



Prifysgol Cymru  
Y Drindod Dewi Sant  
University of Wales  
Trinity Saint David

200  
1822 - 2022

# Inheritance

*Man Geni Addysg Uwch yng Nghymru*  
*Where Higher Education Began in Wales*

[www.uwtsd.ac.uk](http://www.uwtsd.ac.uk)



# Contents

- Inheritance
- Modes of inheritance
- Types of inheritance

# Inheritance- reuse of code

- Base class and Derived Classes

- The new class created is called “derived class” or “child class”
- the existing class is known as the “base class” or “parent class”
-

## Class Bus

`fuelAmount()`  
`capacity()`  
`applyBrakes()`

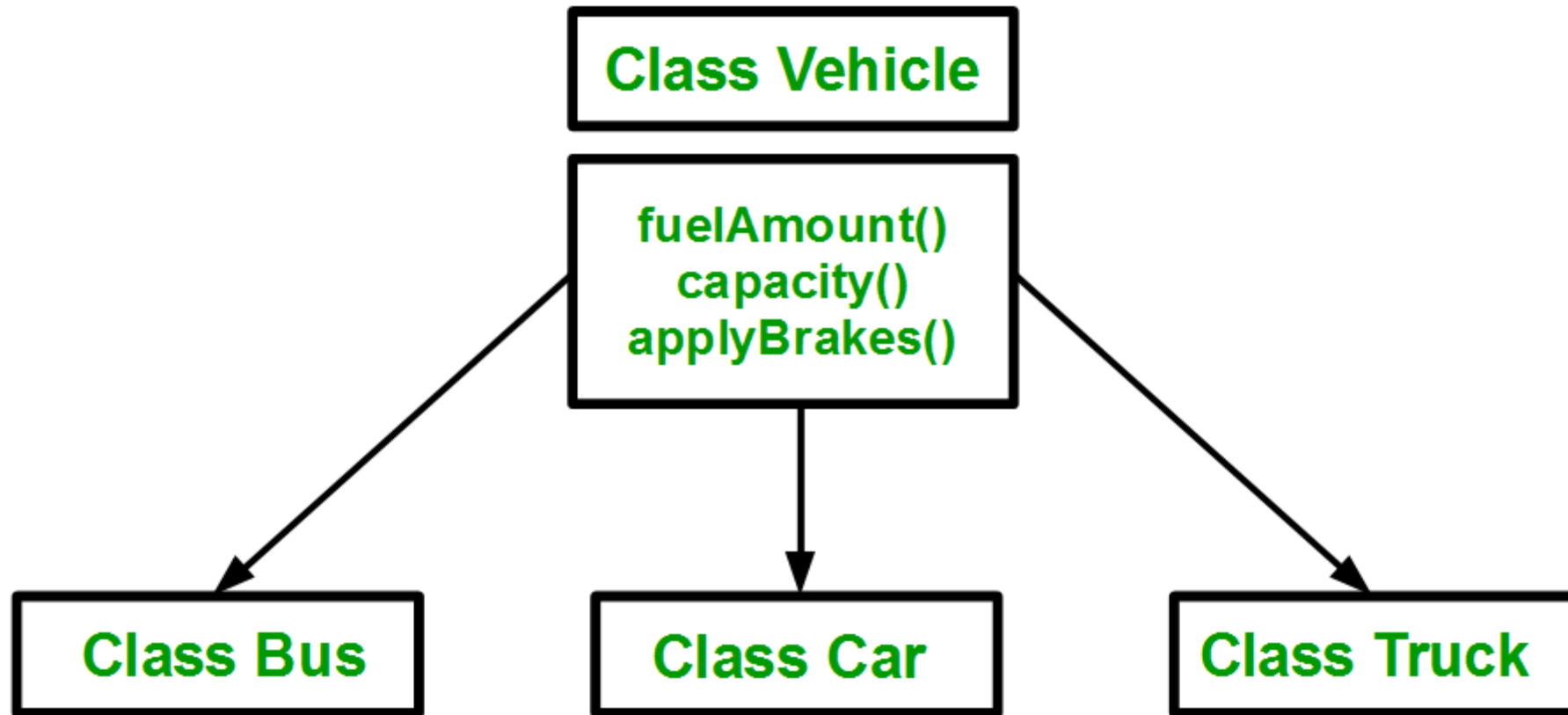
## Class Car

`fuelAmount()`  
`capacity()`  
`applyBrakes()`

## Class Truck

`fuelAmount()`  
`capacity()`  
`applyBrakes()`





# Implementing inheritance

Syntax:

