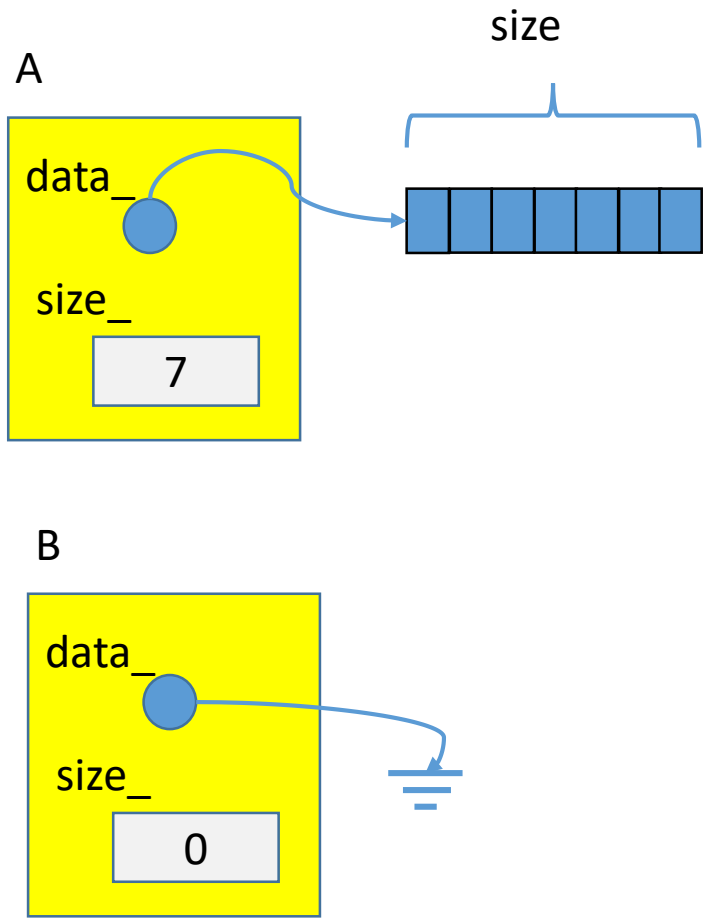
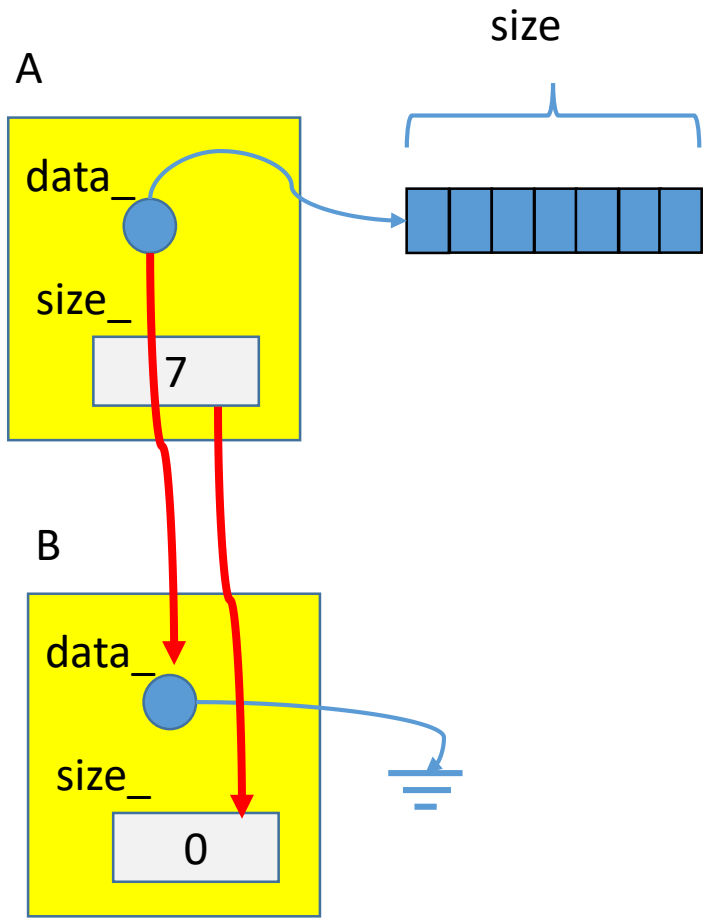
```
DataClass  A(7);
DataClass  B;
```

