

Constructor

Ali Haider

syedalihaider.ciit@gmail.com

Department of Computer Science IUB

Overview

- What is constructor?
- How to use constructor
- Assigning default value by constructor
- Set value to object at time of declaration
- Default copy constructor
- Homework Tasks

What is constructor?

- If an object can initialize itself when it's first created, without requiring a separate call to a member function.
- Automatic initialization is carried out using a special member function called a constructor.
- A constructor is a member function that is executed automatically whenever an object is created.
- Constructor has same name as class
- No return type used for constructor

How to use constructor

- A constructor in C++ is a special method that is automatically called when an object of a class is created.
- To create a constructor, use the same name as the class, followed by parentheses ():

```
class MyClass {           // The class
public:                   // Access specifier
    MyClass() {           // Constructor
        cout << "Constructor is called";
    }
};
```

```
int main() {
    MyClass myObj;        // Create an object of MyClass (this will call the constructor)
    return 0;
}
```

 C:\Users\ShahTab\Documents\OOP\constructor.exe **Output**

Constructor is called

Process exited after 0.1373 seconds with return value 0

Press any key to continue . . .

Assigning default value by constructor

- In default value assignment by constructor the default value is assigned to object on object creation.

```
class Counter{  
    Int count;  
    Int count1;  
    Counter() : count(0),count1(1)           //constructor  
    { /*empty body*/  
        cout<<"constructor";  
    }  
};  
int main(){  
    Counter c;  
}
```

Set value to object at time of declaration

- In value assignment at object at time of declaration the value is assigned to object on object creation through the constructor.

```
class Counter{
    Int count;
    Int count1;
    Counter(int x,int y) : count(x),count1(y)           //constructor
    { /*empty body*/
        cout<<"constructor";
    }
};
int main(){
    Counter c(2,3);
}
```

Default copy constructor

- Two ways to initialize objects:
- A no-argument constructor can initialize data members to constant values, and a multi-argument constructor can initialize data members to values passed as arguments.
- Let's mention another way to initialize an object: you can initialize it with another object of the same type.
- Surprisingly, you don't need to create a special constructor for this; one is already built into all classes. It's called the default copy constructor.
- It's a one-argument constructor whose argument is an object of the same class as the constructor.

Example

```
#include <iostream>
#include <conio.h>
using namespace std;

class Point
{
private:
    int x, y;
public:
    Point(int x1, int y1) { x = x1; y = y1; }
    void show()
    {
        cout << "x = " << x << ", y = " << y << endl;
    }
};

int main()
{
    Point p1(10, 15); // Normal constructor is called here
    Point p2 = p1; // Copy constructor is called here

    // Let us access values assigned by constructors

    p1.show();
    p2.show();

    return 0;
}
```

Homework Tasks

1. Write a program that uses a class (CAR) and declare three data members, (model, part, cost) first get the model number, part name and cost. Take information of three cars from the user using Take () and then display in reverse order. Using Reverse_Print () member function. Make constructor that initialized all the data members with suitable values.
2. Write a program that takes 5 students information (name, age , marks in three subjects, and total marks) and display in that form. Using class.

Name	Age	Marks	Total marks
Ali	18	40+80+100=	220
Aslam	19	60+100+100=	260