EDUCBA
(https://www.educba.com/software-development/)

←
(https://www.educba.com/shell-sort-c/)

→
(https://www.educba.com/encapsulation-in-c/)

Introduction on Constructor in C

A Constructor in C is used in the memory management of C++programming. It allows b data types like int, float and user-defined data types such as class. Constructor in Object-oriented programming initializes the variable of a user-defined data type. Constructor helps in

```
// class with Constructor
class integer
{
int a, b;
public:
integer (void);
// declaration of Constructor
};
integer :: integer (void)
// constructor defined
{
a = 0, b = 0;
}
```

Uses of the Constructor

Below are some uses of the constructor.

- It is a special method that holds the same name as the class name and initializes t object whenever it is created. So it is simple and easy to execute.
- It is mainly used for memory management. They are used to initialize and remove memory

object and function in the program knows that the object exists

- A constructor for an object is created when an instance is an object that is declared. A class can have multiple constructors for different situations. Constructor overloading increases the versatility of the class by having many constructors in an individual class.

```
Overclass X's a, b value:: 0 , 0
Overclass X1's a, b value:: 4 ,5
Overclass X2's a, b value:: 6 , 12
```

Hence the constructor is overloaded with different values.

Types of Constructor in C

The main types of the constructor are explained as follows.

1. Default Constructor

A default constructor has no parameter or the present parameter has default values. If no user-defined constructor is present in class the compiler creates a new one if needed and that is called as default constructor. This is an inline public member of the class. This constructor will have a full body and no initializer. The default constructor doesn't provide anything specific, it simply allocates memory to the object. A constructor of Class X is superficial if all the following statements are true. It is defined implicitly. If X has no virtual base classes and functions. All the base classes and not static members of X have trivial constructors.

2. Parameterized Constructors

The constructor that can accept the arguments is called Parameterized constructors (https://www.educba.com/parameterized-constructor-in-c-plus-plus/). It can specify the

Code:

```
class X {
int i;
public:
void abc(void);
void xyz(void);
// member function definitions
}
```

```
0b1. abc();
0b1. xyz();
```

3. Copy Constructor

It is used to initialize and declare one object from another object

```
integer 10(12);
```

would define object 10 and at the same time initialize it to the value of 12. Another form of this statement is

```
integer 10 = 12;
```

The process is called copy initialization. A copy constructor is used whenever the temporary object is created. The application of copy constructor are as follows; Return of object as function value Initialization of object by another object of the same class

Syntax

```
class_name :: class_name(class_name &ptr)
class <game>
{
//data
public: game(arguments); // parameterized constructor
.........
.........
```

```
class school
{
char name[25];
int rno;
public: school(char,int); //parameterized constructor
};
school :: school (char n,int r)
{
name=n;
rno=r;
}
```

4. Constructor Overloading

When multiple constructors are used in the same class then it is called Constructor Overloading. It gives us multiple ways to initialize objects in a class. It increases flexibility by having multiple constructors in a single class.

Code:

```
#include<iostream>
#include<conio.h>
using namespace std;
class student
{
int score;
char name[30];
score =x;
strcpy(name,y);
}
student()
// normal constructor
{
score =100;
strcpy(name,"y");
}
void input_data()
{
cout<<"\n Enter score :"; cin>>score;
cout<<"\n Enter name :"; cin>>name;
}
void show_data()
{
cout<<"\n score :"<<score;
cout<<"\n Name :"<<name;
}
};
int main()
{
student s(10,"z");
s.show_data();
getch();
return 0;
```

row represents a string and column represents a string value.

