Overloading vs Overriding :

* Declaring more than one function with the same name but with different parameters where as Declaring a function in the base class and derived class with the same name and parameter.
* One is compile time polymorphism where as the other is run-time polymorphism.
* Inheritance is involved in overriding.

Abstract class :

A class containing at-least one pure virtual function. Designed to act as a base class.

Syntax: class A{

Public:

Virtual void show()=0;

};

Multiple Inheritance:

Inheritance in which a class in derived from multiple base classes.

Class A{};

Class B{};

Class C: public A, public B{};

Class vs Object:

* A class is a group of data members and member functions used to hide the abstract details from the user whereas an object is an instance of the class.
* Declared using keyword class whereas an object is declared or created using the respective class name.
* Memory is not allocated at the time of declaration of a class where as memory is allocated for the creation of an object.
* Multiple object can be created for the same class.

Public vs Protected variable:

* Accessible throughout the program whereas not accessible throughout the program like private members but can be inherited by the class derived from the base class like public members.
* Used while declaring member functions in a class so that they are accessible in the main method whereas protected is used in the base class for declaring data members to be inherited by the derived/child class.

Constructor:

Special type of function/method which is used to assign values directly to the object created which can be zero constructor/parameterised/copy.

Syntax of copy constructor:

className(className &var);

Initialization of static variable:

Int className :: var=0;

Use of super keyword:

Used to call the base-Class/ member functions of the base class

This pointer:

Also known as deference operator which allows us to get the value stored at the address held by the pointer.

Types of Inheritance:

* Single Inheritance: One class derived from a single base class
* Multiple Inheritance: One class derived from multiple base classes
* Multilevel Inheritance:One class derived from its parent class which is derived from another base class/ grand parent
* Hierarchical Inheritance: Two or more classes derived from a single class
* Multi -Path Inheritance: One class derived from two base classes which are derived from another single base class.

Use of friend function:

* To access all the non public members of the class.
* Used as a bridge between two classes by operating object of 2 different classes.
* Increase versatility of the program and enhances encapsulation.