Destructor & Copy Constructor

Destructor (dtor)

- **Special member function used for clean up**
- Destructor gets called before object gets destroyed
- Used for deleting dynamically allocated memory of Data Members
- **If programmer does not implement destructor, compiler provide default destructor for clean up.**
- Destructor can not have parameters.

Example:

```
~Complex()
{
//Clean Up code
}
```

Destructor Example


```
#include<iostream>
using namespace std;
class Sample{
private : int id;
public:
Sample(int n): id(n){
cout<<"\n Object with id="<<id<<" Initialized";
}
~Sample(){
cout<<"\n Object with id="<<id<<" Destroyed";
}};
```

```
int main(){
 Sample s1(1);
 {
 Sample s2(2);
 } //s2 gets destroyed here
 cout<<"\n s1 is alive";
return 0;
} // s1 gets destroyed here
```

Need of Destructor

- All classes does not need destructor to be implemented by programmer. **Compiler provides destructor is sufficient in some classes.**
- **Destructor is only needed to destroy dynamically allocated memory of data members**
- Ex. Complex class does not need destructor
- Ex. Array class need destructor.

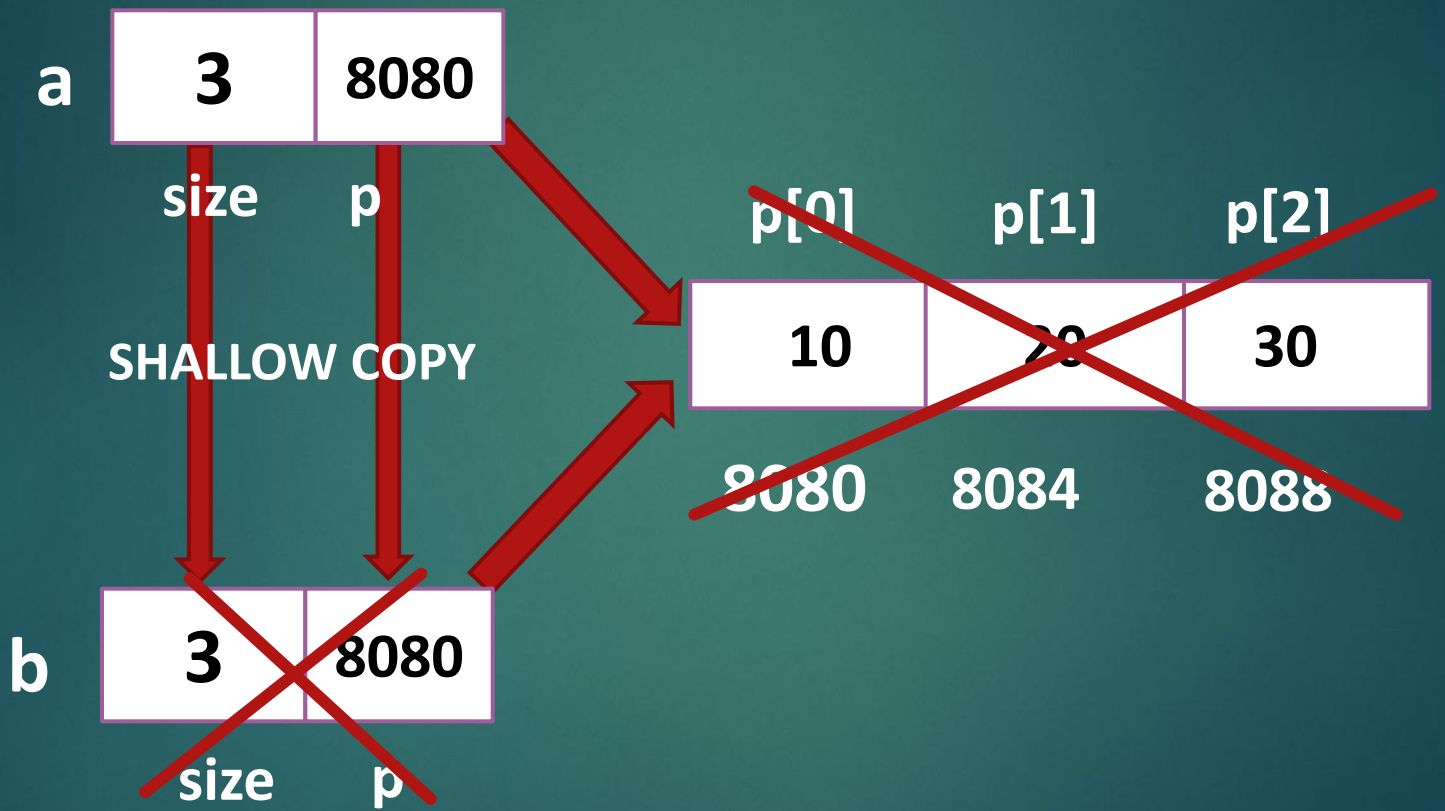
Copy Constructor

- **Copy constructor is needed to create new object using existing object**
- If programmer does not implement Copy Constructor, compiler provides default copy constructor.
- **Compiler provided copy constructor is sufficient for classes where data members are not allocated dynamically.**
- Compiler provide copy constructor does SHALLOW/ BIT COPY
- **DEEP COPY is required for classes where memory to data members is allocated dynamically**
- Compiler provided implementation of assignment Operator also does shallow copy

