In object-oriented programming, a public member of a class can be accessed from outside the class, anywhere within the scope of the class object. Public members provide an interface for interacting with the class and its objects.In addition to public members, classes in C++ can have private or protected members..

In C++, the `inline` keyword is used to suggest to the compiler that a function should be expanded at each point where it is called, instead of being called as a separate function. This can potentially improve performance by avoiding the overhead of function calls. To inline a function, the `inline` keyword is placed before the function name, and the function is defined before any calls to it.

C++ allows for operator overloading, which means you can define more than one definition for an operator within the same scope. Overloaded operators are essentially functions with special names, starting with the keyword "operator" followed by the symbol of the operator being defined.Similar to regular functions, overloaded operators have a return type and a parameter list.

In C++, by declaring a member function as static, you make it independent of any specific object of the class. A static member function can be called even if no objects of the class exist. Static functions are accessed using the class name and the scope resolution operator (`::`).When a member function is declared as static, it can only access other static members of the class, including static data members and other static member functions.

In C++, when creating a class, instead of starting from scratch, you can designate that the new class should inherit the members (data members and member functions) of an existing class. The existing class is known as the base class, and the new class is referred to as the derived class.Inheritance allows the derived class to inherit and reuse the members of the base class, reducing code duplication and promoting code reusability.

When drawing CRC (Class Responsibility Collaboration) cards, relationships between classes can be categorized into five types. One of these categories is inheritance, also known as generalization.Inheritance represents an "is-a-kind-of" relationship between classes. It establishes a relationship between a general thing, referred to as the superclass or parent class, and a more specific kind of thing, known as the subclass or child class.

In object-oriented programming, a "whole/part" relationship represents a strong connection between two classes. It implies that one class represents a larger entity (the "whole"), which is composed of smaller entities (the "parts"). This means that an object of the whole class contains objects of the part class.There are two types of "whole/part" relationships: aggregation and composition

associated with a whole.Understanding these relationships helps in modeling and designing object-oriented systems, as it allows for proper encapsulation and representation of complex systems. It provides a way to depict the hierarchical structure and interactions between different classes in a clear and organized manner.

In object-oriented programming, a dependency is a weaker form of relationship between two classes. It indicates that one class depends on another class because it uses it at some point in time. A change in the depended class may affect the dependent class, but not necessarily the other way around.Graphically, a dependency is represented by a dashed directed line, pointing from the dependent class to the class it depends on.