Code:

```
// overloading class constructors
#include <iostream>
using namespace std;
class CRectangle
{
int width, height;
public:
CRectangle ();
CRectangle (int,int);
int area (void)
{
return (width*height);
}
};
CRectangle::CRectangle ()
{
width = 7;
height = 5;
}

CRectangle::CRectangle (int a, int b)
{
width = a;
CRectangle rect (7,5);
CRectangle rectb;
cout << "rect area: " << rect.area() << endl;
cout << "rectb area: " << rectb.area() << endl;
return 0;
}
```

Syntax

```
char arrayname[x][y];
```

where ‘x’ is the number of rows ‘y’ is the number of columns.

The number of rows should be the same as the number of strings in that array. The number of the column should be greater than or same to the length of the string or it can be plus one

Example

If there are 8 strings in array and length of the longest string is 10, the array is defined as follows

```
Char days[8][11];
```

6. Private Constructor

It is used to create an object of class for a single time. It is defined that class doesn’t hav multiple instances of the class. It is the same as other constructors but defined as private. It can access only the object of that class defined.

EDUCBA (https://www.educba.com/software-development/)

```
private:
```

Only that part of the program and can be accessed and defined within the private section only

Conclusion

A constructor may take null or more parameters. A class may define one or many constructors. It is up to us to decide which constructor to execute during object creation by passing an appropriate argument list to the constructor by setting the default value for the constructor parameter. Pointers cannot be used on constructors because their addresses cannot be considered. Constructors cannot be declared as static, const, or volatile.

Recommended Articles

This is a guide to Constructor in C. Here we discuss the basic concept, use of constructor, types of the constructor with examples, code, and outputs. You can also go through our other related articles to learn more –

1. Copy Constructor In Java (https://www.educba.com/copy-constructor-in-java/)
2. Arrays in C Programming (https://www.educba.com/arrays-in-c-programming/)
3. Constructor in Python (https://www.educba.com/constructor-in-python/)
4. Constructor in JavaScript (https://www.educba.com/constructor-in-javascript/)

EDUCBA (https://www.educba.com/software-development/)

Lifetime Access

Learn More

(https://www.educba.com/software-development/courses/c-programming-course/?btnz=edu-blg-inline-banner3)

About Us

Blog (https://www.educba.com/blog/?source=footer)

Who is EDUCBA? (https://www.educba.com/about-us/?source=footer)

Sign Up (https://www.educba.com/software-development/signup/?source=footer)

Corporate Training (https://www.educba.com/corporate/?source=footer)

Certificate from Top Institutions (https://www.educba.com/educbalive/?source=footer)

Contact Us (https://www.educba.com/contact-us/?source=footer)

Verifiable Certificate (https://www.educba.com/software-development/verifiable-certificate/?source=footer)

Reviews (https://www.educba.com/software-development/reviews/?source=footer)

QUIZ

EDUCBA
(https://www.educba.com/software-development/)

Apps

iPhone & iPad (https://itunes.apple.com/in/app/educba-learning-app/id1341654580?mt=8)

Android (https://play.google.com/store/apps/details?id=com.educba.www)

Resources

Free Courses (https://www.educba.com/software-development/free-courses/?source=footer)

Java Tutorials (https://www.educba.com/software-development/software-development-tutorials/java-tutorial/?source=footer)

Python Tutorials (https://www.educba.com/software-development/software-development-tutorials/python-tutorial/?source=footer)

All Tutorials (https://www.educba.com/software-development/software-development-tutorials/?source=footer)

Certification Courses

All Courses (https://www.educba.com/software-development/courses/?source=footer)

Software Development Course - All in One Bundle (https://www.educba.com/software-development/courses/software-development-course/?source=footer)

Become a Python Developer (https://www.educba.com/software-development/courses/python-certification-course/?source=footer)

Java Course (https://www.educba.com/software-development/courses/java-course/?source=footer)

Become a Selenium Automation Tester (https://www.educba.com/softwa
development/courses/selenium-training-certification/?source=footer)

QUIZ

Become an IoT Developer (https://www.educba.com/software-development/courses/iot-course/?source=footer)

EDUCBA
(https://www.educba.com/software-development/)

QUIZ