


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

class Base
{
public:
    Base() { cout << "Construct Base object\n"; }
    ~Base() { cout << "Destruct Base object\n"; }

protected:
    virtual void func() const { cout << "Function func() of class Base\n"; }
};

class Child: public Base
{
private
    Child() { cout << "Construct Child object\n"; }
    ~Child() { cout << "Destruct Child object\n"; }

protected:
    void func() const { cout << "Function func() of class Child\n"; }

public:
    Child * CreateChild() { return new Child(); }
};

void Function(const Base &obj)
{
    cout << "** Call method func() for object obj" << endl;
    obj.func();
}

int main(int argc, char *argv[])
{
    cout << "** Create Object base" << endl;
    Base base;

    cout << "** Create Object child" << endl;
    Child child;

    cout << "** Call method func() for object base" << endl;
    Function(base);

    cout << "** Call method func() for object child" << endl;
    Function(child);

    return 0;
}
```