

# OPP's in detail

## # Shallow & Deep Copy

CODE:- #include <iostream>  
#include <string.h>  
using namespace std;

```
Class Student {  
    int age;  
    char *name;
```

Public :

```
Student(int age, char *name) {
```

```
    this -> age = age;
```

```
    this -> name = name;
```

```
}
```

```
Void display() {
```

```
    cout << name << " " << age << endl;
```

```
}
```

```
};
```

```
int main() {
```

```
    char name[] = "bhavy";
```

```
    Student s1(20, name);
```

```
    s1.display();
```

```
    name[3] = 'e'; // changing to bhaey
```

```
    Student s2(24, name);
```

```
    s2.display();
```

```
    s1.display();
```

```
}
```

Output:-

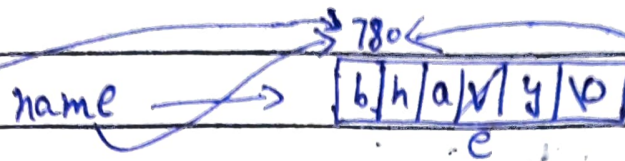
bhavy 20

bhaey 24

bhaey 24

i.e., after changing the name a bit, the changes happened to be in both S1 & S2, lets see why this happens.

In int main —



Now In our Class -

S1. Student (20, name);  
6780

age 20.

home 780

```
age = 20
name = 780
```

81

|            |
|------------|
| age = 24   |
| name = 780 |

82

i.e., name, S1 & S2 are Pointing at same array only, i.e., changing in any of these will leads to change for Everyone & this Copying is called Shallow Copy.

Sol<sup>n</sup> Use deep Copy, i.e, create a new array with same values so that changes will not Reflect

→ it is clear that changes need to be made in class only main has no use in internal working.

⇒ So let's change the class according to deep copy

```
Class Student {
```

```
    int age;
```

```
    char *name;
```

```
Public:
```

```
Student(int age, char *name) {
```

```
    this → age = age;
```

```
    // Shallow Copy
```

```
    // this → name = name;
```

```
    // Deep Copy
```

```
    this → name = new char [strlen(name) + 1];
```

```
    strcpy(this → name, name);
```

```
}
```

```
Void display() {
```

```
    cout << " ";
```

```
}
```

```
};
```

according to this code →

age = 20  
name = 890

s<sub>1</sub>

890

b h a v y \0

name

Student s<sub>1</sub> (20, name)

780

b h a v y \0

780

age = 20, name = 780

now, If we change in any of the array it won't reflect in other —

output :-

bhavy 20

bhavy 24

bhavy 20