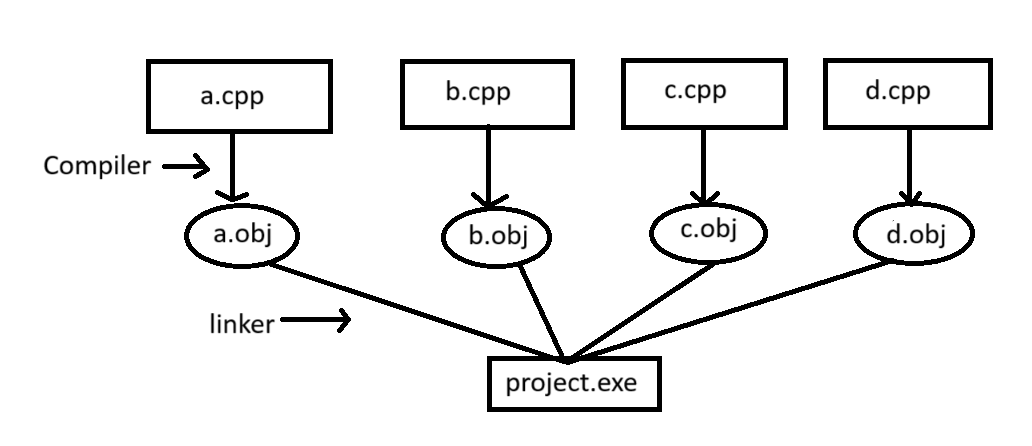
Basic Understanding on cpp:



IDE-Integrated Development Environment

In order to convert our cpp language to machine language to make it understand to cpu->compiler

To convert cpp to object files-> compiler

To link all object files and convert to execute file to run -> linker

These all are compiled in a single IDE -> visual studio

Cpp:  
**link: https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/**

1. OOPs concept: Class, Object, Polymorphism, inheritance, abstraction, encapsulation
2. STL:Containers
3. Design patterns
4. Memory management

**Class**:  
 It is a **user-defined data type**that act as a blueprint representing a group of objects which share some common properties and behaviours. These properties are stored as data members, and the behaviour is represented by member functions.

**Objects**:  
 An [**Object**](https://www.geeksforgeeks.org/c-classes-and-objects/)is an identifiable actual entity with some characteristics and behaviour. In C++, it is an instance of a class.

When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

Objects take up space in memory and have an associated address like a record in pascal or structure or union. When a program is executed, the objects interact by sending messages to one another. Each object contains data and code to manipulate the data. Objects can interact without having to know details of each other's data or code, it is sufficient to know the type of message accepted and the type of response returned by the objects.

OOPs concepts:

1. Inheritance: **link- https://www.geeksforgeeks.org/inheritance-in-c/**
2. Encapsulation
3. Polymorphism
4. Abstraction

**Inheritence**:

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

**Syntax**

class DerivedClass : “mode\_of\_inheritance” BaseClass {

*// Body of the Derived Class*

};

| **Mode** | **Description** |
| --- | --- |
| **Public Inheritance Mode** | Public in base->public in derived Protected in base-> protected in derived |
| **Protected Inheritance Mode** | Public in base->protected in derived Protected in base-> protected in derived |
| **Private Inheritance Mode** | Public in base->protected in derived Protected in base-> protected in derived  Private mode is the **default mode** that is applied when we don't specify any mode. |

*The private members in the base class cannot be directly accessed in the derived class, while protected and public members can be directly accessed. To access or update the private members of the base class in derived class, we have to use the corresponding* [*getter and setter functions*](https://www.geeksforgeeks.org/write-getter-and-setter-methods-in-cpp/) *of the base class or declare the derived class as friend class.*

**Types Of Inheritance in C++**

The inheritance can be classified on the basis of the relationship between the derived class and the base class. In C++, we have **5 types of inheritances:**

* Single inheritance
* Multilevel inheritance
* Multiple inheritance
* Hierarchical inheritance
* Hybrid inheritance