

Department of Electronics & Telecommunication Engineering

Name of the Experiment:

Constructor and Destructor in C++.

Course No. : CSE-284
Course Title : Object Oriented Programming
Experiment No. : 02
Date of Exp. : 02-10-2024
Date of Submission : 24-10-2024

Remarks :

--

Name : Tasnia Afrin
ID : 2108050
Level : 02
Term : 02
Group : 02

Experiment Name :

Constructor and Destructor in C++

Objectives:

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

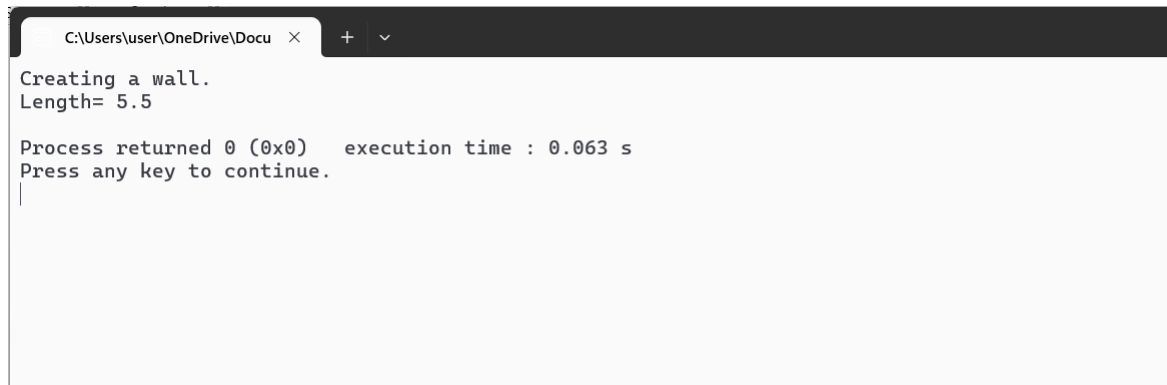
Example-1 :

Write C++ program to demonstrate the use of default constructor.

Input:

```
#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;
}
```

Output :



```
C:\Users\user\OneDrive\Docu  X  +  v
Creating a wall.
Length= 5.5

Process returned 0 (0x0)   execution time : 0.063 s
Press any key to continue.
|
```

Example-2 :

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

Input:

```
include<iostream>
using namespace std;
class Rectangle
{
private:
    double length;
    double height;
public:
    Rectangle(double len, double hgt){
        length = len;
        height = hgt;
    }
    double calcArea(){
        return length*height;
    }
};
int main()
{
    Rectangle rect1(10.5, 8.6);
    cout<<"Area of Rectangle 1: "<<rect1.calcArea()<<endl;
    Rectangle rect2(8.5, 6.3);
    cout<<"Area of Rectangle 2: "<<rect2.calcArea()<<endl;
    return 0;
}
```

Output :



```
C:\Users\user\OneDrive\Docu  × + ▾
Area of Rectangle 1: 90.3
Area of Rectangle 2: 53.55

Process returned 0 (0x0)   execution time : 0.032 s
Press any key to continue.
|
```

Example-3 :

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

Input:

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

Output :



```
C:\Users\user\OneDrive\Docu  × + ▾
Destructor Called.
Process returned 0 (0x0)   execution time : 0.093 s
Press any key to continue.
|
```

Practice Exercise: 1

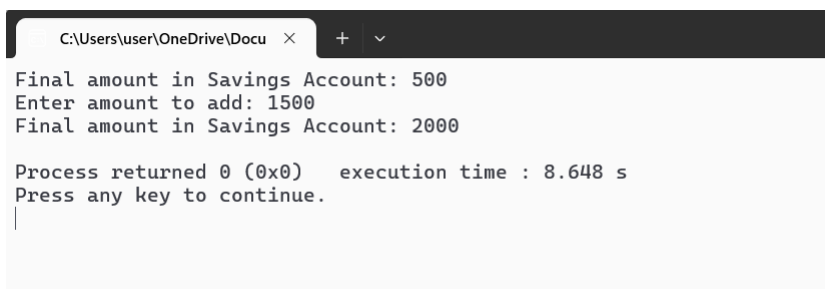
Suppose you have a Savings Account with an initial amount of 500 and you have to add some more amount to it. Create a class 'AddMoney' with a data member named 'amount' with an initial value

of 500. Now make two constructors of this class as follows: i)without any parameter - no amount will be added to the Savings Account. ii)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.

