

e-Portfolio website: <https://patzsantos.github.io/e-portfolio-uoee/>

OOP Github branch: <https://github.com/patzsantos/e-portfolio-uoee/tree/module2/oop>

Object Oriented Programming End of Module Reflective Piece

Word Count: 1045

I had no prior knowledge of Object-Oriented Programming (OOP) concepts before starting this module. Introduction to OOP paradigms acquainted me as to what classes and objects are. Initially, it was hard to fully understand how they are used in OOP. Nevertheless, as the module progressed, I had a better understanding of the logic behind them as I applied them in programming. Joque (2016) describes that in OOP, classes give meaning to how objects associate with each other in order to present the functionalities of programs. I filled out an [action plan](#) that gives an overview of learning goals, personal assessment, and target dates I have set for the OOP module.

Unified Modeling Language (UML), an integral part of our first summative assessment in Unit 7, gave me a clear picture as to how I want to implement the system of my driverless car. At the software designing stage, UML is useful in modeling the way elements respond with each other, which is vital when building superior software (Ozkaya & Kose, 2018). To guide me through the process of creating various UML models, I watched videos that expounded on their step-by-step development. Reading peer-reviewed studies about autonomous vehicles made me realise the importance of academic research in advancing my thought process, as my wider reading aided in my

selection of the operations and hardwares of the car. Through the system design assessment, I got acquainted with using UML syntax and open source tools like 'Visual Paradigm Online' when creating the diagrams.

Developers should choose which data structure would handle records and information best. In coding the system implementation of the driverless car for Unit 11, I employed a list for storing traffic signs, and a dictionary for holding keys and values of stop lights. Initially, I thought that I would not be able to use a queue. Despite that, I did, for Environment Perception, where First In First Out (FIFO) is the principle followed when an object for collision is avoided after detection. It was a concept I learned in a seminar and lecturecast. Furthermore, while writing the README file for the driverless car, I made sure to explain the system functions and code execution in a manner that even those without coding background can understand. This thoughtfulness will definitely be useful when I construct programs in the future.

In order to make sure that codes are properly running in line with their intended function, Sunil (2019) points out that testing is crucial. During the 'Launching Into Computer Science' module, we were required to test our code. Unfortunately, I only executed manual testing then. However, for my system implementation, I was finally able to apply unittests, together with embedding my coding files with assert statements. Though seemingly an extra chore due to executing them in each class, I now recognise that testing is necessary in making sure that the functions in codes are working properly.

My [skills matrix](#) reveals how I rated myself in terms of my professional, legal and ethical, social, technical, and subject application competencies. My limitations motivate me to do better, and the aptitudes I scored myself as proficient in, I will continue working on for my personal and professional growth. Codio exercises were honestly challenging for me to accomplish on my own. Even so, researching online enabled me to review and practice the coding activities that covered classes, functions, methods, polymorphism, and encapsulation in OOP. I have discovered that coding using the PyCharm environment was easier for me to perform. It was helpful in debugging and detecting problems, which included syntax and indentation errors. Math and logic has always been a struggle for me, that is why I find coding to be quite difficult. For that reason, I did not hesitate asking for help from more experienced computer science professionals such as my peers and tutor. Forums, such as '[Discussions on Python.org](#)', were especially helpful for me, as I gained a better understanding of programming and coding concepts I was unsure of earlier in the module.

Generally, I can manage my time well. Unfortunately, this was not the case during the first four units. Afterwards, I worked on this by starting my summative assessments and watching lecturecasts earlier in order for me to be well-prepared for the tasks at hand. Moving forward, I will strive to improve my productivity by using a journal to better keep track of my schedule and duties. Something I made sure to fit in my timetable this module was attending as many live seminars as I could. It gave me a better understanding of OOP concepts and expectations for the submissions. It was a good opportunity for me to interact with my peers and tutor too. After all, mindsets and

perspectives are widened when opinions and worldviews are shared (Chandler-Olcott & Hinchman, 2019).

As someone with an inclination for design, I was most confident in developing UML models for this module. I felt that I was able to represent how I intended to operate my driverless car through the diagrams I created. This newly acquired ability will help me plan out system architectures in the workforce, since according to Ciccozzi et al. (2019), Object Management Group (OMG) has made UML the standard in software engineering since 1997. In addition, building on my [e-Portfolio](#) is fulfilling as it serves as a record of my progress over this course. This major requirement keeps me disciplined because it reminds me to present at least one output per unit as an artefact of my learning. Moreover, Song (2021) acknowledges that making e-Portfolios stimulates independent learning, facilitates sharing of opinions and learnings amongst peers, and promotes mindfulness regarding due dates.

In conclusion, I have a better understanding of how clearly laid out planning can result in feasible implementation of systems through OOP. My interest in UML and building websites makes me ponder whether front-end jobs are something I should consider once I start working in the field of technology. Furthermore, now that I have a better grasp of how to apply OOP concepts in the real world, I can see myself becoming useful in project management. Perhaps, I can bridge the gap between back-end and front-end developers by effectively communicating how quality systems can be executed efficiently and excellently from a design standpoint.

References:

Chandler-Olcott, K. & Hinchman, K. (2019) Collaboration. *Journal of Adolescent & Adult Literacy*. 63(2): 125-126. Available from: <https://www.jstor.org/stable/48556225> [Accessed 15 February 2024].

Ciccozzi, F., Malavolta, I. & Selic, B. (2019). Execution of UML models: a systematic review of research and practice. *Softw Syst Model*. 18: 2313-2360. DOI: <https://doi-org.uniessexlib.idm.oclc.org/10.1007/s10270-018-0675-4>.

Joque, J. (2016). The Invention of the Object: Object Orientation and the Philosophical Development of Programming Languages. *Philos. Technol*. 29: 335-356. DOI: <https://doi-org.uniessexlib.idm.oclc.org/10.1007/s13347-016-0223-5>.

Ozkaya, M. & Kose, M.A. (2018). SawUML - UML-based, contractual software architectures and their formal analysis using SPIN. *Computer Languages, Systems & Structures*. 54: 71-94. DOI: <https://doi-org.uniessexlib.idm.oclc.org/10.1016/j.cl.2018.04.005>.

Song, B.K. (2021) E-portfolio implementation: Examining learners' perception of usefulness, self-directed learning process and value of learning. *Australasian Journal of Educational Technology* 37(1): 68-81. DOI: <https://doi.org/10.14742/ajet.6126>.

Sunil, K. (2019). *Clean Python: Elegant Coding in Python*. California, USA: Apress. Available from: <https://learning.oreilly.com/library/view/clean-python-elegant/9781484248782/>.