A Reflection on Coding, Languages and IDE’s on a Postgraduate Course in Computer Science

The OOP module presented several coding exercises that introduced the concepts of Object Oriented Programming and best practices in software development. In my professional role as a solution architect, these concepts are a critical part of my role and learning and have already formed part of my thinking in delivering software solutions. One current project I am working on required using an operational PowerShell script to manage file artefacts on an operating system. I have applied the knowledge learnt through these assignments. By developing a more modular approach in the script with the use of variables, and data structures and separating the code into smaller more tactical functions it has formed part of a library of reusable scripts that can be used by other resources. One main achievement was the removal of hardcoded values for variables but rather the use of objects and collections from JSON files that hold global application settings. The code I wrote was compared to a previously supplied script and became the preferred solution because of its reusability and modularity.

I work in a professional services organisation where we deliver COTS (Configurable Off The Shelf) applications that can also have their code base extended for customisation. One of the key drivers is configuration first and customisation second. As a global organisation, we do not adhere to any standards in how we deliver customisation for projects so the concepts of OOP presented in this course allow me to adopt a more pragmatic approach to delivery where I can drive the adoption of OOP principles and at the same time drive standards in our code. It will also allow us to work within the standards of the core application SDLC and its approach to OOP and the core product.

One of the key challenges for me was the use of the Codio IDE. Although I am not a developer, I am more familiar with IDEs like Visual Studio so became more comfortable writing and testing my code locally and then transferring to Codio. As a student, it did however present the challenge of the uncertainty of my code working in another IDE seamlessly. Whilst not critical it did force me to consider even further the structure of code, complexity and transportability and especially how locally written code may work in a virtualised environment or different operating systems. In particular, the summative assignment for Python development required us to use Codio but as the assignment had no criteria for configuring a virtual environment to execute code it was disappointing to find that the IDE would not run the code I had written nor had any simple setup instructions to do this. It could be argued that all the code should have been written in Codio to start but the code in itself should be transportable between environments so a documented process for using the Codio IDE platform would have been more beneficial.

One immediate thought was for a piece of work I deployed into AWS as an automation script. Reflecting on this now and what has been learnt in this module in hindsight the code would have still been delivered in the same way but could be dramatically improved. In implementing some of the concepts learnt the code could have been more modular, elegant and better inline commentary and documentation.

This module and more specifically the coding aspects have forced me to reflect on the aim of this course and how standards and OOP positively influence coding in the real world and my professional career.