// C++ program for the above approach

#include <iostream>

using namespace std;

// Box Class

class box {

private:

int length;

int breadth;

int height;

public:

// Function that sets the dimensions

void set\_dimensions(int length1, int breadth1,

int height1)

{

length = length1;

breadth = breadth1;

height = height1;

}

// Function to display the dimensions

// of the Box object

void show\_data()

{

cout << " Length = " << length

<< "\n Breadth = " << breadth

<< "\n Height = " << height

<< endl;

}

};

// Driver Code

int main()

{

// Object of class Box

box B1, B3;

// Set dimensions of Box B1

B1.set\_dimensions(14, 12, 16);

B1.show\_data();

// When copying the data of object

// at the time of initialization

// then copy is made through

// COPY CONSTRUCTOR

box B2 = B1;

B2.show\_data();

// When copying the data of object

// after initialization then the

// copy is done through DEFAULT

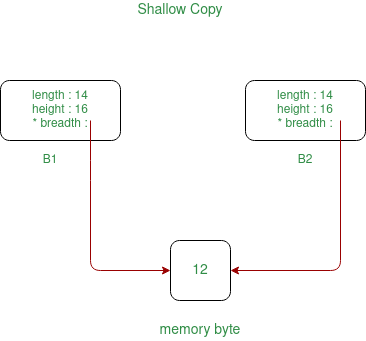
// ASSIGNMENT OPERATOR

B3 = B1;

B3.show\_data();

return 0;

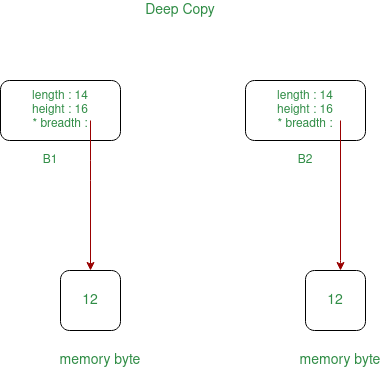
}



—-----------------------------------------------------------------------------------------------------

### **Deep Copy:**

In Deep copy, an object is created by copying data of all variables, and it also allocates similar memory resources with the same value to the object. In order to perform Deep copy, we need to explicitly define the copy constructor and assign dynamic memory as well, if required. Also, it is required to dynamically allocate memory to the variables in the other constructors, as well.



// C++ program to implement the

// deep copy

#include <iostream>

using namespace std;

// Box Class

class box {

private:

int length;

int\* breadth;

int height;

public:

// Constructor

box()

{

breadth = new int;

}

// Function to set the dimensions

// of the Box

void set\_dimension(int len, int brea,

int heig)

{

length = len;

\*breadth = brea;

height = heig;

}

// Function to show the dimensions

// of the Box

void show\_data()

{

cout << " Length = " << length

<< "\n Breadth = " << \*breadth

<< "\n Height = " << height

<< endl;

}

// Parameterized Constructors for

// for implementing deep copy

box(box& sample)

{

length = sample.length;

breadth = new int;

\*breadth = \*(sample.breadth);

height = sample.height;

}

// Destructors

~box()

{

delete breadth;

}

};

// Driver Code

int main()

{

// Object of class first

box first;

// Set the dimensions

first.set\_dimension(12, 14, 16);

// Display the dimensions

first.show\_data();

// When the data will be copied then

// all the resources will also get

// allocated to the new object

box second = first;

// Display the dimensions

second.show\_data();

return 0;

}