Programming with Classes

Classes can be likened to recipes. They provide a set of steps to be followed, include attributes and behaviours which are used in programming.

Programming with classes includes four concepts including:

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

Abstraction

Abstraction is the process of turning complex ideas into simple ones. It is removing characteristics from something so that only the essential ones remain.

The concept of abstraction was applicable in the design of the journal application which included the Entry, Journal and PromptGenerator class. The concept of abstraction is such that we can create custom data types from the classes which we create hence setting the basis of programming with classes. The Journal class is made possible by creating instances of the entry class.

Encapsulation

Encapsulation refers to the practice of bundling data (variables) and methods (functions) that operate on that data into a single unit – normally a class – and restricts direct access to some of the object's components.

I used the concept of encapsulation in the scripture memorizer to make certain parts of the Word, Reference and Scripture classes private meaning only the methods in the classes could access the attributes of the class. Some of the methods were also set to private to prevent use by other classes. The concept of encapsulation is essential as in the event that a program breaks down, we are able to identify the places where an error could be.

Inheritance

Inheritance is a programming concept which as the name suggests, involves the relationship between a parent class (also known as the base class) and a child class (also known as the derived class).

I used the concept of inheritance to develop the Mindfulness program. It involved a variety of activities. The breathing activity, listing activity and reflection activity classes all had certain attributes and behaviours which they share. Hence they all inherited from the Activity class in effect reducing redundancy and increasing efficiency.

Polymorphism

Polymorphism is the ability to take on many forms. In programming, this principle is shown when one line of code can have different behaviour depending on the context.

I used the concept of polymorphism to develop the EternalGoal program. It was essential in developing the program such that a method defined in the base class could be modified to perform a task and function appropriately in regard to context. Hence enabling a method to function in regard to the class object in question.