

## Tutorial 14

1. Create two **dynamic arrays** that can store the **name** and **weight** of bag for passengers in a cruise ship. Follow the instructions below:
  - a. Ask the user to enter the **number of passengers** in the cruise. Then ask the user to **enter the names and weight** of bag for each passenger.
  - b. Ensure that the names entered are added to the name array and the weight added is **added to the weight array**.
  - c. Create a function called **heaviest\_bag** to indicate the weight of the heaviest bag.
  - d. Create a function called **passenger\_list** to list out all the names of the passengers in the cruise.
2. Run these codes in Dev C++ and watch the output of the code. Then explain the areas that are indicated in yellow. Are the areas in yellow shallow copy or deep copy?

```
#include<iostream>
using namespace std;
```

```
class DummyClass
{
private:
    int num1, num2;

public:
    void setData(int x, int y)
    {
        num1 = x;
        num2 = y;
    }

    void showData()
    {
        cout << num1 << " " << num2 << endl;
    }
};
```

```
int main()
{
    DummyClass D1;
    D1.setData(5, 11);
    D1.showData();
    DummyClass D2 = D1;
    D2.showData();

    DummyClass D3;
    D3.setData(1, 11);
    D3.showData();
}
```

```
DummyClass D4;
```

```
D4 = D3;
```

```
D4.showData();
```

```
return 0;
```

```
}
```

**Answer**

**This is called shallow copy**

**In the above example, object D4 is created and then it is initialized with object D3. In this case, copy assignment operator which is overloaded by compiler itself is implemented.**

**When we create a copy of an object:**

- 1. copy constructor will be called or**
- 2. implicit copy assignment operator will be called.**

**When will Copy Constructor call?**

**When you initialize an object with another object of same class during its declaration, copy constructor will be called.**

**For Example**

```
DummyClass D1;
```

```
D1.setData(5, 11);
```

```
D1.showData();
```

**//Copy constructor (created by compiler) will come in action here.**

```
DummyClass D2 = D1;
```

```
D2.showData();
```

**When will Copy Assignment Operator call?**

**When you do not initialize an object during its declaration but initialize it after its declaration with another object of same class then copy assignment operator will be called.**

```
DummyClass D1;
```

```
D1.setData(5, 11);
```

```
D1.showData();
```

```
DummyClass D2;
```

**//Copy assignment operator overloaded by compiler came in action.**

```
D2 = D1;
```

**D4.showData();**

**As a result, creating a copy of object by copying data of all member variables as it is, it is called shallow copy. The compiler will do shallow copy by default.**

3. Run the code below in Dev C++. Observe the output. Describe what is shallow copy based on the areas indicated in yellow.

```
#include<iostream>
```

```
using namespace std;
```

```
class DummyClass {
```

```
private:
```

```
int num1, num2;
```

```
int *ptr; //A pointer called ptr is created as one of the member variable
```

```
public:
```

```
DummyClass() {
```

```
ptr = new int; //This is the default constructor. A new pointer will be created
```

```
}
```

```
void setData(int x, int y, int z) //this is to assign values to num1 and num2 and assign address to pointer
```

```
num1 = x;
```

```
num2 = y;
```

```
*ptr = z;
```

```
}
```

```
void showData() {
```

```
cout << "A = " << num1 << " B = " << num2 << endl;
```

```
}
```

```
//copy constructor
```

```
DummyClass(DummyClass &D) { //creating a new object by copying values form another object is called deep copy. This is a copy constructor responsible for deep copy
```

```
num1 = D.num1;
```

```
num2 = D.num2;
```

```
*ptr = *(D.ptr);
```

```
}
```

```
//destructor -> to deallocate memory consumed by new pointer ptr.
```

```
~DummyClass() {
```

```
delete ptr;
```

```
}
```

```
};
```

```
int main() {  
    DummyClass D1;  
    D1.setData(3, 5, 11);  
    D1.showData();
```

```
    //Copy constructor declared above will come in action and do deep copy.  
    DummyClass D2 = D1;  
    D2.showData();  
}
```

Better read this article: <https://www.studytonight.com/cpp/copy-constructor-in-cpp.php>

4. Create a linked list that can add students to a classroom. Print out all the names of the students in the classroom.
5. Discuss the features of stack. List out the steps for PUSH operations and POP operation.

#### **Basic features of Stack**

**Stack is an ordered list of similar data type.**

**Stack is a LIFO (Last in First out) structure or we can say FILO (First in Last out).**

**push() function is used to insert new elements into the Stack and pop() function is used to remove an element from the stack. Both insertion and removal are allowed at only one end of Stack called Top.**

#### **Algorithm for PUSH operation**

**Check if the stack is full or not.**

**If the stack is full, then print error of overflow and exit the program.**

**If the stack is not full, then increment the top and add the element.**

#### **Algorithm for POP operation**

**Check if the stack is empty or not.**

**If the stack is empty, then print error of underflow and exit the program.**

**If the stack is not empty, then print the element at the top and decrement the top.**