Assignment: Implementing an IoC Container in C#

Objective:

Design and implement a simple Inversion of Control (IoC) container in C#. This container should be capable of managing object creation and dependency injection.

Scenario:

Imagine you are creating a basic IoC container for a simple application. This container should be able to register services and their implementations, resolve instances of these services, and inject dependencies automatically.

Requirements:

Implement a basic IoC container with the following functionalities:

Service registration: Register a service with its implementation.

Instance resolution: Resolve an instance of a registered service.

Dependency injection: Automatically inject dependencies into the constructor of the resolved instances.

Demonstrate the use of the IoC container in a simple application with the following classes:

ILogger interface with a method Log(string message).

ConsoleLogger class implementing ILogger.

IRepository interface with a method GetData().

Repository class implementing IRepository, which depends on ILogger.

Service class which depends on IRepository and has a method Serve() that uses the repository to get data and log a message.

In the main method, use the IoC container to:

Register the interfaces and their implementations.

Resolve an instance of Service.

Call the Serve() method of the resolved Service instance.

Instructions:

Define the ILogger and IRepository interfaces.

Implement the ConsoleLogger and Repository classes.

Implement the Service class.

Create a basic IoC container class with methods for service registration and instance resolution.

Demonstrate the complete setup in the main method.

Submission:

Submit your code along with a brief explanation of how you implemented the IoC container, the reasoning behind your design choices, and how the container handles dependency injection.