Bad Copying



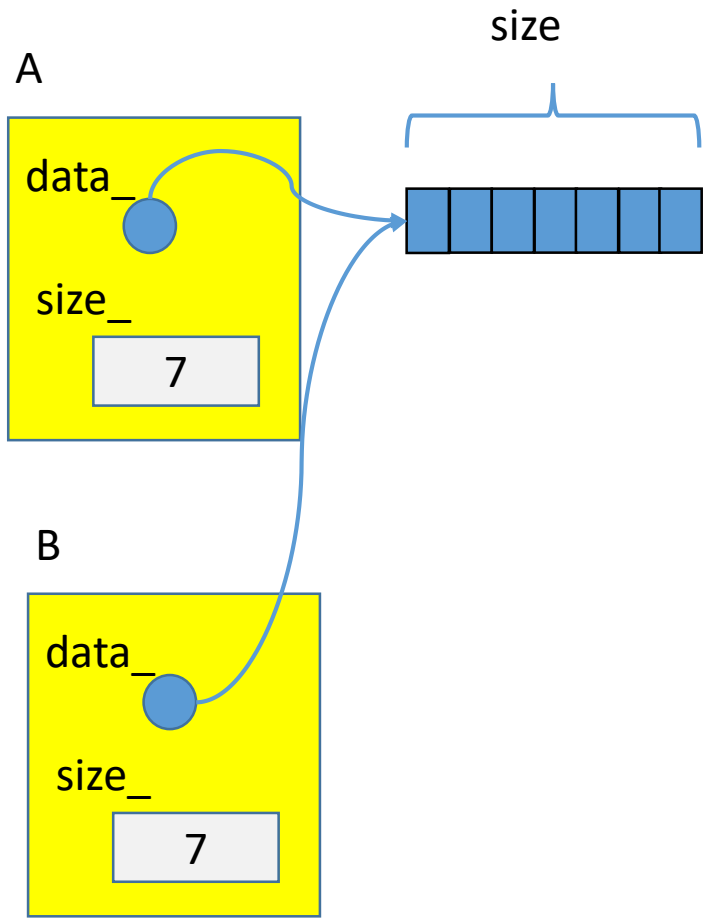
Bad Copying

B = A;
Or
DataClass B = A
Or
DataClass B(A);



