

Professional Skills Matrix

Skill	Proficiency Level	Experience/Examples	Areas for Improvement	Development Plan
Time Management	Intermediate	Managed coursework deadlines effectively, combining two modules.	Occasionally underestimated time required for complex tasks, such as translating UML diagrams into Python programmes.	Use time blocking techniques, set buffer periods for tasks and reflect weekly on time allocation.
Critical Thinking and Analysis	Advanced	Evaluated programming challenges, such as aligning UML class diagrams with Python code and analysing metamodeling in IoT design.	Overanalysed scenarios in tasks like state machine diagram creation, which slowed progress in time sensitive activities.	Practise decision making under constraints with timed design challenges and case studies.
Problem-Solving	Intermediate	Resolved debugging issues using pylint and applied breadth first search to navigate a 2D map for a robot scenario.	Expand understanding of advanced algorithms like A* to optimise solutions for larger or more complex data sets.	Study algorithm design and apply problem solving techniques on coding.
Communication and Literacy Skills	Advanced	Articulated ideas in forum discussions, like analysing reusability factors and wrote clear reflections for each unit.	Improve the ability to adapt communication style to different audiences and contexts.	Practice tailoring communication based on audience needs by participating in group discussions and presenting in various formats.
IT and Digital	Advanced	Proficient in Python programming and tools like unittest, pytest, and pylint. Implemented ANSI colour codes and command line interfaces.	Enhance knowledge of debugging tools and explore additional testing frameworks for large scale software projects.	Enhance debugging skills by regularly practising with more complex codebases, using advanced debugging tools.
Numeracy	Intermediate	Used recursion and applied breadth first search for robotic navigation tasks. Implemented data structures like stacks and queues.	Improve ability to adapt mathematical concepts to advanced algorithmic solutions and optimise performance.	Solve numeracy focused coding challenges like Towers of Hanoi.
Critical Reflection	Intermediate	Reflected on strengths and weaknesses in applying OOP principles and integrating design patterns.	Develop a more structured approach to reflection to ensure insights are consistently applied to future tasks.	Maintain a consistent reflection practice throughout the module by setting aside time at the end of each week to document key learnings, challenges faced and areas for improvement.