

The Class Constructor

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 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. Following example explains the concept of constructor –

```
#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;
}
```

Parameterized Constructor

A default constructor does not have any parameter, but if you need, a constructor can have parameters. This helps you to assign initial value to an object at the time of its creation as shown in the following example –

```
#include <iostream>

using namespace std;
class Line {
public:
    void setLength( double len );
    double getLength( void );
    Line(double len); // This is the constructor

private:
    double length;
};

// Member functions definitions including constructor
Line::Line( double len) {
    cout << "Object is being created, length = " << len << endl;
    length = len;
}
void Line::setLength( double len ) {
    length = len;
}
double Line::getLength( void ) {
    return length;
}

// Main function for the program
int main() {
    Line line(10.0);

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

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

    return 0;
}
```