**C++ virtual function**

C++ virtual function is a member function in base class that you redefine in a derived class. It is declare using the virtual keyword.

It is used to tell the compiler to perform **dynamic linkage or late binding** on the function.

Late binding or Dynamic linkage

In late binding function call is resolved during runtime. Therefore compiler determines the type of object at runtime, and then binds the function call.

|  |  |
| --- | --- |
| **Using virtual** | **Not Using virtual** |
| #include <iostream>  using namespace std;  class A {  public:  virtual void display(){  cout << "Base class is invoked"<<endl;  }  };  class B:public A {  public:  void display() {  cout << "Derived Class is invoked"<<endl;  }  };  int main() {  A\* a; //pointer of base class  B b; //object of derived class  a = &b;  a->display(); //Late Binding occurs  }  //Derived Class is invoked | #include <iostream>  using namespace std;  class A {  public:  void display(){  cout << "Base class is invoked"<<endl;  }  };  class B:public A {  public:  void display() {  cout << "Derived Class is invoked"<<endl;  }  };  int main() {  A\* a; //pointer of base class  B b; //object of derived class  a = &b;  a->display(); //Late Binding occurs  }  //Base class is invoked |