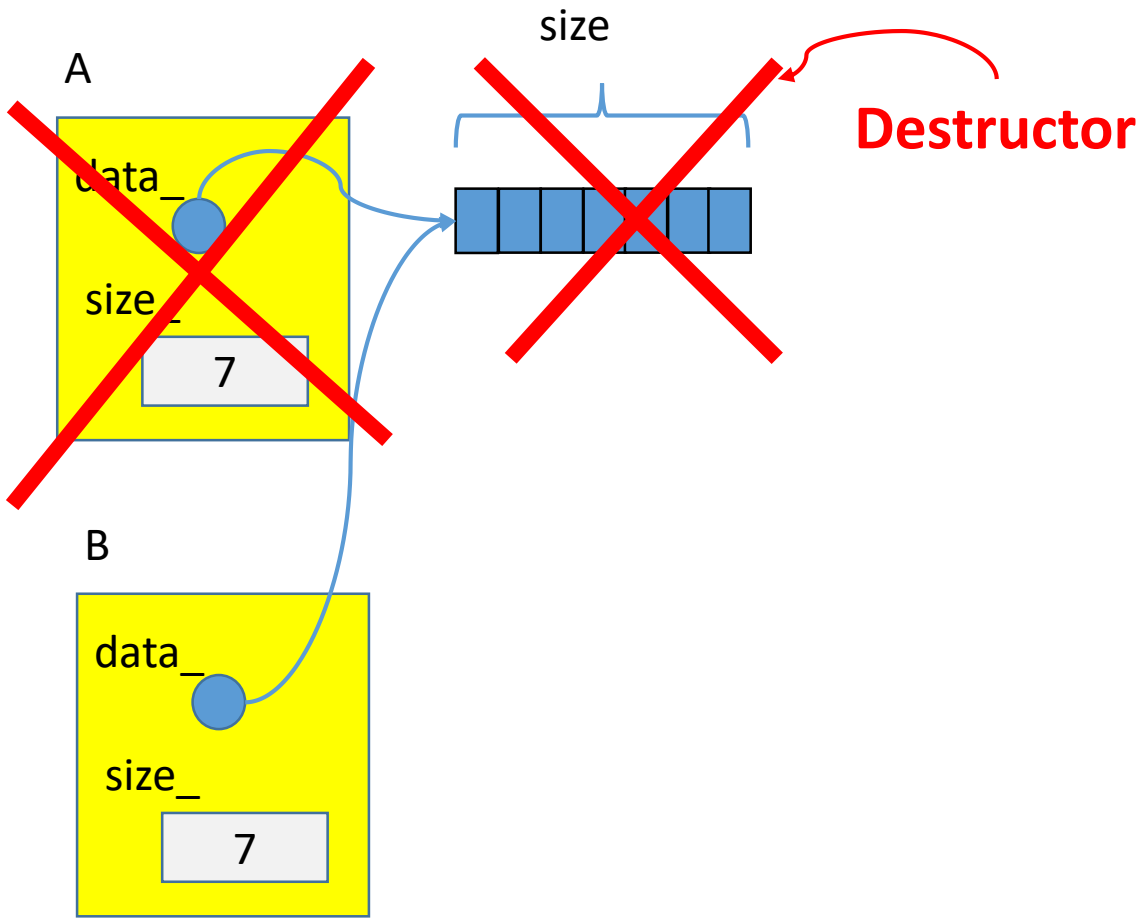
B = A;

