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

class Base{
private:
    int t;
public:
    Base() {
        cout << "Base constructor-0 param" << endl;
        t=0;
    }
    Base(int u) {
        cout << "Base constructor-1 param" << endl;
        t=u;
    }
    Base(const Base& b){
        t=b.t;
        cout << "Base copy constructor" << endl;
    }
    ~Base(){
        cout<<"Base destructor" << endl;
    }
};
```

```cpp
class DerivedA: public Base {
public:
    DerivedA(){
        cout<<"DerivedA constructor-0 param"<<endl;
        a=0;
    }
    DerivedA(int x) {
        cout<<"DerivedA constructor-1 param"<<endl;
        a = x;
    }
    DerivedA(int u, int x) {
        cout<<"DerivedA constructor-2 params"<<endl;
        a = x;
    }
    DerivedA(const DerivedA& da){
        a=da.a;
    }
    ~DerivedA(){
        cout<<"DerivedA destructor" << endl;
    }
    void print() {
        cout << "DerivedA = " << a << endl;
    }
private:
    int a;
};
```

```cpp
class DerivedB: public Base {
public:
    DerivedB() {
        cout << "DerivedB constructor-0 param" << endl;
        b = 0;
    }
    DerivedB(float x) {
        cout << "DerivedB constructor-1 param" << endl;
        b = x;
    }
    DerivedB(int u, float x):Base(u) {
        cout << "DerivedB constructor-2 params" << endl;
        b = x;
    }
    DerivedB(const DerivedB& db):Base(db) {
        cout << "DerivedB copy constructor" << endl;
        b=db.b;
    }
    ~DerivedB(){
        cout<<"DerivedB destructor" << endl;
    }
    void print() {
        cout << "DerivedB = " << b << endl;
    }
private:
    float b;
};
```

```cpp
void f(Base  b){
    cout<<"Function f" << endl;
}
```

```cpp
int main(){
    Base b1;              // Khởi tạo đối tượng Base b1, b1.t=0
    Base b2(3);           // khởi tạo đối tượng Base b2, với b2.t=3
    Base*pb=new Base(1);  // Tạo con trỏ pb kiểu Base với giá trị t=1

    DerivedA da1;         // Khởi tạo biến da1 kiểu DerivedA, với da1.a=0
    DerivedA da2(2);      // Khởi tạo biến da2 kiểu DerivedA với da2.a=2
    DerivedA da3(2,3);    // Khởi tạo biến da3 kiểu DerivedA với da3.t=0, da3.a=3
    DerivedA da4(da2);    // Khởi tạo biến da4 kiểu DerivedA với da4.t=da2.t=0; da4.a=da2.a=2
    pb=new DerivedA(4,5); // Tạo con trỏ pb kiểu DerivedA với giá trị t=4,a=5

    DerivedB db1;         // Khởi tạo biến db1 kiểu DerivedB, với t=0, b=0
    DerivedB db2(2.7);    // Khởi tạo biến db2 kiểu DerivedB, với t=0, b=2.7
    DerivedB db3(3,2.7);  // Khởi tạo biến db3 kiểu DerivedB, với t=3, b=2.7
    DerivedB db4(db2);    // Khởi tạo biến db4 kiểu DerivedB, với t=0, b=2.7
    pb=new DerivedB(4,5); // Tạo con trỏ pb kiểu DerivedB với giá trị t=4,a=5

    f(b1);
    f(da1);
    f(db1);
    delete pb;
    return 0;
}
```

Output annotations:

```
Base constructor-0 param
Base constructor-1 param
Base constructor-1 param

Base constructor-0 param
DerivedA constructor-0 param
Base constructor-0 param
DerivedA constructor-1 param
Base constructor-0 param
DerivedA constructor-2 params
Base constructor-0 param

Base constructor-0 param
DerivedA constructor-2 params

Base constructor-0 param
DerivedB constructor-0 param
Base constructor-0 param
DerivedB constructor-1 param
Base constructor-1 param
DerivedB constructor-2 params
Base copy constructor
DerivedB copy constructor
Base constructor-1 param
DerivedB constructor-2 params
```

```
Base copy constructor
Function f
Base destructor

Base copy constructor
Function f
Base destructor

Base copy constructor
Function f
Base destructor
```

Base destructor