role in modern software development by promoting collaboration, innovation, and accessibility. However, it is equally important to consider the ethical issues of releasing such software, as the lack of restrictions can lead to potential misuse. Without appropriate guidelines or controls, powerful tools can be repurposed for unethical activities, such as violating individuals' privacy, facilitating mass surveillance, or exploiting vulnerable groups. As a result, it becomes necessary to implement ethical safeguards into open-source projects to ensure responsible usage. By doing this, developers can help ensure that their software is used in a positive way while minimizing the risk of harm.

### References

<Use APA referencing style>