Subject: Re: Transfer Semantics - Initializer list and copy constructors
Posted by forlano on Sat, 29 Sep 2007 22:27:31 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 10 July 2007 22:11The simple rule: Do what you need. If you get crash because of broken pick semantics (the ASSERT in Vector with items = -1), use "deep copy variant" (<<= instead of =, constructor with additional int parameter (e.g. 1).

Hello,

for first time I needed to copy the content of a vector in another vector WITHOUT to destroy the original vector. This is my class:

```
class Person : Moveable<Person> {
 int  currentProposer;
 int  cursor;
  public:
 Index<int> preference;
 int  RankProposer(int k);
        int  GetNext();      int  GetCurrentRank();     int  GetCurrentProposer();
        int  GetProposedTo();
 int  FindPosition(int k);
 void AcceptProposal(int k);
 String ToString() const { String s; for (int i=0; i<preference.GetCount();i++)
s<<AsString(preference[i]) + ' '; return s; }
 Person() {currentProposer = 0; cursor = -1; }
};
```

Then I declared two vectors:

```
Vector<Person> M, R;
```

After some operations made with M I would like to copy M in R. I used
R<<=M;
But the program crashed. Which is the correct syntax for this task?
Thanks,
Luigi