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

# Example- C++ program to demonstrate implementation of Inheritance

// Base class

```
class Parent {  
public:  
    int id_p;  
};
```

// Sub class inheriting from Base Class(Parent)

class Child : public Parent

```
{  
public:  
    int id_c;  
};
```

// main function

```
int main()  
{
```

```
    Child obj1;
```

```
    // An object of class child has all data members  
    // and member functions of class parent
```

```
    obj1.id_c = 7;
```

```
    obj1.id_p = 91;
```

```
    cout << "Child id is: " << obj1.id_c << "\n";
```

```
    cout << "Parent id is: " << obj1.id_p << "\n";
```

```
    return 0;
```

```
}
```

**Output**

```
Child id is: 7
```

```
Parent id is: 91
```

# Modes of inheritance

- **Public Mode:**

Subclass derived from public base class-then public members of base class are public members in derived class

- **Protected Mode:**

Subclass derived from protected base class – both public & protected members of base class are protected in derived class

- **Private Mode:**

Subclass derived from private base class- both public & protected members of base class are private in derived class

C++ Implementation to show that a derived class doesn't inherit access to private data members.

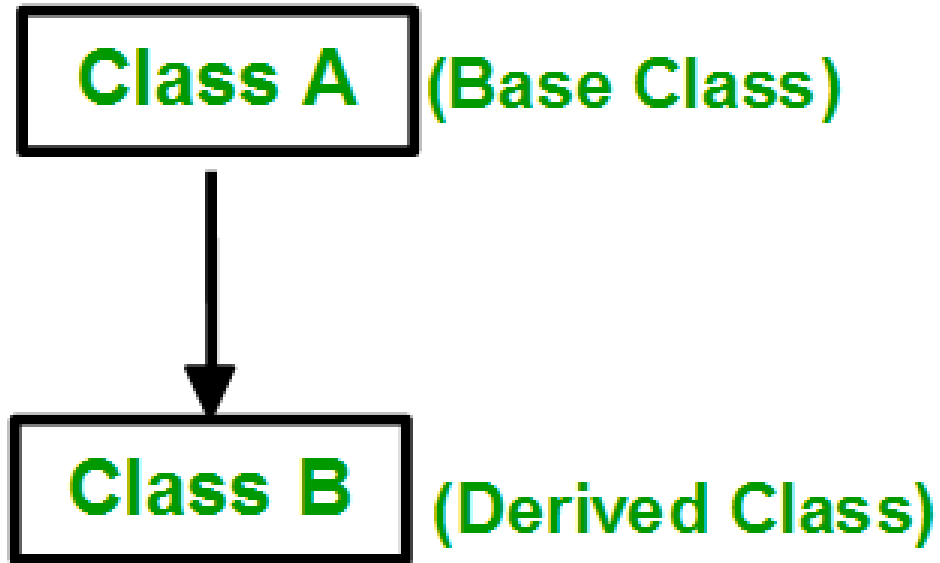
```
class A {  
public:  
    int x;  
  
protected:  
    int y;  
  
private:  
    int z;  
};  
  
class B : public A {  
    // x is public  
    // y is protected  
    // z is not accessible from B
```

```
class C : protected A {  
    // x is protected  
    // y is protected  
    // z is not accessible from C  
};  
  
class D : private A // 'private' is default  
for classes  
{  
    // x is private  
    // y is private  
    // z is not accessible from D  
};
```



