Object Oriented Programming Reflection

What?

This reflection is about what I learnt in the Object Oriented Programming module. Even though I've been working with OOP concepts in my job as a software engineer, this module was a great chance to review them and learn even more. It was especially interesting to see how OOP implementation differs in Python compared to Java, which is the language I use most often at work. You can find out more about my academic and professional journey in my e-portfolio here.

The module offered a diverse range of learning activities that provided a comprehensive understanding of OOP concepts. Collaborative discussions included analysing factors influencing software reusability, and designing a metamodel for a humanoid robot. Various e-portfolio activities involved creating UML diagrams to model system structures, enhancing our understanding of object-oriented design principles, and completing exercises from 'Think Python' to reinforce core programming concepts and problem-solving skills. The summative assessments culminated in the development and implementation of a design document for a humanoid robot, providing a practical application of OOP principles.

The module presented a demanding yet rewarding workload, requiring significant engagement outside of weekends. Prior to this module, my experience with UML

diagrams in the Secure Software Development module had been limited without a deep understanding of their underlying principles. This module provided a valuable opportunity to delve into UML and grasp the purpose and significance of various diagram types. Furthermore, the concept of developing a humanoid robot presented an intellectually stimulating challenge. This module introduced me to new concepts and broadened my understanding of the practical applications of object oriented programming in complex and innovative systems.

So What?

In the previous module, I procrastinated on e-portfolio activities, leading to a stressful and inefficient learning experience. This time, I implemented a more structured approach. I prioritised completing e-portfolio activities during weekdays after work, allowing me to integrate learning with my professional experience. I grouped related activities from different units together, optimising my workflow and enhancing my understanding of interconnected concepts. Furthermore, I dedicated weekends to reading materials and participating in collaborative discussions, fostering a more balanced and sustainable learning pace. These strategies not only reduced stress but also improved my overall engagement with the course material.

Moreover, the module significantly deepened my understanding of UML diagrams, a concept I had encountered frequently throughout my software engineering career. While I had previously encountered UML diagrams in various contexts, such as technical articles and research journals, this module provided a deeper dive into their significance

and practical applications. Recognising the critical role of diagramming skills in software development, I actively sought to enhance my understanding of UML beyond the module's requirements. I proactively sought out additional resources, exploring various online tutorials and articles. This independent learning allowed me to apply my newfound knowledge by creating comprehensive UML diagrams for all features within my design document, demonstrating a practical application of these concepts.

The humanoid robot project presented a unique challenge, requiring the application of OOP principles learnt throughout the module. This summative assessment introduced me to new data structures, such as stacks and queues. Despite careful initial planning and the creation of a class diagram, I encountered deviations between the initial design and the actual implementation. While I initially overlooked the inclusion of setter and getter methods in my class diagram, focusing primarily on primary requirements, this experience mirrored real-world scenarios where unforeseen complexities often arise during the development process. I maintained a strong focus on adhering to core OOP principles, such as encapsulation, inheritance, polymorphism, and abstraction, while simultaneously striving to avoid code redundancy. This emphasis on best practices proved invaluable in ensuring the maintainability and scalability of the final system.

Now What?

The Object Oriented Programming module was a really interesting part of my master's program. Since I already knew some about OOP from my work, I found it pretty easy to grasp the main ideas. To make the most of it, I stayed on top of the assignments and

kept my e-portfolio updated, which helped me stay motivated and avoid feeling overwhelmed. I realised that trying to do every single activity wasn't realistic, so I focused on the ones that would be most helpful for my future career. I figured that by concentrating on the practical exercises, I could learn the skills that would be most valuable in my job. Overall, I think this approach helped me get the most out of the module while still keeping things manageable.

Looking back, I realise there's always room for improvement. I remember feeling a bit overwhelmed due to Collaborative Discussion 2. It was a new and challenging topic for me, and I worried that I wouldn't be able to contribute meaningfully. I had planned to revisit the discussion forum in unit 12 to learn from my peers' insights, but it seems like participation was limited. I noticed that only Todd Edge and Kelly Kitching had shared their thoughts. Moving forward, I'm determined to be more proactive in forum discussions. I'll share my ideas, even if I feel unsure or think they might not be the best. I believe that constructive feedback from my peers will help me learn and grow, and I'm eager to contribute more actively to the learning process.

I can definitely say this module was a valuable experience. I feel like I've grown as a software engineer thanks to what I learnt here. I'm more confident in my understanding of object oriented programming principles, and I feel better equipped to tackle complex software projects. I especially appreciate how the module encouraged me to think critically about design and consider different approaches to problem-solving.