

DPP

1. Exercise 1: Implementing the Singleton Pattern

```
using System;
```

```
public class Singleton
{
    private static Singleton instance;
    private static readonly object lockObj = new object();

    private Singleton()
    {
        Console.WriteLine("Singleton Instance Created");
    }

    public static Singleton GetInstance()
    {
        lock (lockObj)
        {
            if (instance == null)
            {
                instance = new Singleton();
            }
        }
        return instance;
    }
}

class Program
{
    static void Main(string[] args)
    {
        Singleton s1 = Singleton.GetInstance();
        Singleton s2 = Singleton.GetInstance();

        Console.WriteLine(object.ReferenceEquals(s1, s2));
    }
}
```

Output:

```
Singleton Instance Created
True
```

Exercise 2: Implementing the Factory Method Pattern

using System;

```
namespace FactoryMethod
{
    public interface IVehicle
    {
        void Drive();
    }
    public class Car : IVehicle
    {
        public void Drive()
        {
            Console.WriteLine("Driving a Car.");
        }
    }

    public class Bike : IVehicle
    {
        public void Drive()
        {
            Console.WriteLine("Riding a Bike.");
        }
    }
    public abstract class VehicleFactory
    {
        public abstract IVehicle CreateVehicle();
    }
    public class CarFactory : VehicleFactory
    {
        public override IVehicle CreateVehicle()
        {
            return new Car();
        }
    }

    public class BikeFactory : VehicleFactory
    {
        public override IVehicle CreateVehicle()
        {
            return new Bike();
        }
    }
}
class Program
{
    static void Main(string[] args)
    {
        VehicleFactory factory;
        factory = new CarFactory();
        IVehicle car = factory.CreateVehicle();
        car.Drive();

        factory = new BikeFactory();
        IVehicle bike = factory.CreateVehicle();
        bike.Drive();
    }
}
```

```
}  
}  
}
```

Output:

Driving a Car.

Riding a Bike.