e-Portfolio Activities

1. **Review** the article by Di Silvestro & Nadir (2021). Discuss one aspect of this article which you find unexpected.

My observations:

This article highlights multiple subjects that cover various aspects of adult education, including how ePortfolios might promote deeper learning and reflection in graduate adult education. The article also looks at Saudi Arabian adult learners' motivations for learning. Through the intersectionality theory perspective, it focuses on the acculturation experiences of Syrian Muslim refugee women in the United States. The article examines these women's difficulties and the significance of offering immigrants and refugees' long-term education programmes to aid their effective acculturation. The research also emphasises how ePortfolios can be a valuable addition to graduate adult education programmes to promote reflective and in-depth learning.

The fact that students were taken aback by the need to work together when creating their ePortfolios is an unexpected component of this essay. In my opinion and the students' positive experiences, building ePortfolios effectively enhances our learning. Doing so requires us to read widely and conduct extensive research, which helps us properly understand and master the subject matter. Speaking from personal experience, I have encountered more diverse research topics since I began to develop my ePortfolio, which encourages and pleases me as I read to respond to questions from the units I am studying.

2. **Develop a Python program and apply protected and unprotected variables.**

Protected variables are data members of a class that can only be accessed within that class and its descendants. In Python, there are no "Public" instance variables. For

determining who has access control of a data member in a class, however, we use the underscore '_' symbol. Thus, any member prefixed with an underscore, whether a function, method, or data member, must be handled as a non-public API or any Python code component (GeeksforGeeks, 2020).

Access control is a concept in most object-oriented programming languages; this has something to do with abstraction. Some attributes and methods on an object are designated as private, meaning only that object can access them. Others are marked protected, meaning only that class and its subclasses can access them. The rest are public, meaning any other object can access them (Phillips, 2018).

```
class Person:
    def __init__(self, name, age):
        self.name = name                    # Unprotected variable
        self._age = age                     # Protected variable

    def get_age(self):
        return self._age

    def set_age(self, age):
        self._age = age


person1 = Person("Hainadine", 45)

print(person1.name)                         # Output: Hainadine
print(person1.get_age())                    # Output: 45

person1._age = 40                           # Unprotected access to protected variable
print(person1.get_age())                    # Output: 40

person1.set_age(50)                         # Protected access to protected variable
print(person1.get_age())                    # Output: 50
```

References:

GeeksforGeeks. (2020). *Protected variable in Python*. [online] Available at: https://www.geeksforgeeks.org/protected-variable-in-python/ [Accessed 6 May 2023].

Phillips, D. (2018). *Python 3 object-oriented programming: Build robust and maintainable software with object-oriented design patterns in Python 3.8*. Packt Publishing Ltd.

Unit 1 - Reflection

During this first week of studying Object-oriented Programming, I delved into the fascinating world of Python programming. The focus of this first week was to gain a comprehensive understanding of the evolution of programming languages, specifically towards object orientation and the subsequent advancements made to address the challenges faced when programming in an object-oriented manner.

The Lecture cast, *Introduction to Python and the Object-Oriented Programming Philosophy*, was a great way to learn about all Object-oriented programming, as it provided insights into the reasons behind the emergence of object-oriented programming (OOP) paradigms. OOP introduced a new way of structuring code, emphasising the organisation of data and behaviour into cohesive entities called objects—this shift from procedural to object-oriented programming allowed for more modular, reusable, and scalable code.

Exploring the 'Evolution of Programming Languages', I sincerely appreciated Python's elegance and versatility. It was intriguing to see how Python combines the benefits of object-oriented programming with the flexibility of a dynamically-typed language.

Furthermore, studying the challenges encountered when programming object-oriented provided valuable insights into the complexity developers face. While OOP brings numerous advantages, such as encapsulation, inheritance, and polymorphism, it also presents challenges, such as maintaining code clarity, managing dependencies, and designing effective class hierarchies.

Understanding these challenges gave me a deeper appreciation for best practices in object-oriented design and the importance of writing modular, reusable, and maintainable code. I realised that designing object-oriented software systems requires

carefully considering the relationships between objects, identifying appropriate abstractions, and creating well-defined interfaces.

Overall, this first week of studying Object-oriented Programming provided a strong foundation in Python programming and an exploration of the evolution towards object-oriented programming.

Throughout the first week of this module, I devoted my attention to studying and devised a structured plan for the next 12 weeks. My project breaks down my study approach into reading, researching, note-taking, and coding activities. Since coding is a key component of this module, I aim to absorb as much knowledge as possible about Python coding. Although I have more experience with SQL, I am a novice in Python. Despite having some certifications and participating in small Python projects, I aim to enhance my skills through this course. I plan to assimilate this course's theoretical concepts and essential definitions to become a skilled professional with in-depth knowledge, which will help me excel as a data analyst or project manager.