

## **Abstraction**

Abstraction is used as a planning method to plan a large programming solution where it is more oriented and simple to understand, it is to hiding details to focus on fewer concept. Abstraction method is to break down into a smaller object that does specific task. The detail of the task does not be needed but only the key feature from the object is used. For example, in SwinGame we broke the program down to pieces to a few classes such as Drawing class, Circle class, Line class and Rectangle Class, Drawing class does not know the parameter behind Rectangle class but know the key feature of Rectangle class to draw a rectangle, it is to hiding details to focus on fewer concept.

## **Encapsulation**

Encapsulation is a way of hiding the internal mechanisms and data structure of a software component behind a defined interface. It is to ensure the user of an object on dependently on its interface, the implementation could be changing overtime. Encapsulation is to encapsulate code and data feature within a class. We can picture object as a capsule where it is a cohesive entity containing information hiding. It gives us the use of public interface and private implementation. Encapsulation is a packaging of the object as a class which contain private fields (Variable, things it know) and public method(things it can do), a private field known as the information needed to perform the functionality of the object it has all the logic behind the object but private field is kept protected where other object cannot access it directly but instead access it through a method call property. Property allows other object to access and modify. For example, in the Swin Game it requires a lot of encapsulation where as each shape is encapsulating into individual classes such as Circle Class, it contain radius (things it knows) as private field and its draw circle function (things it can do) as public method.

## **Inheritance**

Inheritance is a relationship between classes, one class is a derived class where it can inherit all the feature from the base class. A derived class is a specialized version of the base class, or alternatively the base class is a more general version of the derived class. By using inheritance we can create a new derived class that inherit all the behavior of its parent class(base class) and provide inheritance of all fundamental of base class. Derived class are able to extend the ability of base class by adding more method into the child class. For example, Circle class are the derived class of the base class Shape class, it have specific responsibility to draw a circle but it inherits the method and properties from their base class Shape class.

## **Polymorphism**

Polymorphism enable derived class to be use as an individual base class with similar but different functionality. Polymorphism allows child class to override the method in parent class to do specific task. Polymorphism also allows subcasting of classes. Polymorphism is a type of overriding and overloading between derived class and base class. For example Circle and line class inherit the property of draw class but each of them override Draw method to draw their desire shape.