

Types of Object Copy

One of the most common procedures that occurs in computer programs is the copying of data. An object is a composite data type in object-oriented programming languages. Object copy thus describes the action wherein an object has its attributes copied to another object of the same data type. An object may be copied in order to reuse all or part of its data in a new context.

Shallow Copy

A shallow copy of an object copies all of the member field values. This works well if the fields are values, but may not be what you want for fields that point to dynamically allocated memory. The pointer will be copied, but the memory it points to will not be copied -- the field in both the original object and the copy will then point to the same dynamically allocated memory, which is not usually what you want. The default copy constructor and assignment operator make shallow copies.

Example

Code: JavaScript

```
var o = { name: 'Go4expert', version: 2007, authors: ['shabbir', 'pradeep']
};

var o2 = Object.clone(o);

o2.version = '2007 amazing';
o2.authors.pop();

o.version
// -> 1.5

o2.version
// -> '2007 amazing'

o.authors
// -> ['shabbir'] // Shallow copy!
```

Deep Copy

A deep copy copies all fields, and makes copies of dynamically allocated memory pointed to by the fields. To make a deep copy, you must write a copy constructor and overload the assignment operator, otherwise the copy will point to the original, with disastrous consequences.

Example

Code: CPP

```
class A
{
    string s;
};

A a;
A b;
a=b; //deep copy
```

Lazy Copy

A lazy copy is a combination of both strategies above. When initially copying an object, a (fast) shallow copy is used. A counter is also used to track how many objects share the data. When the program wants to modify an object, it can determine if the data is shared (by examining the counter) and can do a deep copy if necessary.

Lazy copy looks to the outside just as a deep copy but takes advantage of the speed of a shallow copy whenever possible. The downside are rather high but constant base costs because of the counter. Also in certain situation circular references can also cause problems.

Source: <http://www.go4expert.com/articles/types-object-copy-t6391/>