**WEEK 1**

SUPERSET ID-6362542

NAME- ANISHA RAJ

DOMAIN-DOTNET

**1.IMPLEMENTING THE SINGLETON PATTERN.**

**CODE:-**

using System;

public class Singleton

{

private static Singleton instance = null;

private static readonly object lockObj = new object();

private Singleton()

{

Console.WriteLine("Singleton instance created.");

}

public static Singleton GetInstance()

{

if (instance == null)

{

lock (lockObj)

{

if (instance == null)

{

instance = new Singleton();

}

}

}

return instance;

}

public void ShowMessage()

{

Console.WriteLine("Hello from Singleton!");

}

}

class Program

{

static void Main(string[] args)

{

Singleton obj1 = Singleton.GetInstance();

obj1.ShowMessage();

Singleton obj2 = Singleton.GetInstance();

obj2.ShowMessage();

Console.WriteLine("Are both instances the same? " + (obj1 == obj2));

}

}

**OUTPUT:-**



**2.IMPLEMENT THE FACTORY METHOD PATTERN.**

**CODE:-**

using System;

public interface IShape

{

void Draw();

}

public class Circle : IShape

{

public void Draw()

{

Console.WriteLine("Drawing a Circle");

}

}

public class Square : IShape

{

public void Draw()

{

Console.WriteLine("Drawing a Square");

}

}

public class ShapeFactory

{

public IShape GetShape(string shapeType)

{

switch (shapeType.ToLower())

{

case "circle":

return new Circle();

case "square":

return new Square();

default:

throw new ArgumentException("Unknown shape type");

}

}

}

class Program

{

static void Main(string[] args)

{

ShapeFactory factory = new ShapeFactory();

IShape shape1 = factory.GetShape("circle");

shape1.Draw();

IShape shape2 = factory.GetShape("square");

shape2.Draw();

}

}

**OUTPUT:-**

