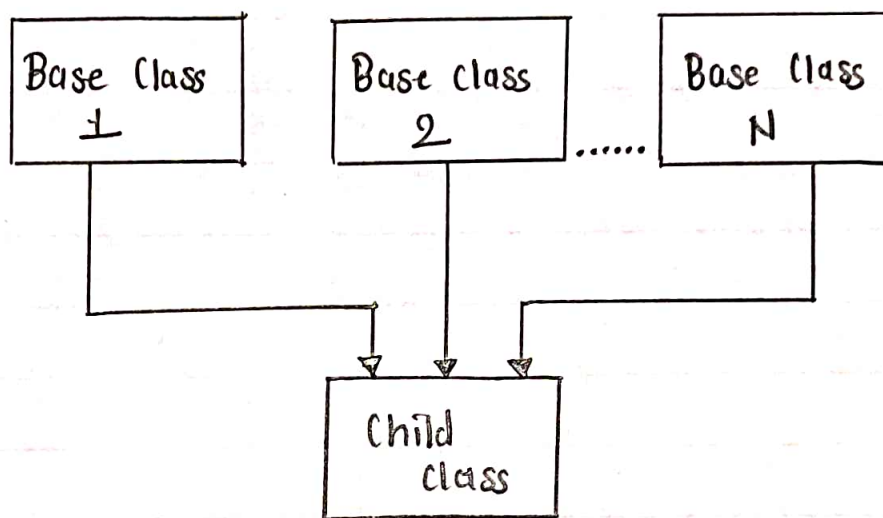


## Experiment No. 8

Date:

**Aim:** Write a C++ program to demonstrate the concept of multiple inheritance along with an example.

**Diagram:** General Representation of Multiple Inheritance:



Date:

**Aim:** Write a C++ program to demonstrate the concept of Multiple inheritance along with an example.

**Theory:** About inheritance: It is a key feature of Object oriented programming (OOP) that allows one class (called the derived class) to inherit attributes & methods from another class (called the base class). This promotes reusability & hierarchical classification.

Types of inheritance in C++:

\*) C++ supports 5 types of inheritance:

- A) Single Inheritance
- B) Multiple Inheritance
- C) Multilevel Inheritance
- D) Hierarchical Inheritance
- E) Hybrid Inheritance.

ARISE & SHINE

About Multiple Inheritance:

Multiple inheritance allows a class to form more than one base class. The derived class inherits the properties & behaviours (methods & variables) of all the base classes. This can be useful in modelling complex relationships, but it can also introduce complications such as the Diamond Problem.

Syntax of multiple inheritance :

```
class Base1 {
    //members of Base 1
};
```

```
class Base2 {
    //members of Base 2
};
```

// Derived class inherits from both Base 1 & Base 2 class.

```
class Derived : public Base1, public Base2 {
    //members of derived class.
};
```

Code: // code to illustrate the concept of multiple inheritance.

```
#include <iostream>
```

```
using namespace std;
```

```
class Engine { // Base class
    public:
```



```

Engine () {
    cout << "Engine is ready" << endl;
}

void start () {
    cout << "Engine started" << endl;
}
};

```

```

class Body { // Base Class 2
public:
    Body () {
        cout << "Car body is ready" << endl;
    }
    void paint () {
        cout << "Car body painted" << endl;
    }
};

```

```

// Derived class
class Car : public Engine, public Body {
public:
    Car () {
        cout << "Car is fully assembled" << endl;
    }
    void drive () {
        start ();
        paint ();
    }
};

```

Conclusion: Hence, according to the code & its correct output the concept of multiple inheritance is studied & executed successfully.

```

        cout << "Car is driving" << endl;
    }
};

```

```

int main () {
    Car myCar;
    myCar.drive ();

```

```

    return 0;
}

```

Output: Engine is ready  
 Car body is ready  
 Car is fully assembled  
 Engine started  
 Car body painted  
 Car is driving.

Conclusion: Hence, according to the code & it's corr. correct output the concept of multiple inheritance is studied & executed successfully..