Difference between property of class and property of initialization list of class in C++?

[intro]

Before answer this question, we must know the difference of shallow copy and deep copy (especially used in Python).

1. Shallow copy: Does NOT copy the object . Just bind the object to another object. Behaves as a pointer in C.

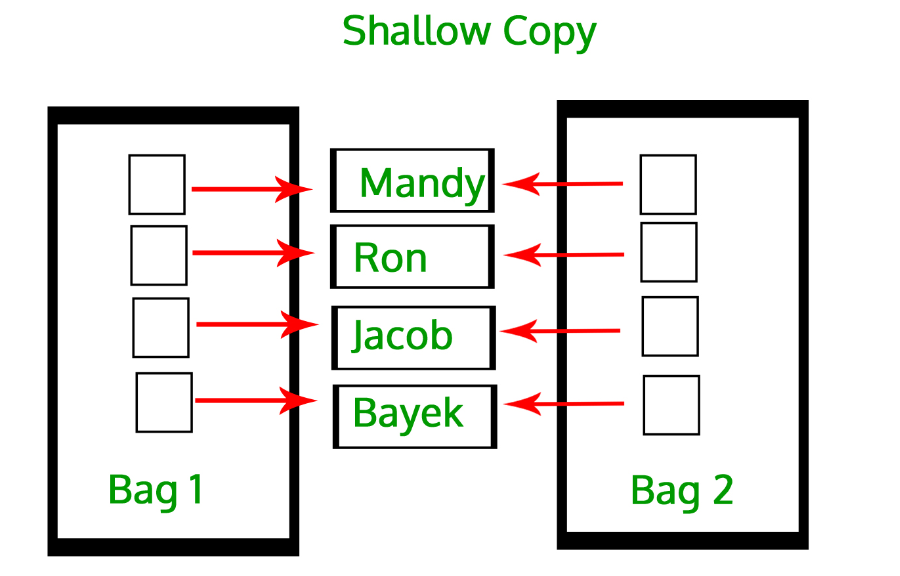
When the value which is shallowed copy is changed, the original value will also be changed.

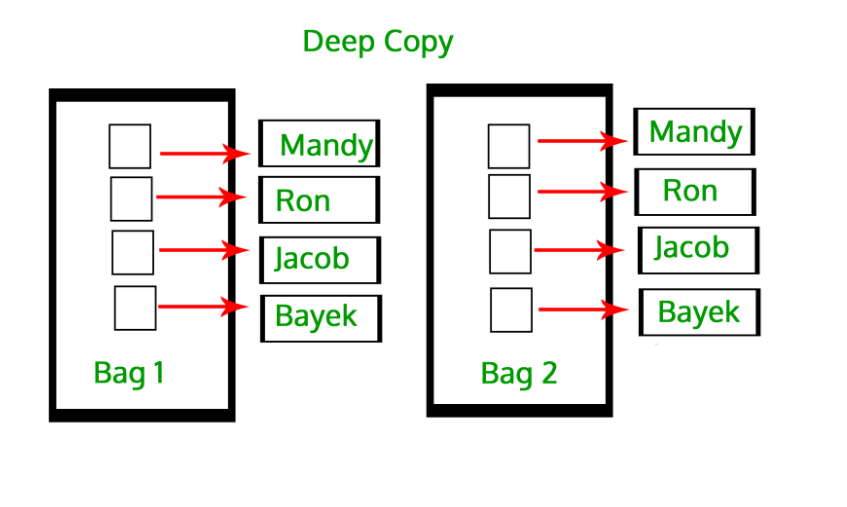
1. Deep copy: Does copy the object to another object. Behaves as a memory address movement.

In deep copy, the value of the memory address are loaded, then copy into the register,

finally the register will be loaded into the memory address of destination.

[figure]





[Example]

Consider the following situation.

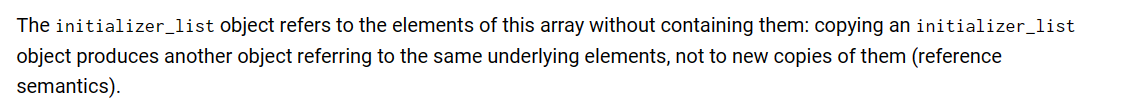
(More details on the following link)

[intro]

After discuss about shallow copy and deep copy,

let’s talk about the difference between property of class and property of initialization list.

First, let’s look at the introduction in Cplusplus.



I guess the meaning of this is as follows.

In C, the property which is NOT in initializer list will be deep copied when its class is assigned to another class. (including the statement)

While the property which is in initializer list will be shallow copied when its class is assigned to another class.

[declaration]

(1)How to declare an initializer list in C?

Declare with a template “initializer\_list<>” bracket.

[invoke]

1. How to call an initializer list in C?

Declare with a curly bracket.

struct myclass {

myclass (int,int);

myclass (initializer\_list<int>);

/\* definitions ... \*/

};

myclass foo {10,20}; // calls initializer\_list ctor

myclass bar (10,20); // calls first constructor

[ref]

[Difference between shallow copy and deep copy -- Geeksforgeeks](https://www.geeksforgeeks.org/copy-python-deep-copy-shallow-copy/)

<https://cplusplus.com/reference/initializer_list/initializer_list/>