# Types of Inheritance in C++

- **Single Inheritance**



# Syntax:

```
class subclass_name : access_mode base_class  
{  
    // body of subclass  
};
```

OR

```
class A { ... .. };  
class B: public A { ... .. };
```

# Explaining Single inheritance

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

```
// base class
```

```
class Vehicle {
public:
    Vehicle()
    {
        cout << "This is a Vehicle\n";
    }
};
```

```
// sub class derived from a single base classes
```

```
class Car : public Vehicle {
```

```
};
```

```
// main function
```

```
int main()
{
    Car obj;
    return 0;
}
```



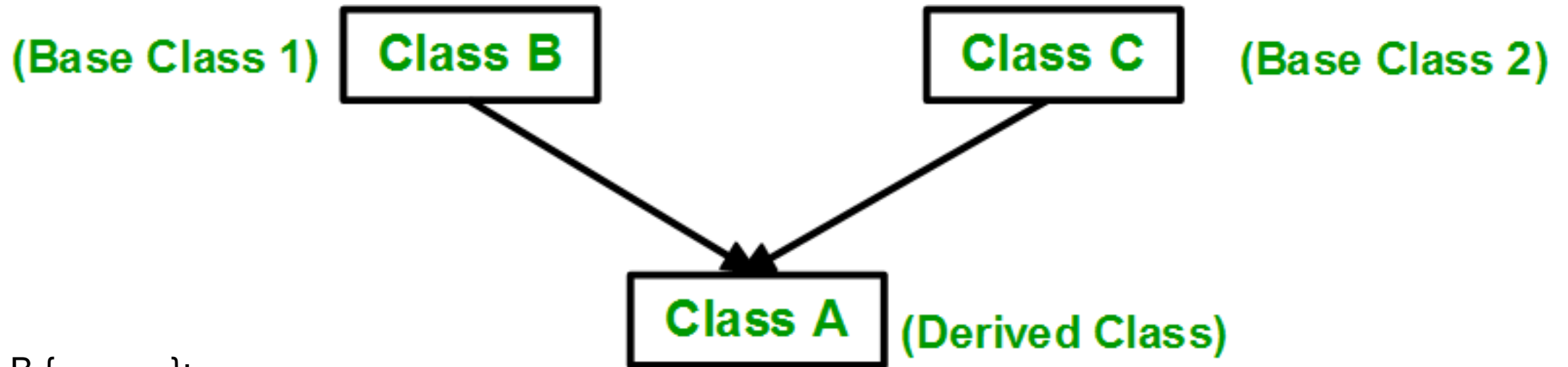
# Exercise

- Write a code for finding the product of 2 numbers using single inheritance

Note: get first data in base class and second data in derived class

Perform the operation in derived class using member function

# Multiple Inheritance



```
class B { ... .. };  
class C { ... .. };
```

```
class A: public B, public C { ... .. };
```

# // C++ program to explain multiple inheritance

```
class A
{
    public:
    int x;
    void getx()
    {
        cout << "enter value of x: "; cin >> x;
    }
};

class B
{
    public:
    int y;
    void gety()
    {
        cout << "enter value of y: "; cin >> y;
    }
};
```

