



اسم المادة : برمجة الحاسوب
اسم التدريسي : زيد ابراهيم رسول
المرحلة : الثانية
السنة الدراسية : 2023-2024
عنوان المحاضرة : oop



Ministry of Higher Education and Scientific Research

Al-Mustaqbal University

Computer Engineering Techniques Department



C/C++ programming language

Prepared By

MSc. Zaid Ibrahim Rasool



اسم المادة : برمجة الحاسوب
اسم التدريسي : زيد ابراهيم رسول
المرحلة : الثانية
السنة الدراسية : 2023-2024
عنوان المحاضرة : oop



Constructor and Destructor

Constructors:

A constructor is a member function whose name is the same as the class name. The constructor is used to initialize the objects of its class, i.e. it constructs the values of data members of the class. The constructor is automatically invoked whenever object of its associated class is created.

Example:

```
#include <iostream>
using namespace std;
class integer
{
private:
    int m, n;
public:
    integer(); // constructor declaration
    void printdata();
};
integer :: integer() // constructor definition
{
    m = 0; n = 0;
}
void integer :: printdata()
{
    cout << " m = " << m << "\n" << " n = " << n << "\n";
}
void main()
{
    integer intl;
    intl.printdata();
}
```



اسم المادة : برمجة الحاسوب
اسم التدريسي : زيد ابراهيم رسول
المرحلة : الثانية
السنة الدراسية : 2023-2024
عنوان المحاضرة : oop



The declaration statement

```
integer intl;
```

not only creates the object intl of type integer but also automatically initializes its data members m and n to zero.

The output of the above program is

```
m = 0
```

```
n = 0
```

Notes:

- The constructor should be declared in the public section.
- The constructor does not have return type, (not even void), and therefore it cannot return any value.
- There is no need to write any statement to invoke the constructor function because it is invoked automatically when the object is created.
- A constructor that accepts no arguments is called the *default constructor*.
- If no constructor is defined, then the compiler creates an *implicit constructor*.
- The constructor can take arguments like other C++ functions. This is called *parameterized constructor*.

Parameterized Constructors

Sometimes we need to initialize the data elements of different objects with different values when they are created. This can be done by passing Data Structures arguments to the constructor function when objects are created. Such constructor is called *parameterized constructor*.



Example

```
#include <iostream>
using namespace std;
class integer
{
private:
int m, n;
public:
integer(int , int ); //parameterized constructor
void printdata();
};
integer :: integer(int x, int y)
{ m = x; n = y; }
void integer :: printdata()
{
cout << "m = " << m << "\n"
<<"n = " << n << "\n";
}
void main()
{
integer int1(1, 100);
int1.printdata();
}
```

The output of the above program is

m = 1

n = 100

Note that when we use parameterized constructor, we must pass the initial values as arguments to the constructor function when an object is declared.

For example, in the above program, the following declaration statement

integer int1;

may not work. In this case we need to define *multiple constructors* as in the following Example:



اسم المادة : برمجة الحاسوب
اسم التدريسي : زيد ابراهيم رسول
المرحلة : الثانية
السنة الدراسية : 2023-2024
عنوان المحاضرة : oop



```
#include <iostream>
using namespace std;
class integer
{
private:
int m, n;
public:
integer();
integer(int , int ); //parameterized constructor
void printdata();
};
integer :: integer() { m = 0; n = 0; }
integer :: integer(int x, int y)
{ m = x; n = y; }
void integer :: printdata()
{
cout << "m = " << m << "\n"
<<"n = " << n << "\n";
}
void main()
{
integer int1, int2(1, 100);
cout<<"OBJECT 1 \n";
int1.printdata();
cout<<"OBJECT 2 \n";
int2.printdata();
}
```

The output of the above program is

OBJECT 1

m = 0

n = 0

OBJECT 2

m = 1

n = 100

Note: When more than one constructor function is defined in a class, we say that *constructor overloading*.



اسم المادة : برمجة الحاسوب
اسم التدريسي : زيد ابراهيم رسول
المرحلة : الثانية
السنة الدراسية : 2023-2024
عنوان المحاضرة : oop



Destructors:

A destructor is a member function whose name is the same as the class name but is preceded by a tilde ~ . The destructor is used to destroy the objects that have been created by a constructor. For example, the destructor for the class integer is defined as follows:

```
~integer() { }
```

Notes

- **A destructor never takes any arguments nor does it return any value.**
- **A destructor will be invoked implicitly by the compiler upon exit from the program (or block or function) to free memory space.**



اسم المادة : برمجة الحاسوب
اسم التدريسي : زيد ابراهيم رسول
المرحلة : الثانية
السنة الدراسية : 2023-2024
عنوان المحاضرة : oop



Example:

```
#include <iostream>
using namespace std;
int count = 0;
class alpha
{
public :
    alpha ()
    {
        count++;
        cout << "\nObject#" << count << " is created ";
    }
    ~alpha()
    {
        cout << "\nObject#" << count << " is destroyed ";
        count--;
    }
};

void main()
{
    cout << "\n\nENTER MAIN\n";

    alpha A1, A2, A3, A4;

    {
        cout << "\n\nENTER BLOCK\n";
        alpha A5;
        alpha A6;
    }

    cout << "\n\nRE-ENTER MAIN THEN EXIT PROGRAM";
}
```



اسم المادة : برمجة الحاسوب
اسم التدريسي : زيد ابراهيم رسول
المرحلة : الثانية
السنة الدراسية : 2023-2024
عنوان المحاضرة : oop



The output of the above program is:

ENTER MAIN

Object#1 is created

Object#2 is created

Object#3 is created

Object#4 is created

ENTER BLOCK

Object#5 is created

Object#6 is created

Object#6 is destroyed

Object#5 is destroyed

RE-ENTER MAIN THEN EXIT PROGRAM

Object#4 is destroyed

Object#3 is destroyed

Object#2 is destroyed

Object#1 is destroyed

Note that the objects are destroyed in the reverse order of creation.



اسم المادة : برمجة الحاسوب
اسم التدريسي : زيد ابراهيم رسول
المرحلة : الثانية
السنة الدراسية : 2023-2024
عنوان المحاضرة : oop



HW: Write a C++ program using constructor and destructor to find the following output:

The values of X and Y and Z:

X=22

Y=14

Z=16

X= 90

Y= 99

Z=110

The object is deleted

The object is deleted