

Experiment No-02: Constructor and Destructor in OOP

Objectives

- Introduce the Constructor Class in C++.
- Define different types of constructors.
- Learn Constructor and Destructor in C++ with the help of Examples.

Example 1: Write a C++ program to demonstrate the use of the default constructor.

Data member: [length]

```
#include <iostream>
using namespace std;

// declare a class
class Wall {
private:
    double length;

public:
    // default constructor to initialize variable
    Wall() {
        length = 5.5;
        cout << "Creating a wall." << endl;
        cout << "Length = " << length << endl;
    }
};

int main() {
    Wall wall1;
    return 0;
}
```

Example 2: Write C++ program to demonstrate the use of Parameterized Constructor.

Data member: [length, height]

```
// C++ program to calculate the area of a rectangle

#include <iostream>
using namespace std;

// declare a class
class Rectangle {
private:
    double length;
    double height;
```

```
public:
    // parameterized constructor to initialize variables
    Rectangle(double len, double hgt) {
        length = len;
        height = hgt;
    }

    double calculateArea() {
        return length * height;
    }
};

int main() {
    // create object and initialize data members
    Rectangle rect1(10.5, 8.6);
    Rectangle rect2(8.5, 6.3);

    cout << "Area of Rectangle 1: " << rect1.calculateArea() << endl;
    cout << "Area of Rectangle 2: " << rect2.calculateArea();

    return 0;
}
```

Example 3: Write a C++ program to demonstrate the use of Copy Constructor.

Data member: [length, height]

```
#include <iostream>
using namespace std;

// declare a class
class Rectangle {
private:
    double length;
    double height;

public:

    // initialize variables with parameterized constructor
    Rectangle(double len, double hgt) {
        length = len;
        height = hgt;
    }

    // copy constructor with a Wall object as a parameter
    // copies data of the obj parameter
    Rectangle(Rectangle &obj) {
        length = obj.length;
        height = obj.height;
    }
}
```

```
    }

    double calculateArea() {
        return length * height;
    }
};

int main() {
    // create an object of the Rectangle class
    Rectangle rect1(10.5, 8.6);

    // copy contents of rect1 to rect2
    Wall rect2 = rect1;

    // print areas of rect1 and rect2
    cout << "Area of Rectangle 1: " << rect1.calculateArea() << endl;
    cout << "Area of Rectangle 2: " << rect2.calculateArea();

    return 0;
}
```

Example 3: Write a C++ program to understand the Destructor Class in C++.

```
#include <iostream>
#include <cstring>
using namespace std;
class Cube
{
    int side;
public:
    ~Cube()
    {
        cout<<"Destructor Called";
    }
};

int main()
{
    Cube c;
}
```

*** For better understanding please feel free to search on the internet because it is the best source of learning. ***

Practice Exercise

1. Suppose you have a Savings Account with an initial amount of 500 and you have to add some more amount to it. Create a class 'AddMoney' with a data member named 'amount' with an initial value of 500. Now make two constructors of this class as follows:

- without any parameter - no amount will be added to the Savings Account.
- having a parameter which is the amount that will be added to the Savings Account.

Create an object of the 'AddMoney' class and display the final amount in the Savings Account.

2. Write a C++ Program to define a class Car with the following specifications:

Private members:

car_name, model_name, fuel_type: string type

mileage: float type

price: double type

Public members:

displaydata(): Function to display the data members on the screen.

Use Constructor (both Default and parameterized) and destructor. When no parameter is passed the default constructor will be called with the message "Default constructor has been called".

[\[Resource Link 1\]](#)
[\[Resource Link 2\]](#)
[\[Questions\]](#)