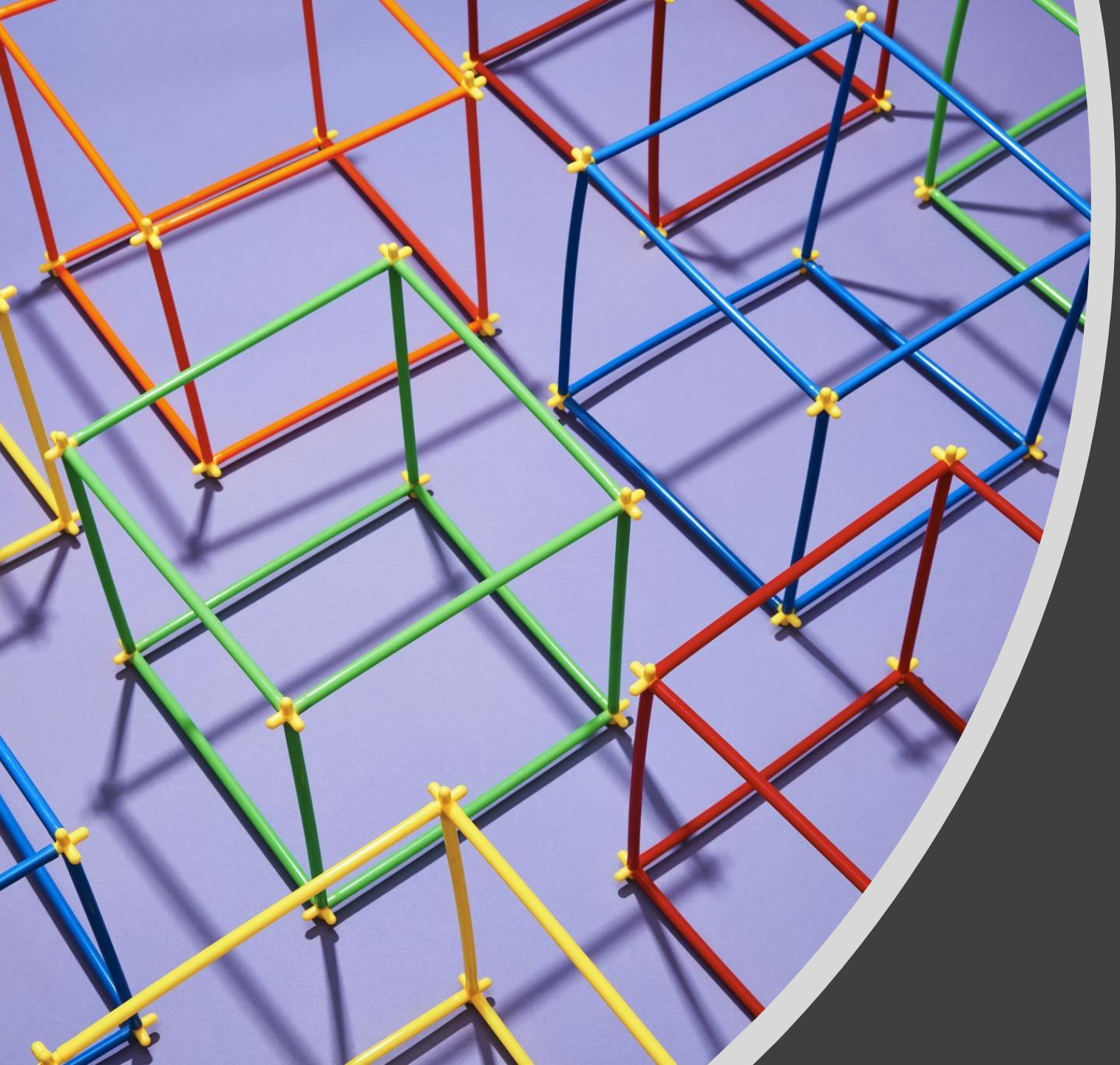
Chapter 15, 16

https://csharp.christiannagel.com

Dependency Injection, Logging, Configuration



Dependency Injection

Dependency Injection

- Hollywood Principle
- Inversion of Control
- Injection is passing of a dependency to a dependent object

Without Dependency Injection

```
public class SampleController
{
  private readonly FooService _service = new();

  public void Action()
  {
    _service.Foo();
  }
}
```

With Dependency Injection

```
public class SampleController
{
  private readonly IFooService _service;

  public SampleController(IFooService service) =>
    _service = service;

  public void Action() => _service.Foo();
}
```

Dependency Injection Container

- Register all services and contracts in one place
- Singleton, transient, or scoped type, or passing a factory method

Dependency Injection Container

- Without Container

```
SampleController controller = new(new FooService());
controller.Foo();
```

- With Container

```
var controller = Container.GetService<SampleController>();
controller.Foo();
```

Services Configuration

- Transient
- Scoped
- Singleton

Passing Configuration to Services

- Use IOptions Interface
- IOptionsSnapshot for dynamic configuration

Dependency Injection Advantages

- Unit Tests
  - Mocking
- Platform Independence
  - Create different implementations

Host Class

Host Features

- Dependency Injection Container
- Configuration
- Logging
- IHostedService

Windows Service / Systemd

IMPLEMENT A WINDOWS SERVICE

IMPLEMENT A SYSTEMD SERVICE

Configuration

Configuration Options

- Configuration Files (JSON, XML, INI)
- Environmental Variables
- Command-Line Arguments
- Provider-based
- Azure Key Vault
- Azure App Configuration

Production and Development Settings

- DOTNET_Environment environmental variable
- Use IHostEnvironment for access

User Secrets

- Don’t put secrets in source code repos!
- > dotnet user-secrets
- Secrets stored in user profile

Logging and Metrics

Logging

- ILogger
- ILoggerFactory

Log Levels

- Trace
- Debug
- Information
- Warning
- Error
- Critical

Logging Providers

- Console
  - Write messages to the console
- Debug
  - System.Diagnostics.Debug
- EventSource
  - Name: Microsoft-Extensions-Logging
- EventLog
  - Only on Windows

Filtering

- Filter logging based on providers and log levels

OpenTelemetry

- OpenTelemetry Standards
- Distributed Tracing
- AddOpenTelemetry Logging Provider
- ActivitySource.StartActivity

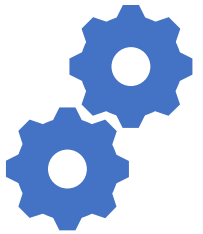
Metrics

- Read with dotnet counters
- Built-in counters: .NET, ASP.NET Core, EF Core (5.0+)
- Create a custom EventSource for custom counts
- Use XXEventCounter and XXPollingCounter

Summary

Dependency Injection

Logging



Configuration