**OOP Lab #7**

**Constructor in C#**

**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 has exactly the same name as that of class and it does not have any return type.

**Default Constructor:**

A constructor which has not defined any parameters or we can say without any parameters is called default constructor. It initializes the same value of every instance of class.

**Code Syntax**

namespace Constructor\_Overloading

{

class Program

{

public Program()

{ //Default Constructor....

Console.WriteLine("Constructor is execut.");

}

static void Main(string[] args)

{

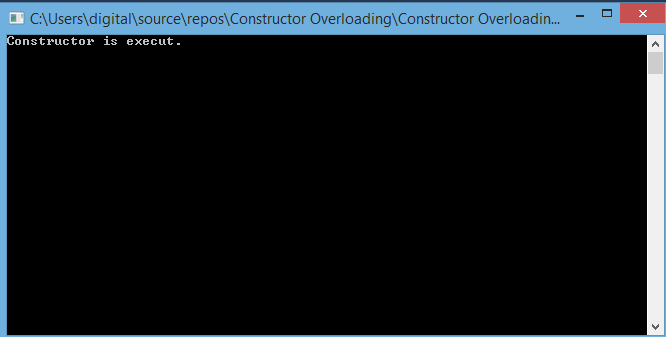
Program p = new Program();// execute when obj is make.

Console.ReadLine();

}

}

}

****

**Parameterized Constructor:**

A constructor which has at least one parameter is called Parameterized constructor. Using this type of constructor we can initialize each instance of the class to different values.

**Code Syntax**

namespace Constructor\_Overloading

{

class Program

{

class Employe

{

string name;

string fname;

int id;

public Employe(string name, string fname, int id)

{ //Parameterized Constructor....

this.name = name;

this.fname = fname;

this.id = id;

}

public string getname()

{

return this.name;

}

public string getfname()

{

return this.fname;

}

public int getid()

{

return this.id;

}

}

static void Main(string[] args)

{

Employe Emp = new Employe("Zohaib","Amjad Maseeh",51928);

Employe Emp1 = new Employe("Zeeshan", "Amjad Maseeh", 51828);

Console.WriteLine("Employee Name is :\n"+Emp.getname());

Console.WriteLine("Employee F/Name is :\n" + Emp.getfname());

Console.WriteLine("Employee Id is :\n" + Emp.getid());

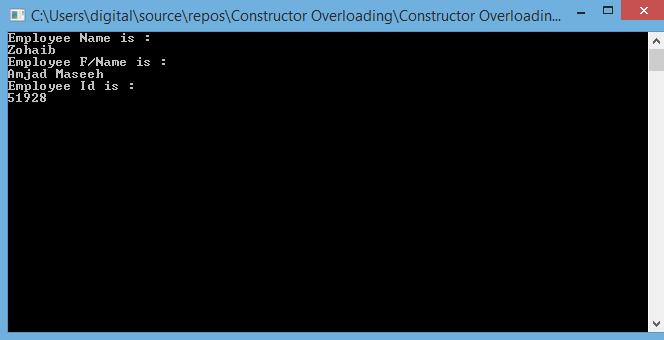
Console.ReadLine();

}

}

}

**Output**

****

**Constructor Overloading:**

When we declare more than one constructor with the same name is defined in the same name class overloaded, if the Parameters are different for each constructor.

**Code Syntext:**

namespace Constructor\_Overloading

{

class Program

{// Constructor Overloading..

class Employe

{

public Employe()//same name constructor but different by their Parameters..

{

Console.WriteLine("1st Constructor");

}

public Employe(int a,int b)

{

Console.WriteLine("2nd Constractor :{0}",(a+b));

}

public Employe(int a, int b,int c)

{

Console.WriteLine("3rd Constructor :{0}",(a+b+c));

}

}

static void Main(string[] args)

{

Employe E1 = new Employe();

Employe E2 = new Employe(10,15);

Employe E3 = new Employe(10, 15, 35);

Console.ReadLine();

}

}

}

**Output**

