Access Specifiers

Access Specifier:-

Access Specifiers in a class are used to assign access to the class members. It sets some restrictions on the class members from accessing the outside functions directly. Access specifiers have a vital role in securing data from unauthorized access.

It allows us to determine which class members are accessible to other classes and functions and which are not.
There are three types of access modifiers available in C++:
- Public
- Private
- Protected

➔ **Public:** All the class members with a public modifier can be accessed from anywhere(inside and outside the class).

```
class person{
    public:
    string name;
};
```

➔ **Private:** All the class members with a private modifier can only be accessed by the member function inside the class.

```
class person{
    private:
    int fb_password;
};
```

➔ **Protected:** The access level of a protected modifier is within the class and outside the class through child class (or subclass). If you do not make the child class, it cannot be accessed outside the class.

```
class person{
    protected:
    string assets;
};
```

➔ **By default**, in C++, all class members are private if you don't specify an access specifier.

```
class person{
    int name; //by default, it is a private data member
};
```

Example using smartphone class:-

```
class smartphone{
    //Data members
    string model;  // by default private

    public:
    int year_of_manufacture; // public data member

    protected:
    string company_name;  // protected data member

    private:
    int password  // private data member

    //methods
    private:
    void unlock_lockscreen(){
        //private method
    }

    public:
    void call(){
        //public method
    }

    protected:
    void about_phone(){
```

```
        //protected method
    }
};
```

Scope Table:-

	Inside class	Child (or sub-class)	Outside class
Public	✔	✔	✔
Protected	✔	✔	✘
Private	✔	✘	✘