Bad Copying

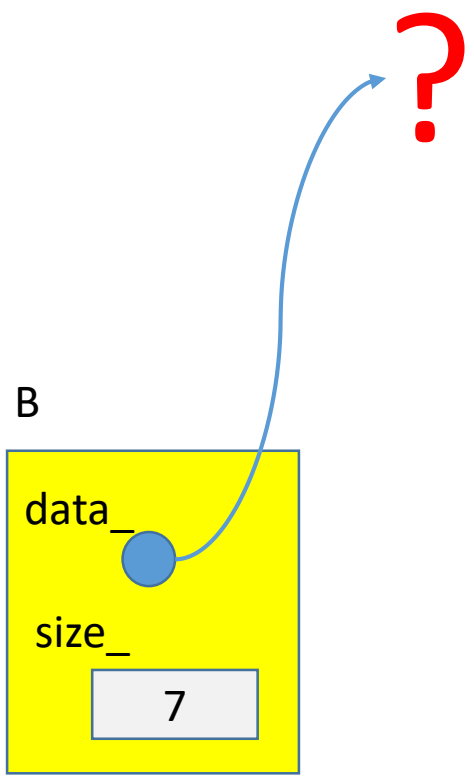


A goes out of scope

Bad Copying

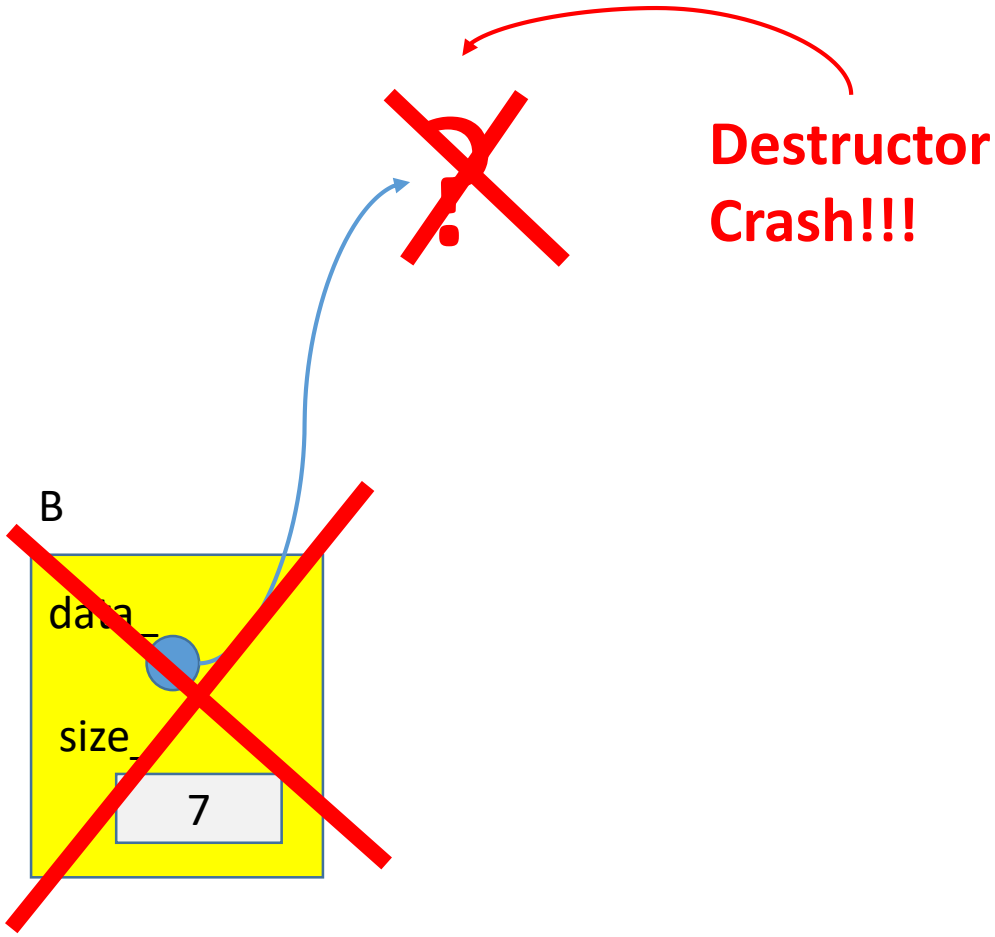


Bad Copying



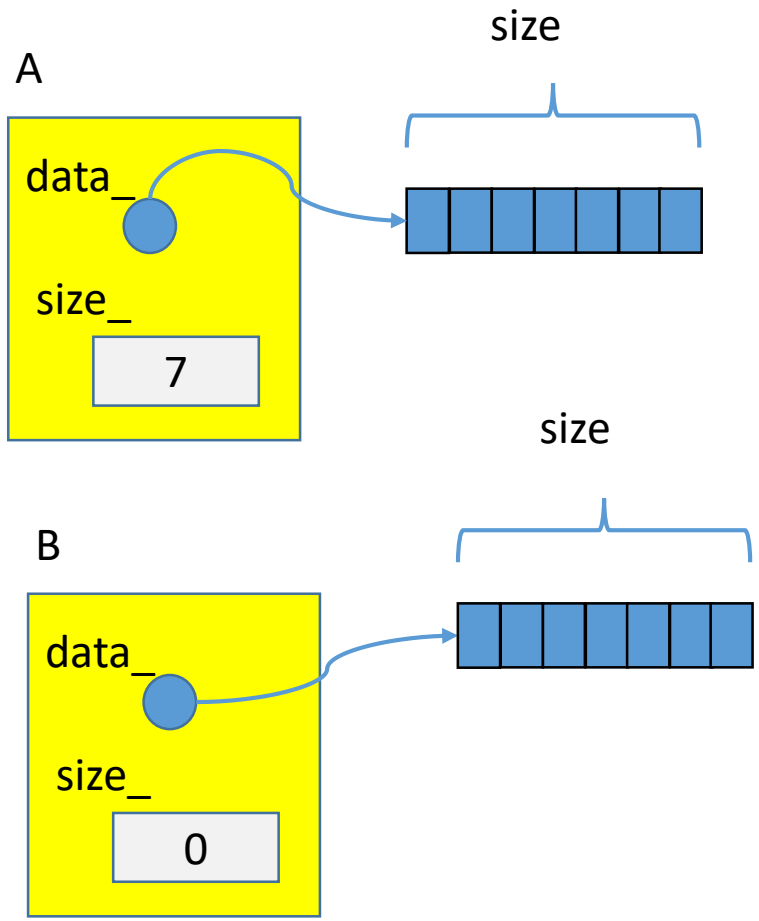
B goes out of scope

