

Constructor in C++

A class constructor is a special member function of a class that is executed whenever we create new objects of that class.

A constructor will have an exact same name as the class and it does not have any return type at all, not even void. Constructors can be very useful for setting initial values for certain member variables.

Special characteristics of Constructors:

- They should be declared in the public section
- They do not have any return type, not even void
- They get automatically invoked when the objects are created
- They cannot be inherited though derived class can call the base class constructor
- Like other functions, they can have default arguments
- You cannot refer to their address
- Constructors cannot be

```
virtual #include <iostream>
```

```
using namespace std;
```

```
class
```

```
Line {
```

```
public:
```

```
void setLength( double len );
```

```
double getLength( void );
```

```
    Line(); //This is the constructor
```

```
private:
```

```
double length;
```

```
};
```

```
// Member functions definitions including constructor
```

```
Line::Line(void) {
```

```
cout << "Object is being created" << endl;
```

```
}
```

```
void Line::setLength( double len )
```

```
{ length = len;
```

```
}
```

```
double Line::getLength( void
```

```
) { return length;
```

```
}
```

```
//Main function for the
```

```
program int main() {
```

```
Line line;
```

```

// set line length
line.setLength(6.0);
cout << "Length of line : " << line.getLength() <<endl;

return 0;
}
class A
{
int
x;
pub
lic:
A(); //Constructor
};

```

Constructors are special class functions which perform initialization of every object. The Compiler calls the Constructor whenever an object is created. Constructors initialize values to object members after storage is allocated to the object.

Types of Constructors

Constructors are of three types:

- Default Constructor

The default constructor is the constructor which doesn't take any argument. It has no parameter.

- Parametrized Constructor

These are the constructors with a parameter. Using this Constructor, you can provide different values to data members of different objects, by passing the appropriate values as an argument.

- Copy Constructor

These are a special type of Constructors which takes an object as an argument and is used to copy values of data members of one object into another object. We will study copy constructors in detail later.

Destructors

The destructor is a special class function which destroys the object as soon as the scope of an object ends. The destructor is called automatically by the compiler when the object goes out of scope.

The syntax for destructor is same as that for the constructor, the class name is used for the name of the destructor, with a tilde ~ sign as a prefix to it.

```

class A
{
public:
~A();
};

```

Destructors will never have any arguments.