

Object Oriented Programming (OOP)

Classes & Objects

```
class Box
{
  private:
    // data members
  public:
    // member functions
};
```

← **Class**

Box b1

Box b2

Box b3

← **Objects**

Accessing Data Members and Member functions

- Data members and member functions can be accessed in similar way the member of structure is accessed using member operator(.).
- **Syntax for accessing the member function of class**

Object_Name . Function_Name;

- **Syntax for accessing the Data member of class**

Object_Name . Data_Memeber

Note : We cannot access the data member of the class in other class if the data members are private or protected.

Example :

```
Class Employee {
Public: int salary;    // data member
Void sal() {         // member function
    cout<<"Enter Salary: ";
    cin>>salary;
    cout<<"Salary: "<<salary;
}
};

int main() {
Employee E; //creating an object of
E.sal(); //Calling the function using object of Employee
}
```

Example:

```
class Box {  
    public:  
        float length;  
        float breadth;  
        float height;  
};  
int main( ) {  
    Box Box1;  
    Box Box2;  
    float volume = 0.0;  
  
    // box 1 specification  
    Box1.height = 5.0;  
    Box1.length = 6.0;  
    Box1.breadth = 7.0;
```

```
// box 2 specification
Box2.height = 10.0;
Box2.length = 12.0;
Box2.breadth = 13.0;

// volume of box 1
volume = Box1.height * Box1.length * Box1.breadth;
cout << "Volume of Box1 : " << volume <<endl;

// volume of box 2
volume = Box2.height * Box2.length * Box2.breadth;
cout << "Volume of Box2 : " << volume <<endl;
return 0;
}
```

Defining Member Function Inside the Class

```
class Employee {
    int Id; char Name[25]; int Age; long Salary;
    public:
void GetData() //Statement 1 : Defining GetData()
{
    cout<<"\nEnter Employee Id : ";
    cin>>Id;
    cout<<"\nEnter Employee Name : ";
    cin>>Name;
    cout<<"\nEnter Employee Age : ";
    cin>>Age;
    cout<<"\nEnter Employee Salary : ";
    cin>>Salary;
}
```



```
void PutData()
{
    cout<<"\n Employee Id : "<<Id;
    cout<<"\n Employee Name : "<<Name;
    cout<<"\n Employee Age : "<<Age;
    cout<<"\n Employee Salary : "<<Salary;
}

};

int main()
{
    Employee E;
    E.GetData();
    E.PutData();
}
```

Defining Member Function Outside the Class

Defining Class Member Functions

- We can define member function inside or outside class.
- **Syntax of defining member function outside class**

```
return_type class_name :: function_name(argument
list)
    {
        -----
        body of function
        -----
    }
```

Defining Class Member Functions

- **return_type** Type of value function will return.
- **class_name::** A program may contain more than one class and these classes may have similar member functions. `Class_name::` tells the compiler which class the function belongs to and the scope of the member function is restricted to the `class_name`.
- **function_name** Can be any valid C++ identifier.
- **argument list** Represents the type and number of value function will take, values are sent by the calling statement.

```
class Employee {
    int Id; char Name[25]; int Age; float Salary;
public:
    void GetData();
    void PutData();
};

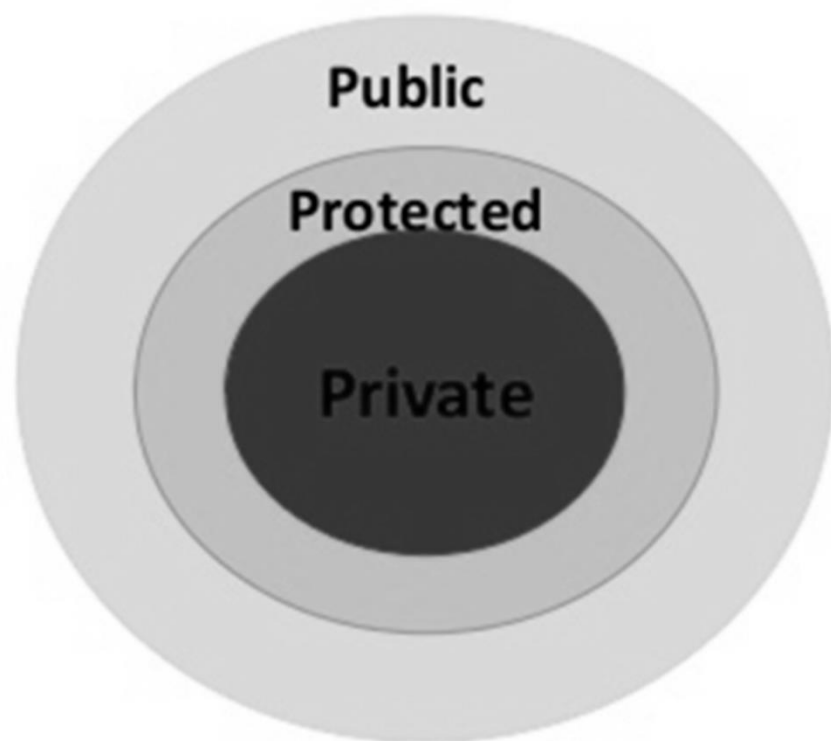
void Employee :: GetData() {
    cout<<"\nEnter Employee Id : ";
    cin>>Id;
    cout<<"\nEnter Employee Name : ";
    cin>>Name;
    cout<<"\nEnter Employee Age : ";
    cin>>Age;
    cout<<"\nEnter Employee Salary : ";
    cin>>Salary;
}
```

```
void Employee :: PutData()
{
    cout<<"\n Employee Id : "<<Id;
    cout<<"\n Employee Name : "<<Name;
    cout<<"\n Employee Age : "<<Age;
    cout<<"\n Employee Salary : "<<Salary; }
int main()
{

    Employee E;

    E.GetData();
    E.PutData();
}
```

Access Specifiers in C++



Access Specifiers in C++

- There can be three access specifiers in C++ :
 - **Private** : This access specifier makes sure that the private members can be accessed from other members of the class only or from the friends of class (this is an advanced concept, will learn it later).
 - **Protected** : This access specifier makes sure that the protected members can be accessed from members of the same class ,from the friend classes and also from members of the derived classes.
 - **Public** : This access specifier makes sure that the public members can be accessed from anywhere through the object of the class.

Access Specifiers in C++

- Private, Protected, Public are also called **visibility labels**.
- The member that declare **private** can be accessed only from within class.
- **Public** member can be accessed anywhere through the object of the class.
- In C++ data can be hidden by making private.

Access Specifiers in C++

