**Prompt: you will need to briefly explain how you used these concepts into your program. You will need to provide examples from your code. Please be as brief as possible.**

Method overloading

* In my project, method overloading was used in areas where data needs to be handled differently for classes that share a parent class
* For example, sharing the same “UserParent”, the user “Admin” only needs username, password and identity properties. While student users need to have more properties such as name, and registered courses etc.

Method overriding

* Method overriding can be useful when a basic method is provided in parent class and could be customized to meet specific needs.
* In this project, I did not use overriding except for overriding functions from interfaces.
* The reason is that there isn’t much overlapping functions between Admin App and Students App. The only exceptions are the methods in their parent class “Utility”, which don’t really need to be overridden.

Abstract Class/Methods

* Abstract methods are used to make my programs more organized and to help me develop the application more efficiently.
* For example, I used interface for my “Admin App” and “Student App”. I outlined the major functions in their respective interfaces and implemented them in the classes. This helped me to keep track of methods in otherwise lengthy and messy workspaces and easier to spot potential issues

Inheritance

* Both of my “Admin App” and “Student App” inherited from a common “Utility” class
* I did this because these two apps share some common functions.

Polymorphism

* I implemented the concept of polymorphism in areas where I think the object’s form is not absolute case by case.
* I declared User object with “UserParent” class. While in the runtime, a user can be an Admin or a student based on their identity.

Encapsulation

* Encapsulation helped me to better organize data and access them only when needed.
* The restriction level of almost all my variables are private or package-private.
* For example, the data in user objects are accessed or modified through getters and setters.

The concept of ADT (Abstract Data Types)

* ADT makes my main class as streamlined as possible and easier to maintain.
* I made separate “Admin App” and “Student App” to handle specific functions and used main to only read/write files and present main menu