Composition
===========
To qualify as a composition, an object and a part must have the following relationship:

The part (member) is part of the object (class)
The part (member) can only belong to one object (class) at a time
The part (member) has its existence managed by the object (class)
The part (member) does not know about the existence of the object (class)

Aggregation
===========
To qualify as an aggregation, a whole object and its parts must have the following relationship:

The part (member) is part of the object (class)
The part (member) can (if desired) belong to more than one object (class) at a time
The part (member) does not have its existence managed by the object (class)
The part (member) does not know about the existence of the object (class)

Association
===========
To qualify as an association, an object and another object must have the following relationship:

The associated object (member) is otherwise unrelated to the object (class)
The associated object (member) can belong to more than one object (class) at a time
The associated object (member) does not have its existence managed by the object (class)
The associated object (member) may or may not know about the existence of the object (class)

Dependency
==========
A dependency occurs when one object invokes another object’s functionality in order to accomplish some specific task. This is a weaker relationship than an association, but still, any change to object being depended upon may break functionality in the (dependent) caller. A dependency is always a unidirectional relationship.
Implemented through function members rather data members.