

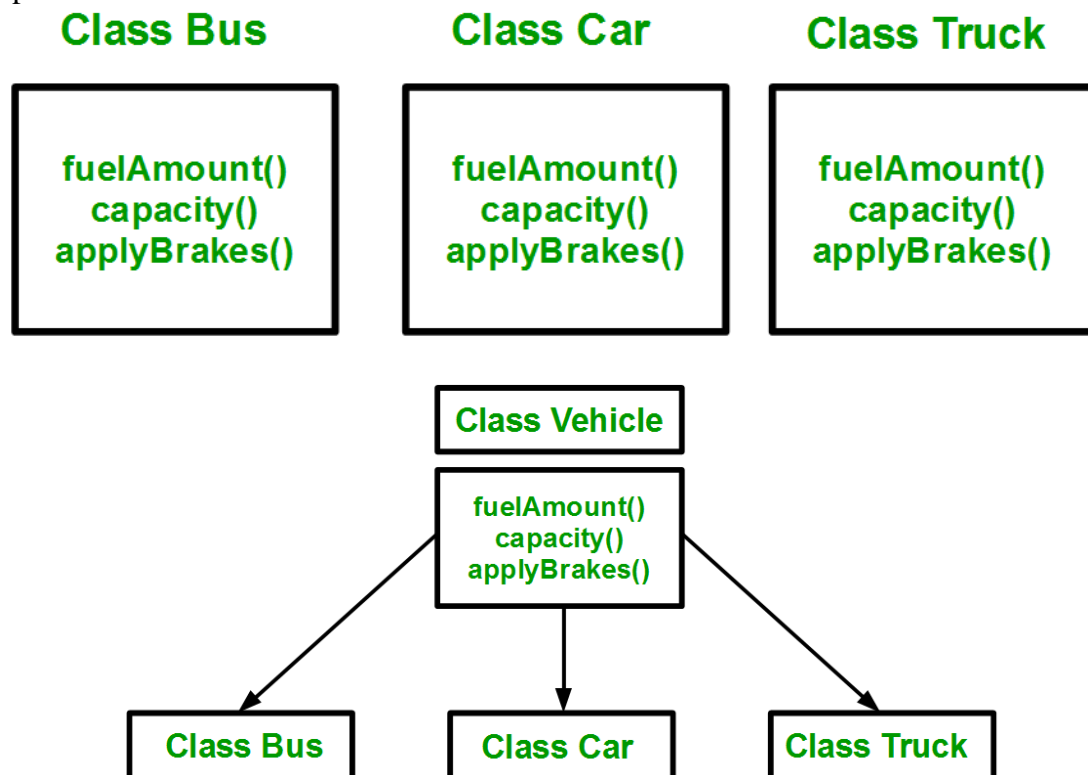


Inheritance: Derived Class and Base Class

Inheritance enables you to create new classes that reuse, extend, and modify the behavior defined in other classes. **The class whose members are inherited is called the base class, and the class that inherits those members is called the derived class.**

These new features in the derived class will not affect the base class. The derived class is the specialized class for the base class.

- **Sub Class:** The class that inherits properties from another class is called Subclass or Derived Class.
- **Super Class:** The class whose properties are inherited by a subclass is called Base Class or Superclass.



Derived Classes: A Derived class is defined as the class derived from the base class.

Syntax:

```
class <derived_class_name> : <access-specifier> <base_class_name>
{
    //body
}
```

Base Class: A base class is a class in Object-Oriented Programming language, from which other classes are derived. The class which inherits the base class has all members of a base class as well as can also have some additional properties. The Base class members and member functions are inherited to Object of the derived class. A base class is also called **parent class** or **superclass**.

Derived Class: A class that is created from an existing class. The derived class inherits all members and member functions of a base class. The derived class can have more functionality with respect to the



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Base class and can easily access the Base class. A Derived class is also called a **child class** or **subclass**.

Syntax for creating Derive Class:

```
class BaseClass{  
    // members....  
    // member function  
}
```

```
class DerivedClass : public BaseClass{  
    // members....  
    // member function  
}
```

Difference between Base Class and Derived Class:

S.No.	BASE CLASS	DERIVED CLASS
1.	A class from which properties are inherited.	A class from which is inherited from the base class.
2.	It is also known as parent class or superclass.	It is also known as child class subclass.
3.	It cannot inherit properties and methods of Derived Class.	It can inherit properties and methods of Base Class.
4.	Syntax: Class base_classname{ ... }.	Syntax: Class derived_classname : access_mode base_class_name { ... }.

Below is the program to illustrate **Base Class** and **Derived Class**:

```
// C++ program to illustrate  
// Base & Derived Class
```

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

```
// Declare Base Class  
class Base {  
public:  
    int a;  
};
```

```
// Declare Derived Class  
class Derived : public Base {  
public:  
    int b;  
};
```

```
// Driver Code  
int main()  
{  
    // Initialise a Derived class geeks  
    Derived geeks;
```



SNS COLLEGE OF TECHNOLOGY, COIMBATORE –35
(An Autonomous Institution)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

```
// Assign value to Derived class variable  
geeks.b = 3;
```

```
// Assign value to Base class variable  
// via derived class  
geeks.a = 4;
```

```
cout << "Value from derived class: "  
    << geeks.b << endl;
```

```
cout << "Value from base class: "  
    << geeks.a << endl;
```

```
return 0;
```

```
}
```

Output:

Value from derived class: 3

Value from base class: 4

/ Example: define member function without argument within the class

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

```
class Person
```

```
{
```

```
int id;  
char name[100];
```

```
public:
```

```
void set_p()
```

```
{
```

```
cout<<"Enter the Id:";  
cin>>id;  
fflush(stdin);  
cout<<"Enter the Name:";  
cin.get(name,100);
```

```
}
```

```
void display_p()
```

```
{
```

```
cout<<endl<<id<<"\t" <<name<<"\t";
```

```
}
```

```
};
```

```
class Student: private Person
```

```
{
```

```
char course[50];  
int fee;
```



SNS COLLEGE OF TECHNOLOGY, COIMBATORE –35
(An Autonomous Institution)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

```
public:
void set_s()
{
    set_p();
    cout<<"Enter the Course Name:";
    fflush(stdin);
    cin.getline(course,50);
    cout<<"Enter the Course Fee:";
    cin>>fee;
}

void display_s()
{
    display_p();
    cout<<course<<"\t"<<fee<<endl;
}
};

main()
{
    Student s;
    s.set_s();
    s.display_s();
    return 0;
}
```

Output:

```
Enter the Id: 101
Enter the Name: Dev
Enter the Course Name: GCS
Enter the Course Fee:70000
```

```
101   Dev   GCS   70000
```