



Pointers to Derived Classes in C++ | C++ Tutorials for Beginners #55



[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

Pointers to Derived Classes in C++ | C++ Tutorials for Beginners #55

In this tutorial, we will discuss pointer to derived class in C++

Pointer to Derived Class in C++

In C++ we are provided with the functionality to point the pointer to derived class or base class. An example program is shown below to demonstrate the concept of pointer to a derived class in C++

```
#include<iostream>
using namespace std;
class BaseClass{
public:
```

```
    int var_base;
    void display(){
        cout<<"Dispalying Base class variable var_base "<<var_base<<endl;
    }
};

class DerivedClass : public BaseClass{
    public:
        int var_derived;
        void display(){
            cout<<"Dispalying Base class variable var_base "<<var_base<<endl;
            cout<<"Dispalying Derived class variable var_derived "<<var_derived<<endl;
        }
};
```

Code Snippet 1: Pointer to Derived Class Program Example

As shown in Code snippet 1,

1. We created a class “BaseClass” which contains public data member “var_base” and member function “display”. The member function “display” will print the value of data member “var_base”
2. We created another class “DerivedClass” which is inheriting “BaseClass” and contains data member “var_derived” and member function “display”. The member function “display” will print the values of data members “var_base” and “var_derived”

The code for the main program is shown below,

```
int main(){
```

```
BaseClass * base_class_pointer;
BaseClass obj_base;
DerivedClass obj_derived;
base_class_pointer = &obj_derived; // Pointing base class pointer to derived class

base_class_pointer->var_base = 34;
// base_class_pointer->var_derived= 134; // Will throw an error
base_class_pointer->display();

base_class_pointer->var_base = 3400;
base_class_pointer->display();

DerivedClass * derived_class_pointer;
derived_class_pointer = &obj_derived;
derived_class_pointer->var_base = 9448;
derived_class_pointer->var_derived = 98;
derived_class_pointer->display();

return 0;
}
```

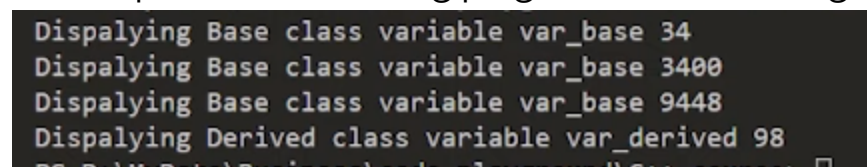
Code Snippet 2: Main Program

As shown in code snippet 2,

1. We created a pointer “base_class_pointer” of the data type “Baseclass”.
2. Object “obj_base” of the data type “BaseClass” is created.

3. Object "obj_derived" of the data type "DerivedClass" is created
4. Pointer "base_class_pointer" of the base class is pointing to the object "obj_derived" of the derived class
5. By using the pointer "base_class_pointer" of the base class we have set the value of the data member "var_base" by "34". The main thing to note here is that we cannot set the value of the derived class data member by using the base class pointer otherwise the compiler will throw an error.
6. The function "display" is called using a base class pointer. The main thing to note here is that the base class "display" function will run here.
7. Again by using the pointer "base_class_pointer" of the base class we have set the value of the data member "var_base" by "3400" which will update the previous value and the function "display" is called.
8. We created a pointer "derived_class_pointer" of the data type "DerivedClass"
9. Pointer "Derived_class_pointer" of the derived class is pointing to the object "obj_derived" of the derived class
10. By using pointer "Derived_class_pointer" of the derived class we have set the value of the data member "var_base" of the base class by "9448". The main thing to note here is that this will not throw an error because we can set the value of base class data member by using derived class pointer but we cannot set the value of derived class data member by using base class pointer
11. By using pointer "Derived_class_pointer" of the derived class we have set the value of the data member "var_derived" of the derived class by "98".
12. The function "display" is called using a derived class pointer. The main thing to note here is that the derived class "display" function will run here.

The output of the following program is shown in figure 1,



```
Dispalying Base class variable var_base 34
Dispalying Base class variable var_base 3400
Dispalying Base class variable var_base 9448
Dispalying Derived class variable var_derived 98
```

Figure 1: Program Output

[Previous](#)

[Next](#)



CodeWithHarry

Copyright © 2022 CodeWithHarry.com

