

27/8/2021

Name : Durvesh Naresh Patil
PRN : 2019BTEEN00035

Object Oriented Programming.

Exp. 3 : Scope Resolution

- ① Activity : Write a C++ code to illustrate use of scope resolution operator in accessing member functions outside a class.

```
#include <iostream>
using namespace std;
```

```
class Book
{
    int id;
    string name;
    int price;
public:
    void setData(int);
    void display();
    int getId();
    int getPrice();
};
```

```
void Book::setData(int a)
{
    id = a;
    cout << "Enter Book Name: ";
    cin >> name;
    cout << "Enter price: ";
    cin >> price;
}
```

```
void Book :: display()
{
    cout << "\n Book Data \n Id : " << id << "\n Name : "
    << name << "\n Price : " << price ;
}
```

```
int Book :: getId()
{
    return id ;
}
```

```
int Book :: getPrice()
{
    return price ;
}
```

```
int main()
{
    Book b1 ;
    b1.setData(121);
    b1.display();
    cout << b1.getPrice() << endl ;
    return 0 ;
}
```


○ Questions :

1. To define a function outside a class, which operator is used.

⇒ ::

2. Which operator is used to eliminate ambiguity in multiple inheritance.

⇒ ::

3. How can you access the class nested inside another class ?

⇒ We can access the nested class using the scope resolution operator (::).

Let's consider, the name of class which is nested is called 'Inner' & the name of class in which it is nested is 'Outer'. Now, we have to access the Inner class, we can access it using the following syntax.

Outer :: Inner obj;

4. Operator such as :: cannot be overloaded.

5. The scope-resolution operator usually

⇒ specifies a particular class.

① Conclusion : The scope resolution operator (::) is used for following purposes.

- (i) To define a function outside a class
- (ii) To eliminate ambiguity in multiple inheritance
- (iii) To access global variable when there is a local variable with same name.
- (iv) To access class's static variables.
- (v) To access a class inside another class i.e. nested class.

Scope resolution operator cannot be overloaded.