Tutorial 39 - Protected Access Modifier in C++

Definition:

- The **protected** access modifier allows members to be accessed by **derived classes**, unlike private members which cannot be inherited.
- Protected members cannot be accessed outside the class hierarchy.

Access Behavior Based on Derivation Modes

Access Modifier	Public Derivation	Private Derivation	Protected Derivation
Private Members	Not Inherited	Not Inherited	Not Inherited
Protected Members	Protected	Private	Protected
Public Members	Public	Private	Protected

Key Observations:

1. **Public Derivation**:
   - Protected members remain **protected** in the derived class.
   - Public members remain **public** in the derived class.
2. **Private Derivation**:
   - Protected members become **private** in the derived class.
   - Public members also become **private**.
3. **Protected Derivation**:
   - Protected members remain **protected**.
   - Public members become **protected**.

Code Example

Code Snippet 1: Protected Access Modifier

```
#include <iostream>
using namespace std;
class Base {
protected:
    int a; // Protected member
private:
    int b; // Private member
};
class Derived : protected Base {
    // 'a' is accessible within the derived class but remains protected
    // 'b' is not inherited
};
int main() {
    Base b;
    Derived d;
    // cout << d.a; // Error: 'a' is protected and cannot be accessed outside the class hierarchy
    return 0;
}

```

Explanation:

1. **Base Class**:
   - Contains:
     - Protected member: `a`.
     - Private member: `b` (not inherited).
2. **Derived Class**:
   - Inherits `Base` in **protected mode**.
   - `a` remains **protected**.
   - `b` is not inherited.
3. **Main Program**:
   - Attempting to access `a` using object `d` throws an error since `a` is protected and cannot be accessed outside the class.

Short Notes for Notebook

Protected Access Modifier:

1. Allows access to derived classes but not outside the class hierarchy.
2. **Private members** are never inherited.

Behavior of Members Based on Derivation Mode:

Mode	Private Members	Protected Members	Public Members
Public	Not Inherited	Protected	Public
Private	Not Inherited	Private	Private
Protected	Not Inherited	Protected	Protected

Code Example:

```
class Base {
protected:
    int a; // Accessible in derived classes
private:
    int b; // Not inherited
};
class Derived : protected Base {};

```

Key Points:

1. **Public Derivation**:

    - Protected → Protected, Public → Public.

2. **Private Derivation**:
    - Protected → Private, Public → Private.
3. **Protected Derivation**:
    - Protected → Protected, Public → Protected.

Output Behavior:

- Attempting to access protected members from outside the class causes a **compilation error**.