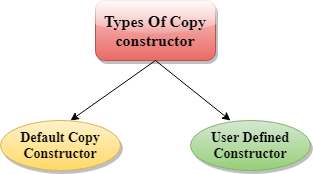
* **Copy Constructor in C++**

A **copy constructor** is a member function that initializes an object using another object of the same class. Copy constructor is used to initialize the members of a newly created object by copying the members of an already existing object.Copy constructor takes a reference to an object of the same class as an argument.

## Copy Constructor is of two types:

* **Default Copy constructor:** The compiler defines the default copy constructor. If the user defines no copy constructor, compiler supplies its constructor.
* **User Defined constructor:** The programmer defines the user-defined constructor.



## Syntax Of User-defined Copy Constructor:

Class\_name(**const** class\_name &old\_object);

Consider the following situation:

**class** A

{

    A(A &x) //  copy constructor.

   {

       // copy constructor.

   }

}

In the above case, **copy constructor can be called in the following ways:**

C++ Copy Constructor

## When Copy Constructor is called

Copy Constructor is called in the following scenarios:

* When we initialize the object with another existing object of the same class type. For example, Student s1 = s2, where Student is the class.
* When the object of the same class type is passed by value as an argument.
* When the function returns the object of the same class type by value.

**// program of the copy constructor.**

#include <iostream>

**using** **namespace** std;

**class** A

{

**public**:

**int** x;

    A(**int** a)                // parameterized constructor.

    {

      x=a;

    }

    A(A &i)               // copy constructor

    {

        x = i.x;

    }

};

**int** main()

{

  A a1(20);               // Calling the parameterized constructor.

 A a2(a1);                //  Calling the copy constructor.

 cout<<a2.x;

**return** 0;

}

**Output:**

20

**Example2 :**

#include <iostream>

using namespace std;

// declare a class

class Wall {

private:

double length;

double height;

public:

// initialize variables with parameterized constructor

Wall(double len, double hgt) {

length = len;

height = hgt;

}

// copy constructor with a Wall object as parameter

// copies data of the obj parameter

Wall(Wall &obj) {

length = obj.length;

height = obj.height;

}

double calculateArea() {

return length \* height;

}

};

int main() {

// create an object of Wall class

Wall wall1(10.5, 8.6);

// copy contents of wall1 to wall2

Wall wall2 = wall1;

// print areas of wall1 and wall2

cout << "Area of Wall 1: " << wall1.calculateArea() << endl;

cout << "Area of Wall 2: " << wall2.calculateArea();

return 0;

}

# Constructors With Default Arguments In C++

Default arguments of the constructor are those which are provided in the constructor declaration. If the values are not provided when calling the constructor the constructor uses the default arguments automatically.

#include<iostream>

using namespace std;

class Simple{

int data1;

int data2;

int data3;

public:

Simple(int a, int b=9, int c=8){

data1 = a;

data2 = b;

data3 = c;

}

void printData();

};

void Simple :: printData(){

cout<<"The value of data1, data2 and data3 is "<<data1<<", "<< data2<<" and "<< data3<<endl;

}