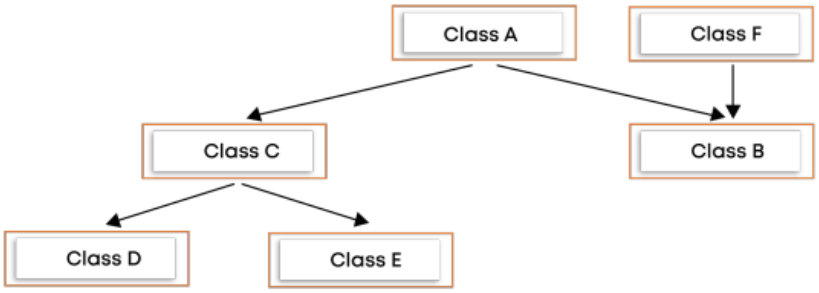
Hybrid Inheritance

Overview

Hybrid inheritance is a combination of more than one type of inheritance. For example, A child and parent class relationship that follows multiple and hierarchical inheritances can be called hybrid inheritance.



Hybrid Inheritance- (a combination of Hierarchical and multiple)

Syntax:

```
class parent_class1 {
    //Body of parent class1
};
class parent_class2 {
    //Body of parent class1
};
class child_class1: access_modifier parent_class1 {
    //Body of child class1
};
class child_class2: access_modifier parent_class1, access_modifier
parent_class2 {
    //Body of child class2
};
```

Example:

```
#include <iostream>

using namespace std;

// Parent class1
class Vehicle {
    public:
        Vehicle() {
            cout << "This is a Vehicle" << endl;
        }
};

//Parent class2
class Fare {
    public:
        Fare() {
            cout << "Fare of Vehicle\n";
        }
};

//Child class1
class Car: public Vehicle {

};

//Child class2
class Bus: public Vehicle, public Fare {

};

// main function
int main() {
    // creating object of sub class will
    // invoke the constructor of base class
    Bus obj2;
    return 0;
}
```

Output:

```
This is a Vehicle
Fare of Vehicle
```