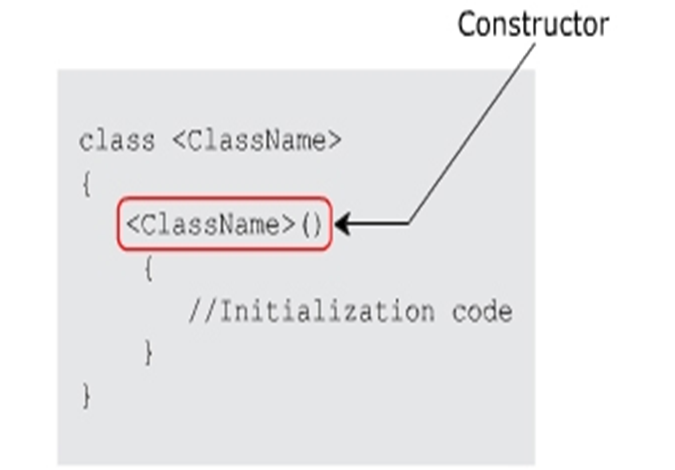
**What Is Constructor In C# ?**

* A Constructor is used to initialize objects.
* Constructors can initialize the variables of a class or perform startup operations only once when the object of the class is instantiated.
* A class constructor is a special member function of a class that is executed whenever we create new objects of that class.
* They are automatically executed whenever an instance of a class is created.
* A constructor has exactly the same name as that of class and it does not have any return type.
* A C# class can contain one or more special member functions having the same name as the class, called constructors.
* A constructor is a method having the same name as that of the class.

**The following figure shows the constructor declaration:**



**It is possible to specify the accessibility level of constructors within an application by using access modifiers such as:**

* public: Specifies that the constructor will be called whenever a class is instantiated. This instantiation can be done from anywhere and from any assembly.
* private: Specifies that this constructor cannot be invoked by an instance of a class.
* protected: Specifies that the base class will initialize on its own whenever its derived classes are created. Here, the class object can only be created in the derived classes.
* internal: Specifies that the constructor has its access limited to the current assembly. It cannot be accessed outside the assembly.

**The following code creates a class Circle with a private constructor:**

using System;

 public class Circle

{

private Circle()

{

}

}

class CircleDetails

{

public static void Main(string[] args)

{

Circle objCircle = new Circle();

}

}

**In Above Code,**

* The program will generate a compile-time error because an instance of the Circle class attempts to invoke the constructor which is declared as private. This is an illegal attempt.
* Private constructors are used to prevent class instantiation.
* If a class has defined only private constructors, the new keyword cannot be used to instantiate the object of the class.
* This means no other class can use the data members of the class that has only private constructors.
* Therefore, private constructors are only used if a class contains only static data members.
* This is because static members are invoked using the class name.

**The following figure shows the output for creating a class Circle with a private constructor:**



**The following code used to initialize the values of \_empName, \_empAge, and \_deptName with the help of a constructor:**

using System;

class Employees

{

string \_empName;

int \_empAge;

string \_deptName;

Employees(string name, intnum)

{

\_empName = name;

\_empAge = num;

\_deptName = “Research & Development”;

}

static void Main(string[] args)

{

Employees objEmp = new Employees(“John”, 10);

Console.WriteLine(objEmp.\_deptName);

}

}

**In Above Code,**

* A constructor is created for the class Employees. When the class is instantiated, the constructor is invoked with the parameters John and 10.
* These values are stored in the class variables empName and empAge respectively. The department of the employee is then displayed in the console window.

**Source Code Of Constructor**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConstructorsInCsharp

{

class Employees

{

int EmpId;

string EmpName;

int EmpAge;

public Employees(int EmpId, string EmpName, int EmpAge)

{

this.EmpId = EmpId;

this.EmpName = EmpName;

this.EmpAge = EmpAge;

}

public int getId()

{

return this.EmpId;

}

public string getName()

{

return this.EmpName;

}

public int getAge()

{

return this.EmpAge;

}

static void Main(string[] args)

{

Employees Ali = new Employees(11,"Ali Khan",22);

Employees Zain = new Employees(12, "Zain Ali", 21);

Console.WriteLine("Employee Id is {0}",Ali.getId());

Console.WriteLine("Employee Name is {0}", Ali.getName());

Console.WriteLine("Employee Age is {0}", Ali.getAge());

Console.WriteLine("--------------------");

Console.WriteLine("Employee Id is {0}", Zain.getId());

Console.WriteLine("Employee Name is {0}", Zain.getName());

Console.WriteLine("Employee Age is {0}", Zain.getAge());

Console.ReadLine();

}

}

}