

# Constructors

---

MS. KANWAL LODHI

# Constructors

---

- The constructor is a special member function of a class that is executed whenever an object is created.
- A constructor will have same name as the class and it does not have any return type at all, not even void.  
[cslearnerr.com](http://cslearnerr.com)
- Constructors can be very useful for initializing the data members.

# Syntax

---

Syntax:

```
class A
```

```
{
```

```
public:
```

```
    cslearnerr.com
```

```
A() //constructor name is same as class name
```

```
//body of the constructor
```

```
}};
```

# Characteristics

---

## **Characteristics of constructors:**

Constructors have same name as the class name.

Constructors are executed whenever an object is created or declared.

Constructors have neither return type, nor void.

The main function of the constructor is to initialize the data members.

Constructors can have default values and can be overloaded.

The constructor without arguments is called "Default Constructor".

The constructor with arguments is called "Parameterized Constructor".

# Applications

---

## **Applications with Constructors**

The initialization of the data members is carried out using the constructors.

The constructor also allocates the memory to the object.

When an object is declared or created its data members are automatically initialized to the given values.

The compiler automatically calls the constructors.

The constructor is called with each object separately.

A constructor with no parameters is known as a **default constructor**.

```
1 // C++ program to demonstrate the use of default constructor
2
3 #include <iostream>
4 using namespace std;
5
6 // declare a class
7 class Wall {
8     private:
9         double length;
10
11     public:
12         // default constructor to initialize variable
13         Wall() {
14             length = 5.5;
15             cout << "Creating a wall." << endl;
16             cout << "Length = " << length << endl;
17         }
18 };
19
20 int main() {
21     Wall wall1;
22     return 0;
23 }
```

cslearnerr.com

E:\Programming-Lectures\default cons.exe

```
Creating a wall.
Length = 5.5
-----
Process exited after 0.04307 seconds with return value 0
Press any key to continue . . .
```

In C++, a constructor with parameters is known as a parameterized constructor. This is the preferred method to initialize member data.

```
2
3 #include <iostream>
4 using namespace std;
5
6 // declare a class
7 class Wall {
8     private:
9         double length;
10        double height;
11
12    public:
13        // parameterized constructor to initialize variables
14        Wall(double len, double hgt) {
15            length = len;
16            height = hgt;
17        }
18
19        double calculateArea() {
20            return length * height;
21        }
22    };
23
24 int main() {
25     // create object and initialize data members
26     Wall wall1(10.5, 8.6);
27     Wall wall2(8.5, 6.3);
28
29     cout << "Area of Wall 1: " << wall1.calculateArea() << endl;
30     cout << "Area of Wall 2: " << wall2.calculateArea();
31
32     return 0;
33 }
```

cslearner.com

E:\Programming-Lectures\parameterised cons.exe

```
Area of Wall 1: 90.3
Area of Wall 2: 53.55
-----
Process exited after 0.02598 seconds with return value 0
Press any key to continue . . .
```

# Constructor Overloading

---

- Constructors can be overloaded in a similar way as function overloading.
- Overloaded constructors have the same name (name of the class) but the different number of arguments. Depending upon the number and type of arguments passed, the corresponding constructor is called.



# Example

```
1 // C++ program to demonstrate constructor overloading
2 #include <iostream>
3 using namespace std;
4
5 class Person {
6     private:
7         int age;
8
9     public:
10        // 1. Constructor with no arguments
11        Person() {
12            age = 20;
13        }
14
15        // 2. Constructor with an argument
16        Person(int a) {
17            age = a;
18        }
19
20        int getAge() {
21            return age;
22        }
23 };
24
25 int main() {
26     Person person1, person2(45);
27
28     cout << "Person1 Age = " << person1.getAge() << endl;
29     cout << "Person2 Age = " << person2.getAge() << endl;
30
31     return 0;
32 }
```

cslearnn.com

E:\Programming-Lectures\constructor overloading.exe

```
Person1 Age = 20
Person2 Age = 45

-----
Process exited after 0.07664 seconds with return value 0
Press any key to continue . . .
```