School of Information Technology, Deakin University

SIT320 — Advanced Algorithms

Pass Task 2: **Design Principles**

About this Task and its related Module

At the completion of the module (**Module 2: Design Principles**), you are required to fill a lesson review by doing following activities.

Your tutor will then review your submission and will give you feedback. If your submission is incomplete they will ask you to include missing parts. They can also ask follow-up questions, either to clarify something, or to double check your understanding of certain concepts.

Task List

- **(0)** Provide a short overview of what you learned in the module. This should be based on your learning summary from lecture (seminar), module content on cloud Deakin, your interaction with Unit Chair/Tutors/Peers, your research in the library or the internet and/or your interaction with chatGPT (make sure to provide the prompts you use).
- **(1)** In module 1, we wrote an algorithm for tic-tac-toe. Though, your focus was on solving the actual problem, let us now focus on building an efficient design. For this activity you are expected to re-factor the code that you wrote for tic-tac-toe to demonstrate your understanding of DIP and OCP principles that we have covered in this module. If you only wrote the algorithm in module 1, and not the code -- you can download code from here: tic-tac-toe Code , and re-factor it. You can make any reasonable assumptions about the extension of the code, e.g., a) user can specify a board size (not just 3x3, but 5x5 or 9x9 as well), b) user can specify various algorithms to play against, e.g., minimax, reinforcement learning, etc. Note, for the later you do not have to implement various algorithms -- you can use a stub which suggests that you will fill the code in later modules. You are expected to develop a simple framework, where multiple algorithms can easily be implemented for various complexities. Your submission must also include a class diagram using correct UML syntax -- you can integrate that in ipynb by including it as a picture.
- **(2)** Check activity 1 to make sure that your use of inheritance is safe. Describe why you preferred to use it an not containment.