OOP_PCOM7E September 2022 Object Oriented Programming

Unit 12 Summative Assessment: e-Portfolio Submission

Reflection

Assignment Details

As a learning and development e-portfolio, specific requirements for what should be included in your e-portfolio are detailed in the “Module E-Portfolio Learning Activities” section. Elaborating on the requirements and the applicable grading criteria (please see the full outline on the Module Resources page), your e-portfolio should consist of:

- Your answers to and outputs from exercises carried out during the module, including all your coding scripts and system designs. Aim to showcase at least 1 artefact from each unit. (Application of Knowledge and Understanding weighted at 10%, Independent working weighted at 10%)

- Your analysis of the data/information gathered in the module, as well as information from applicable contributions to the collaborative discussion forums (Application of Knowledge and Understanding weighted at 10%)

- Your evaluation of your final project in weeks 7 and 11 (Independent working weighted at 10%).

- Reflection on (Criticality weighted at 40%).
  - Object-Orientated Programming based on what you have learned in this module.
  - Your work in the module.
  - The impact on your professional/personal development.

These are some exploratory questions to help you develop your reflection:

- Have you clearly described the topic of your project and the various aspects of project which are key to your reflection?
- Emotional Response and Analysis. Critically analyse yourself and your own behaviour during this project. How did working on the project make you feel? How did these emotions affect you and your work? Have you reviewed the views of others, your prior experience, and correctly referenced literature, in relation to your current behaviour and the work you’ve done?
- Learning and Changed Actions. Show that you have thought deeply about what exactly produced your learning (or a new way of thinking) while taking this project.
- Do you have evidence of skills and knowledge developed through this project? Provide a clear description of your evidence of learning during this project and how you can apply (or have applied) them in a real-world experience.
- You will need to evidence and evaluate your individual contribution to a group project, completing tasks to the required standard. You will also need to demonstrate clear participation in the team activities as it informs your reflection. Legible screenshots from your e-portfolio are acceptable and will not affect your word count.
- Is there an accurate use of citations and references throughout this piece of work? Have you demonstrated the required integrity in your submission?

Reflection

https://elsiewongpgcs.github.io/eportfolio/

In this module, I learn about program development using object oriented programming, i.e. use an object as a basic unit in the development and how different objects interact and affecting each other. The coding will firstly define a class where the object belongs to and the methods to be develop in the class.

Being a non-IT profession, I always think that software development is no more than program writing. Being a starter to programming, therefore I spent the first five to six weeks from the module starts in enhancing and reinforcing my basic Python knowledge such as syntax about list, dictionaries etc. thought some self-learning platforms W3School and other Python tutorials. It is somehow my style of learning because I do not believe in short cuts. And the fact that I am very new to Python made me spent a considerable time to get familiarized with the syntax. It is because with a better knowledge, I will have more confidence to pick up new knowledge from the lectures. That's why I am behind the schedule and just right on time to submit my assignment.

Throughout this module, I started to realize that software development does not equal to program writing. Coding is just one part of it. If you ask me now, I will say software development comprises four important parts: requirements gathering, system design, code writing, and final testing.

I believe software development starts with customer requirements gathering. Although the three operations are selected by myself, it is still challenging, not only because driverless is a new technique, but in fact I don't drive and surely do not have a car. I live in a city with well-developed public transportation system and go around easily without a private car. Therefore driving is just a leisure activities here but not a necessity. In order to close to reality, I need to first research on what functions driverless cars have. To be honest, I have difficult time in struggling with several functions and scopes that I would like to include but due to the fact that they are not realistic and so have been abandon.

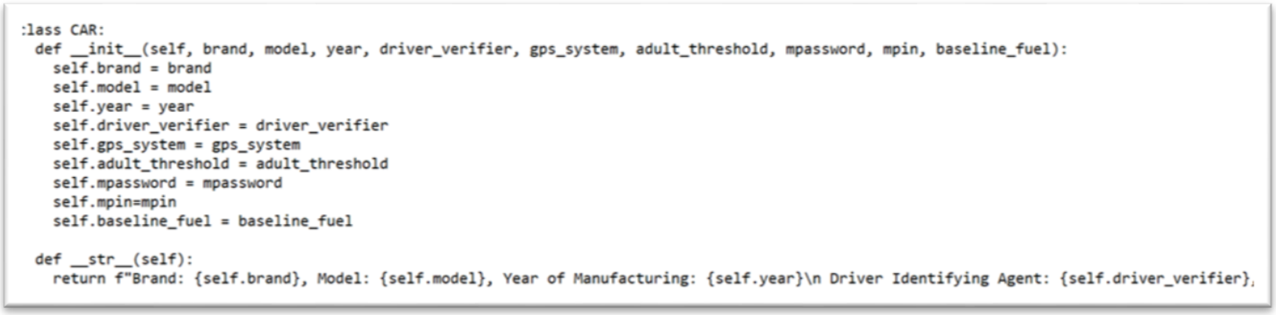
The final design of the three operations of the driverless car are (1) starting, (2) auto-navigation and (3) parking a car. Starting and parking means the initiation and end of the driving process whereas auto-navigation and traffic watching is one of the most significant part of driverless car (Reddy, 2019).

In the Starting module, the first thing to do is verified the driver's identity. This is performed by a three step authentication which is an actual system employing by IDnow (IDnow, N.D.). Child passenger safety is a major concern and therefore weight sensor is used to identify child passengers. In my setting, exception is allowed by the driver because in real-life, the seat may actually be occupied by an under-weighted adult or it is not occupied by a person. In the next operation, which is navigator, the GPS and traffic watch functions of a real-life system, Carmenta TrafficWatch (Carmenta TrafficWatch, N.D.) are used. In this case a scenario of change in traffic condition is assumed to show the function of the system. The final system is simulating the auto-parking solution developing by Baidu (New China TV, N.D.). The parking solution involves a selection of car parking space and obstacle handling. I believe the scope of the autonomous driving project is quite comprehensive and reasonably close to actual.

