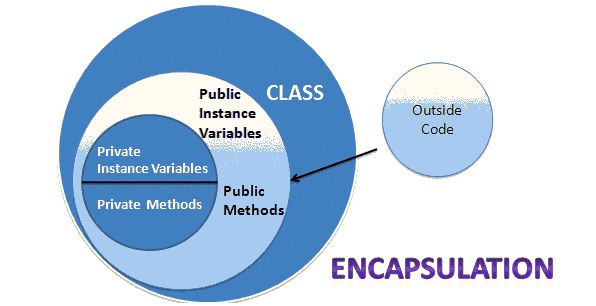
**Class Abstraction** is one of the feature of in OOP, where you show only relevant details to the user and hide irrelevant details.

**Attributes / Properties- Attributes** are one of the key features of modern C++ which allows the programmer to**specify additional information to the compiler to enforce constraints (conditions), optimize certain pieces of code or do some specific code generation.** A **property** is a member that behaves as if it were a field. Properties are used to implement data hiding within the class itself.

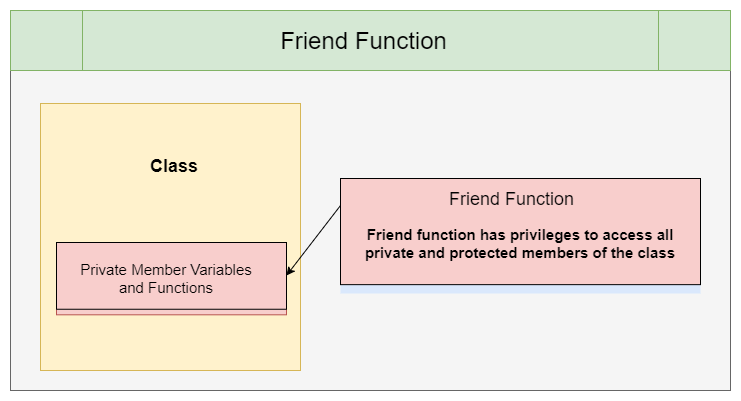
**Class Variable** is any variable declared with the modifier of which a single copy exists, regardless of how many instances of the class exist.

**Composition** Object Composition is useful in a C++ context because it allows us to create complex classes by combining simpler, more easily manageable parts. This reduces complexity, and allows us to write code faster and with less errors because we can reuse code that has already been written, tested, and verified as working.

**Encapsulation** It describes the idea of bundling data and methods that work on that data within one unit. This concept is also often used to hide the internal representation, or state, of an object from the outside. This is called[information hiding](https://en.wikipedia.org/wiki/Encapsulation_(computer_programming)#An_information-hiding_mechanism).



**Friends**

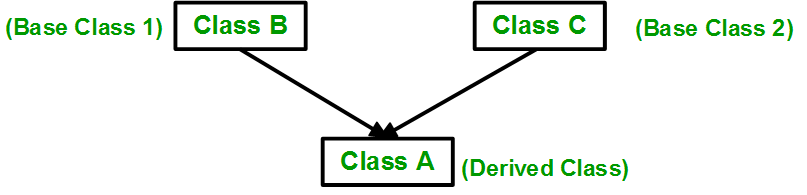


**Inheritance** is the capability of a class to derive properties and characteristics from another class.

**Instance Variable** are non-static variables and are declared in a class outside any method, constructor or block.

**Member Variable** is a variable (an object) that is part of a class.

**Method** are functions that belongs to the class.

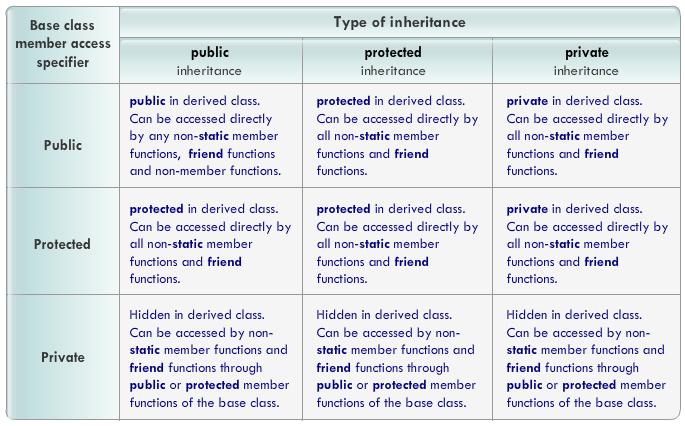
**Multiple Inheritance** is a feature of **C++** where a class can **inherit** from more than one classes.

An **Object** is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated memory is allocated.

**Overloading** is a C++ feature that allows us to have more than one function having same name but different parameter list,

**Polymorphism** is a feature of [OOPs](https://beginnersbook.com/2017/08/cpp-oops-concepts/) that allows the object to behave differently in different conditions.

**Public / Private / Protected- Public**: All the class members declared under public will be available to everyone. **Private**: The class members declared as private can be accessed only by the functions inside the class. **Protected** access modifier is similar to that of private access modifiers, the difference is that the class member declared as Protected are inaccessible outside the class but they can be accessed by any subclass (derived class) of that class.



**Virtual** functions are member functions whose behavior can be overridden in derived classes.