Input:

```
#include <iostream>
using namespace std;
class AddMoney {
private:
    int amount;
public:
    AddMoney() {
        amount = 500;
    }
    AddMoney(int addedAmount) {
        amount = 500 + addedAmount;
    }
    void displayAmount() {
        cout << "Final amount in Savings Account: " << amount << endl;
    }
};
int main() {
    AddMoney acc1;
    acc1.displayAmount();
    int addedAmount;
    cout << "Enter amount to add: ";
    cin >> addedAmount;
    AddMoney acc2(addedAmount);
    acc2.displayAmount();
    return 0;
}
```

Output :



```
C:\Users\user\OneDrive\Docu >
Final amount in Savings Account: 500
Enter amount to add: 1500
Final amount in Savings Account: 2000

Process returned 0 (0x0)   execution time : 8.648 s
Press any key to continue.
|
```

Practice Exercise: 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.

Input:

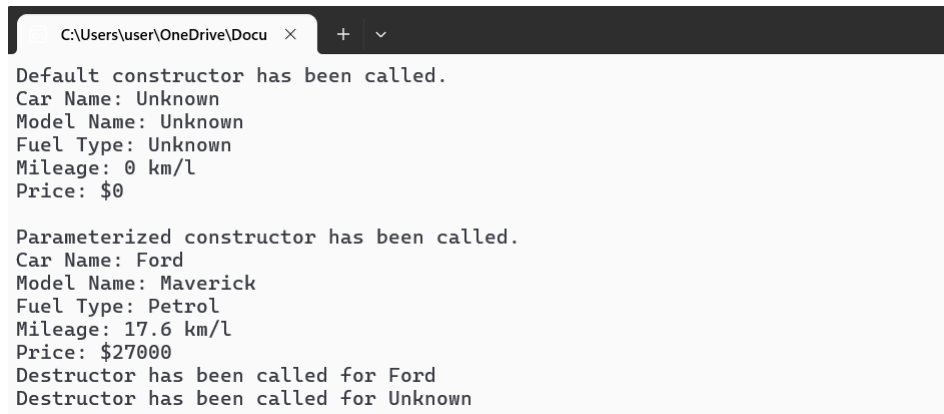
```

#include <iostream>
#include <string>
using namespace std;
class Car {
private:
    string carName;
    string modelName;
    string fuelType;
    float mileage;
    double price;
public:
    Car() {
        carName = "Unknown";
        modelName = "Unknown";
        fuelType = "Unknown";
        mileage = 0.0;
        price = 0.0;
        cout << "Default constructor has been called." << endl;
    }
    Car(string cName, string mName, string fType, float m, double p) {
        carName = cName;
        modelName = mName;
        fuelType = fType;
        mileage = m;
        price = p;
        cout << "Parameterized constructor has been called." << endl;
    }
    ~Car() {
        cout << "Destructor has been called for " << carName << endl;
    }
    void displayData() {
        cout << "Car Name: " << carName << endl;
        cout << "Model Name: " << modelName << endl;
        cout << "Fuel Type: " << fuelType << endl;
        cout << "Mileage: " << mileage << " km/l" << endl;
        cout << "Price: $" << price << endl;
    }
};

int main() {
    Car car1;
    car1.displayData();
    cout << endl;
    Car car2("Ford", "Maverick", "Petrol", 17.6, 27000.0);
    car2.displayData();
    return 0;
}

```

Output :

A screenshot of a terminal window showing the output of a C++ program. The window has a dark title bar with a tab labeled 'C:\Users\user\OneDrive\Docu' and standard window controls. The output text is as follows:

```
Default constructor has been called.  
Car Name: Unknown  
Model Name: Unknown  
Fuel Type: Unknown  
Mileage: 0 km/l  
Price: $0  
  
Parameterized constructor has been called.  
Car Name: Ford  
Model Name: Maverick  
Fuel Type: Petrol  
Mileage: 17.6 km/l  
Price: $27000  
Destructor has been called for Ford  
Destructor has been called for Unknown
```

Discussion:

In this experiment above, constructor and destructor in C++ were introduced. Here, public and private access of data members and member functions were also familiarized. Different types of default and parameterized constructor was created and observed. While solving the problems, I faced a little difficulty in understanding the use of destructor and accessing private members. Also some errors occurred while doing with parameterized constructor but then I fixed the errors soon and solved the problems as well.