Bad Copying



Good Copying

B = A
Or
DataClass B = A
Or
DataClass B(A);

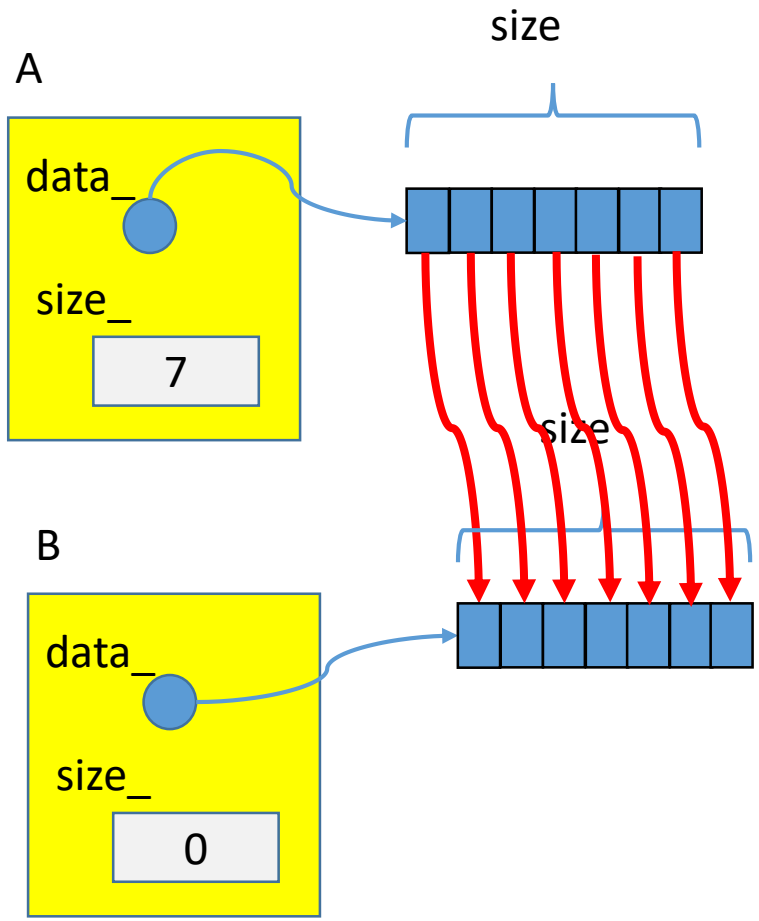


Copy Constructor or operator=()
Will allocate new memory:

data_ = new type[A.size_]

