


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

class Unit
{
    int life;
public:
    Unit(): life(200) { cout << "Construct Unit object\n"; }
    ~Unit() { cout << "Destruct Unit object\n"; }

public:
    virtual int status() const { cout << "Get status function Unit\n"; return life; }
    virtual int operator = (int newlife) { cout << "Set status function Unit\n"; return life = newlife; }
};

class Base
{
protected:
    Unit u;

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

    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"; }

public:
    void status() const { cout << "Current status" << u.status() << "\n"; }

    static 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 = CreateChild();

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

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

    return 0;
}
```