After gathering of the system requirements, it goes to the system design stage. The diagrams are amazing tools in system development. System requirements and interaction between objects are broken down into various diagrams, each of which taken care of a significant part of the coding development. It makes a complicated idea simple, easy and documented in an orderly manner. Honestly I am not good at turning my ideas into vistualised formats and therefore I am weak in drawing flow charts and diagrams, but I must admit that the pictorial approach of these diagrams is simple and easy to understand. You only have to understand how they are used and the standards in writing these diagrams.

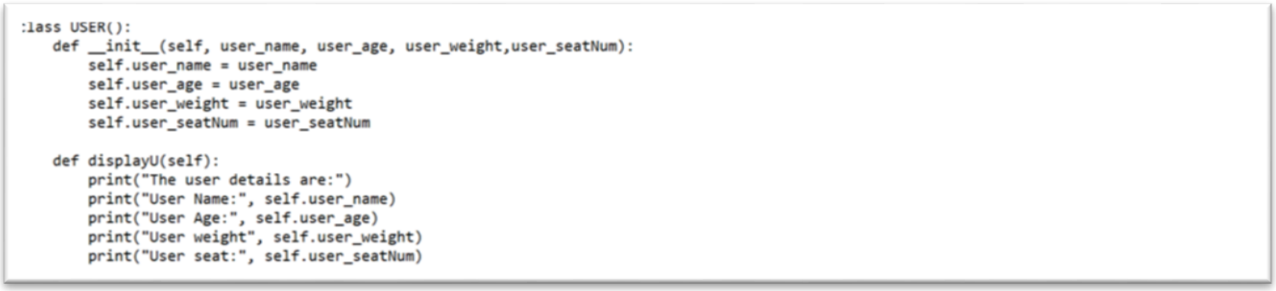
Unfortunately, I am quite wrong in understanding these diagram tools, which is reflect in the comments from my tutor after submitting assignment one (system design). After obtaining the feedback from tutor, I have re-visited the standards for those diagrams and re-write all diagrams when I submit the assignment two (system implementation).

Thereafter, it comes to code writing. Learning new syntax and how to obtain same result from different coding makes me feel thrilling. For example, after defining the class, I like to add the coding for printing the object parameters out so that I will know that if the data of the objects have been entered correctly. In showing such object parameters, I have tried different syntax "return" and "print" as follows:



```
:lass CAR:
  def __init__(self, brand, model, year, driver_verifier, gps_system, adult_threshold, mpassword, mpin, baseline_fuel):
    self.brand = brand
    self.model = model
    self.year = year
    self.driver_verifier = driver_verifier
    self.gps_system = gps_system
    self.adult_threshold = adult_threshold
    self.mpassword = mpassword
    self.mpin=mpin
    self.baseline_fuel = baseline_fuel

  def __str__(self):
    return f"Brand: {self.brand}, Model: {self.model}, Year of Manufacturing: {self.year}\n Driver Identifying Agent: {self.driver_verifier},
```



```
:lass USER():
    def __init__(self, user_name, user_age, user_weight,user_seatNum):
        self.user_name = user_name
        self.user_age = user_age
        self.user_weight = user_weight
        self.user_seatNum = user_seatNum

    def displayU(self):
        print("The user details are:")
        print("User Name:", self.user_name)
        print("User Age:", self.user_age)
        print("User weight", self.user_weight)
        print("User seat:", self.user_seatNum)
```

Honestly, code writing will be very enjoyable - if you do not have to do debugging. Some bugs are clerical errors and some are not. Common clerical errors are: incorrect indent, confusion in the use of "=" and "==" and missing of ":" at the end of "if" or "while". Sometimes silly mistakes as such make me really angry about myself but I know debugging needs a lot of patience. Fortunately the Codio program is helpful in pointing out such clerical errors such as errors.

Another coding technique that I have learnt is the public and private variables, i.e. variables that defined in a function are private and can only be used in that function.

To conclude, the key takeaway from this module is that from the process of software development, I am able to appreciate the importance of requirements setting. This extends to the importance of understanding client's business because by knowing how the business is running, I could stand at the client's angle to think about their requirements. The better I understand the client's business, the more precise final product could be produced. As coding is usually in modular basis, if time allows, I could test the result and present to client when one module is finished. This will help to narrow down the expectation gaps and also help client to find out some additional requirements so that they will not raise them at the last minute.

I myself is not working in IT industry, but the same applied when in my accounting field because understanding the client's business logic will help understanding and interpreting the client's figures.

Reference:

Reddy, P. P. (2019) Driverless Car: Software Modelling and Design using Python and Tensorflow.

IDnow. (N.D.) Fully automated for the fastest identity verification [Available from:https:// www.idnow.io/autoident-express-fully-automated-identity-verification-with-a-powerful-document-learning-] Accessed on 2 November 2022.

Carmenta TrafficWatch. (N.D.) [Available from: https://carmenta.com/en/ automotive/carmenta-trafficwatch/?gclid=Cj0KCQjwk5ibBhDqARIsACzmgLQGjK88miy-Ty4_YD_bKCKzoD-Qu7S5Kv54yQn1yXjPwzSlYyqy7kYaAhchEALw_wcB] Accessed on 1 November 2022.

New China TV. (N.D.) How self-driving car completes parking [Available from: https://www.youtube.com/watch?v=GKxFZQxCB8M] Accessed on 6 November 2022.