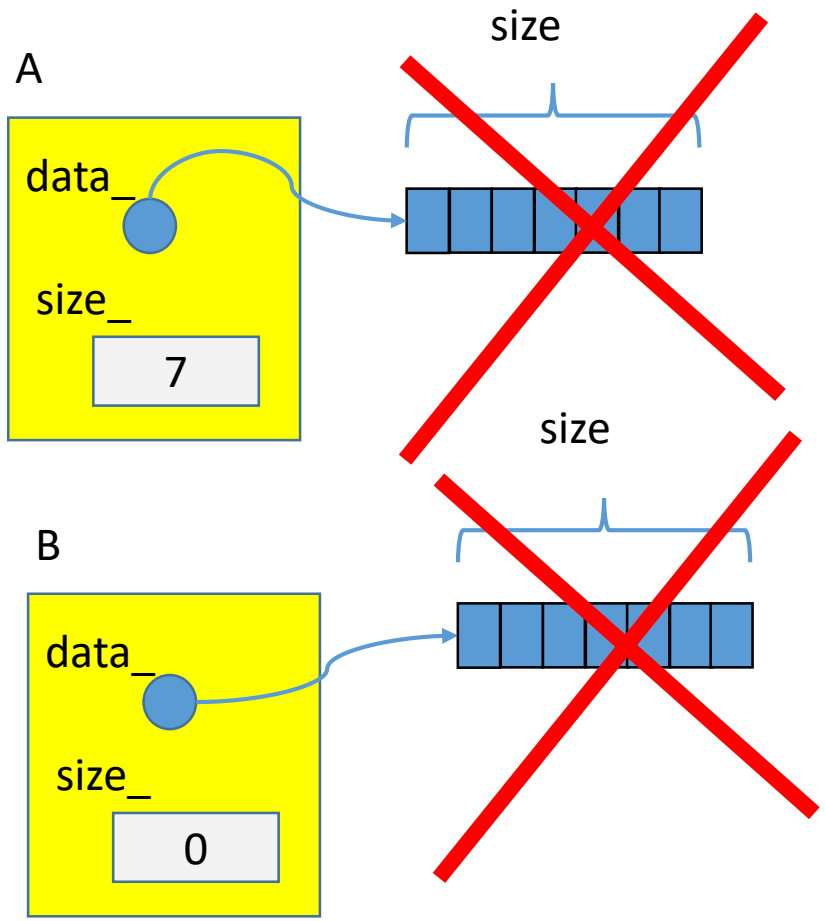
DataClass B = A
Or
DataClass B(A);

Good Copying



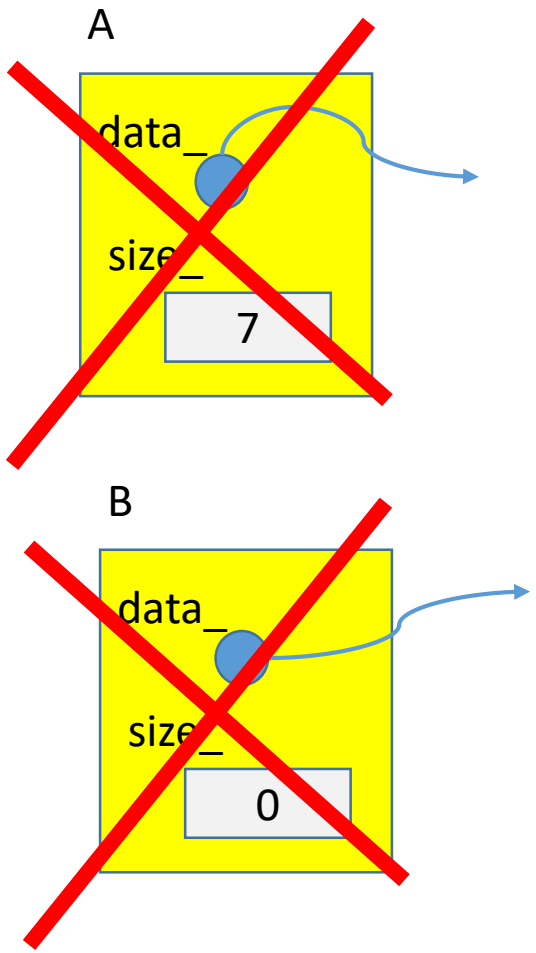

```
Copy Constructor or operator=()
will copy the data:
for(i=0;i<size_;i++){
   data_[i] = A.data_[i];
}
```

Good Copying



A and B go out of scope or are deleted then:
First destructor will delete the memory

Good Copying



Then classes will be removed
by the system.
Everything is OK!