**Session 15: Constructor\_TypesAndDestructorProgram**

**Constructor:** its special method of class which gets automatically invoked whenever instance of class is created. Constructor have same name as class. It cannot be abstract final and synchronised. It’s like method. In a class only one static constructor. No return type. Static constructor can’t be parameterised.

**Constructor Overlaoding:-** More than one constructor with same name. but the parameters are difrent. Can be overloaded with three types: different parameter, no of parameter, argument order.

* **Types of Constructor:** **Default Constructor:-** no parameter . it initialise the same value of very instance of class. Called when object creation.no constructor created then CLR Automatically create constructor and it will empty. Its define at compile time.
* **Parameterized Constructor:-** constructor having parameter. Initialise each instance of the class to different values. Its define runtime.
* **Copy Constructor:**- it copies data from one constructor to another.create an object by copying variables from another object. Main use to initialise new instance to values of existing instance. Parameters required for this.
* **Private Constructor:**- private keyword is used. Not possible to derive from this class(we cannot create child class) and not create instance of the same class. Used only class have static members: property, method, class variable. We can have parameter in it.

* **Static constructor:** used to initilaise static variables of the class. And perform particular action only once. Static constructor called befor instance(default or prameterised). Dosnt take parameter and dosnt use access modifiers. Only one static constructor can be create.

**Destructor:-** special method as the class. It starts with ~(tield) before the class name and immediately de-allocates memory of object. It cannot be overloaded and inherited. Cant be explicitly invoked. Cannot specify access modifiers and canot take parameters. Cannot define with **structure** only use with class. When program exit it will call. It called the finalize method.

S15\_\_ConstructorAndTypes.cs

using System;

using System.Collections.Generic;

using System.Text;

namespace OOPS\_\_AllSession

{

class S15\_\_ConstructorAndTypes //: PrivateConstructor Error

{

int id;

string name;

int age;

string address;

//Default Constructor

public S15\_\_ConstructorAndTypes()

{

int rollNo = 30;

string className = "Avengers";

Console.WriteLine($"Roll No is: {rollNo}\nClassName: {className}");

//Console.ReadLine();

}

//Destructors

~S15\_\_ConstructorAndTypes()

{

Console.WriteLine("Destructor Calle...");

}

//Parameterised Constructor

public S15\_\_ConstructorAndTypes(int empId, string empName, int empAge, string empAddress)

{

this.id = empId;

this.name = empName;

this.age = empAge;

this.address = empAddress;

this.ParameterizedConstructor();

}

//Copy Constructor

public S15\_\_ConstructorAndTypes(S15\_\_ConstructorAndTypes copyConstructor)

{

this.id = copyConstructor.id;

this.name = copyConstructor.name;

this.age = copyConstructor.age;

this.address = copyConstructor.address;

}

public void ParameterizedConstructor()

{

Console.WriteLine("\n\nYou Are in Construcotr\n Employee ID: {0}\n Employee Name :{1}\n Employee Age: {2}\n Employee Address: {3}", id, name, age, address);

Console.WriteLine("Message is: " + PrivateConstructor.msg);

PrivateConstructor.PrivateConstructors();

}

}

//Private and Static

class PrivateConstructor

{

public static string msg = "Hello Imran...";

public static string gender;

public static int age;

private PrivateConstructor() //Private

{

}

public static void PrivateConstructors()

{

DateTime dateTime = new DateTime();

Console.WriteLine("\n\nYou Are in Private Constructor\n");

Console.Write("\nFull Date Is : " + DateTime.Now);

}

static PrivateConstructor() //Static

{

gender = "Male";

age = 44;

}

public static void GetData()

{

Console.Write($"Gendr is {gender}\nAge is {age}");

}

}

}

OopsSessions.cs

using OOPS\_\_AllSession;

using System;

using static OOPS\_\_AllSession.S11\_\_ClassAndTypes;

namespace Oops\_\_AllSession

{

class OopsSessions

{

static void Main(string[] args)

{

Console.WriteLine("\*\*\*\*\*\*\*\*Welcome To Main Method\*\*\*\*\*\*\*\*\*\*\*");

//S15\_\_ConstructorAndTypes

S15\_\_ConstructorAndTypes defaultConstructor = new S15\_\_ConstructorAndTypes();

S15\_\_ConstructorAndTypes parameteriseConstructor = new S15\_\_ConstructorAndTypes(101, "Abhishek", 33, "Pune");

parameteriseConstructor.ParameterizedConstructor();

S15\_\_ConstructorAndTypes copyConstructor = new S15\_\_ConstructorAndTypes(parameteriseConstructor);

copyConstructor.ParameterizedConstructor();

//PrivateConstructor privateConstructor = new PrivateConstructor();//Not Able to Create Object

PrivateConstructor.GetData();

Console.ReadLine();

}

}

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Program\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Q]] Write Down the Program which is having Parametersised and Private Constructro along with private and non private method.

Q]] Write Down the Program using copy constructor which will contain the method and overloading of the method.