**TIGHT VS LOOSE COUPLING**

Tight coupling is when two classes have many dependencies on each other. Loose coupling on the other hand is when two classes are not dependent on each other. The latter promotes greater reusability for classes and makes maintaining them easier. If you had to change one class that was tightly coupled, you would have to makes changes in the other coupled as well. With loose coupling, making changes in one class would not affect the other classes.

**ENCAPSULATION**

This term refers to the fact that each class should have only data concerning itself. Encapsulation involves hiding information about the state or fields of an object from being directly accessed by other classes. Only that class has access to that information and if other classes want to access it, it will be indirect access by asking the owner class.

**CLASS COHESION**

This term pertains to the degree in which the class’ methods and data correlate with one another. A class is highly cohesive if its functions and data have a lot in common and if its methods have a small amount of activity. High cohesion makes classes less complex and easier to maintain. It makes maintenance and changes easier as changes will affect a very small portion of the code. It aims to reduce each method to a single concept, rather than them having multiple functions.

**LAW OF DEMETER**

This involves classes having limited information on other classes. A class should only know a limited amount of information about other classes, and even then, those classes must be closely related to that class. Ex. A poker class should not know anything about cards, just hands. The hand needs to know information about cards however.