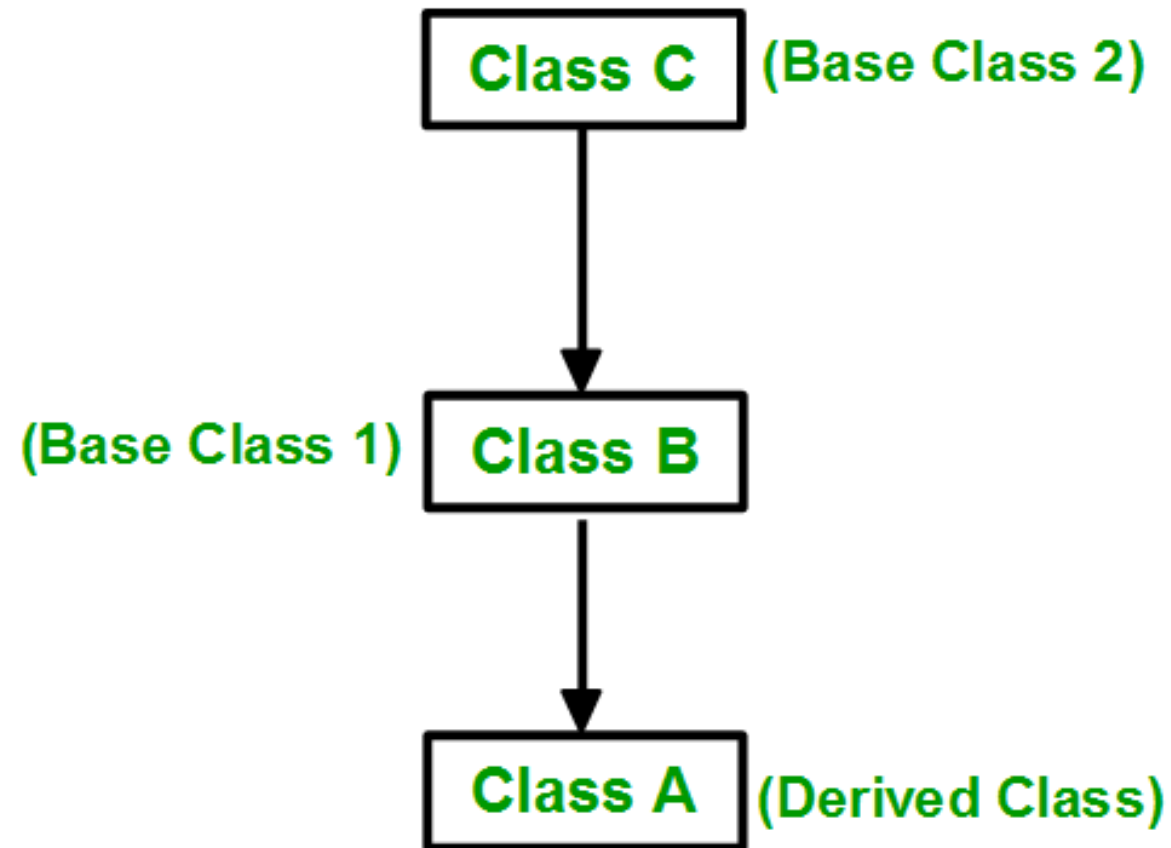
```
class C : public A, public B
{
    public:
    void sum()
    {
        cout << "Sum = " << x + y;
    }
};

int main()
{
    C obj1;
    obj1.getx();
    obj1.gety();
    obj1.sum();
    return 0;
}
```

Output:

```
enter value of x: 5
enter value of y: 4
Sum = 9
```

# Multilevel Inheritance



# syntax

```
class A // base class
```

```
{
```

```
.....
```

```
};
```

```
class B : access_specifier A // derived class
```

```
{
```

```
.....
```

```
};
```

```
class C : access_specifier B // derived from derived class B
```

```
{
```

```
.....
```

```
};
```



```
class base //single base class
```

```
{  
    public:  
    int x;  
    void getdata()  
    {  
        cout << "Enter value of x= ";  
        cin >> x;  
    }  
};
```

```
class derive1 : public base
```

```
{  
    public:  
    int y;  
    void readdata()  
    {  
        cout << "\nEnter value of y= ";  
        cin >> y;  
    }  
};
```

```
class derive2 : public derive1
```

```
{  
    private:  
    int z;  
    public:  
    void indata()  
    {  
        cout << "\nEnter value of z= ";  
        cin >> z;  
    }  
    void product()  
    {  
        cout << "\nProduct= " << x * y * z;  
    } };
```

```
int main()
```

```
{  
    derive2 a;    //object of derived class  
    a.getdata();  
    a.readdata();  
    a.indata();  
    a.product();  
    return 0;  
}
```

Output:

```
Enter value of x= 2  
Enter value of y= 3  
Enter value of z= 3  
Product= 18
```



# How does multiple inheritance differ from multilevel inheritance?

# How does multiple inheritance differ from multilevel inheritance?

- **Multilevel inheritance** : Inheritance of characters by a child from father and father inheriting characters from his father (grandfather)
- **Multiple inheritance** : Inheritance of characters by a child from mother and father

End of session