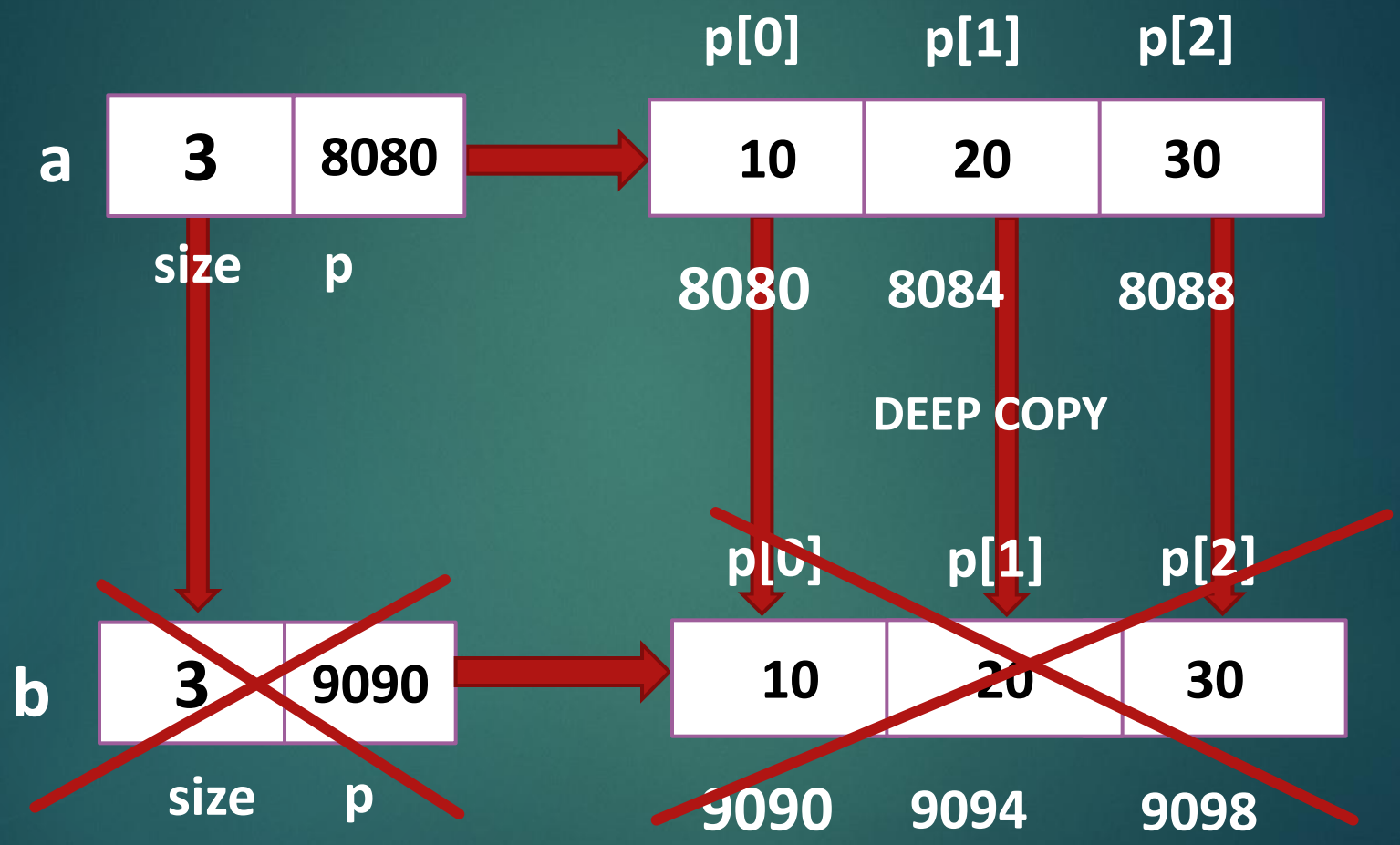
Shallow Copy Vs Deep Copy

SHALLOW COPY	DEEP COPY
The default copy constructor and default assignment operators do shallow copies, which is fine for classes that contain no dynamically allocated variables	Classes with dynamically allocated variables need to have a copy constructor and assignment operator that do a deep copy.
Shallow copy copies data members bit by bit without any intelligence. It is also called as bit copy.	Deep copy copies contents pointed by data members
Ex. Complex class	Ex. Array class

Shallow Copy



Deep Copy


p[0]
p[1]
p[2]
a
3
8080
10
20
30
size
p
8080
8084
8088
DEEP COPY
p[0]
p[1]
p[2]
b
3
9090
10
20
30
size
p
9090
9094
9098

Thank You

Enjoy copying!!!!! ……COPY CTOR