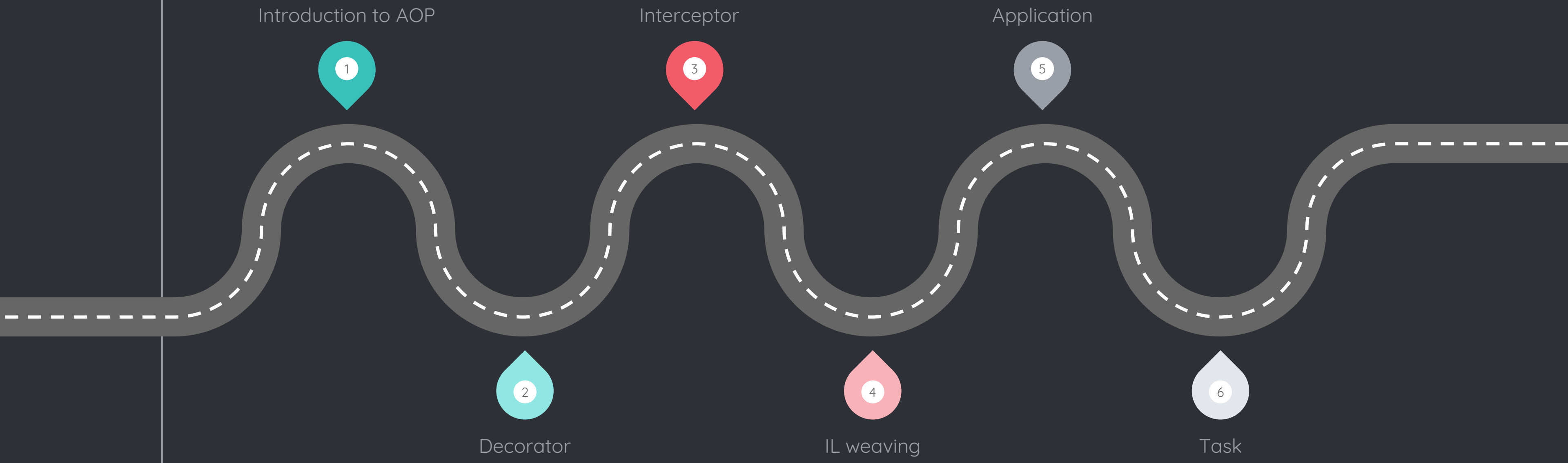
# AOP

(Aspect Oriented Programming)

Waleed Elwakeel
Full Stack developer

# Course Roadmap



# 1 Introduction to AOP

## 1.1 Definition:

Programming paradigm that aims to increase modularity by allowing the separation of crosscutting (non-functional) concerns.

Examples of crosscutting concerns:

1. Logging
2. Caching
3. Security
4. Transactions

## 1.2 Advantages:

1. Modularizes cross-cutting concerns.
2. Improving code maintainability.
3. Minimalism; keeping the code clean

# 1.3 Terminology

## 1 Cross-cutting concerns:

Functionality that occurs in multiple parts of the system which is non functional and separates it from the core of the application

## 2 Advice:

This is the additional code that you want to apply to your existing model

## 3 Pointcut:

This is the term given to the point of execution in the application at which cross-cutting concern needs to be applied.

## 4 Aspect:

The combination of the pointcut and the advice is termed an aspect.

# 1.4 Methods

1 Decorator

2 Interception

3 IL weaving

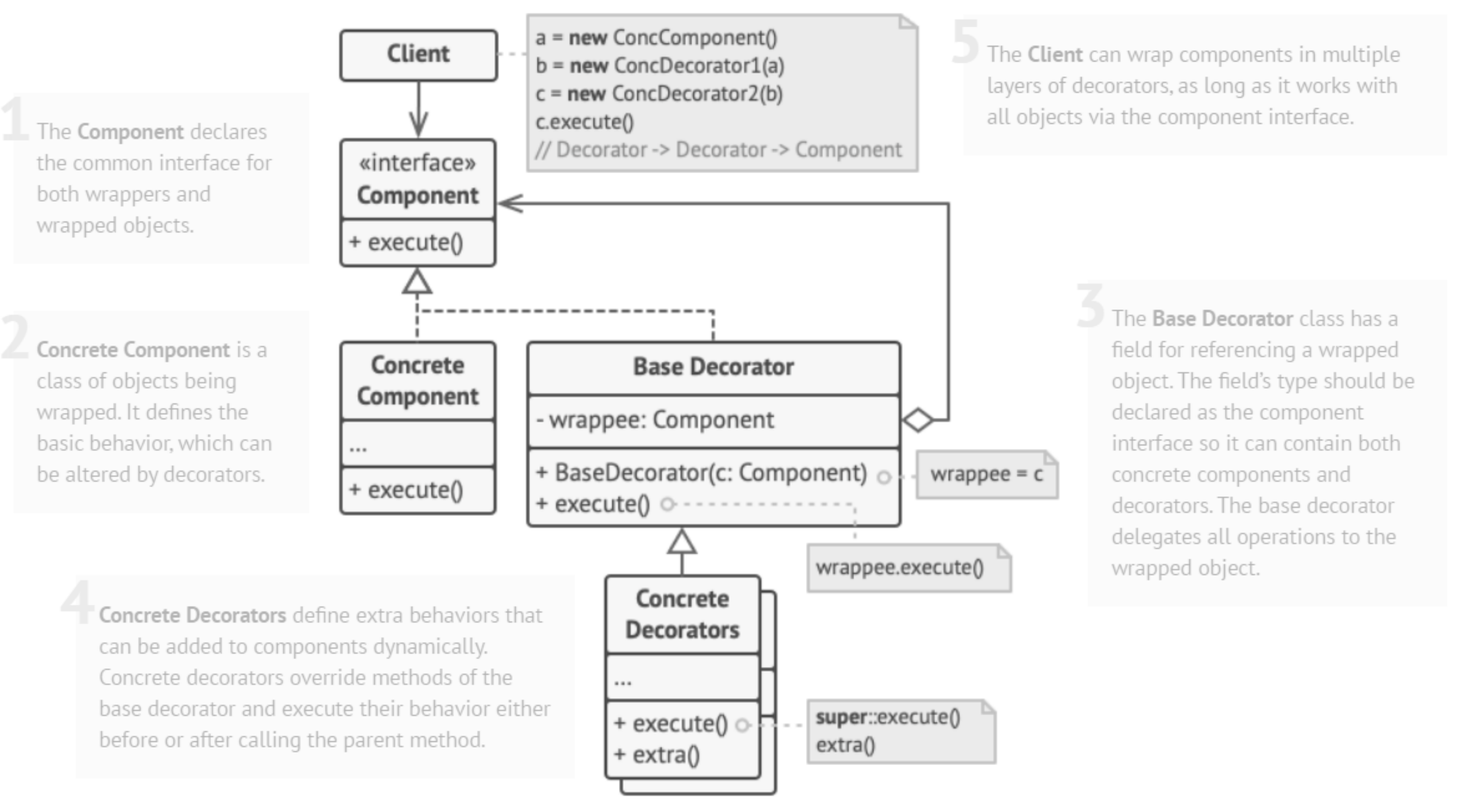
2

# Decorator

## 2.1 Definition

Decorator is a structural design pattern that lets you attach new behaviors to objects by placing these objects inside special wrapper objects that contain the behaviors.

# 2.2 Mechanism



1 The **Component** declares the common interface for both wrappers and wrapped objects.

2 **Concrete Component** is a class of objects being wrapped. It defines the basic behavior, which can be altered by decorators.

3 The **Base Decorator** class has a field for referencing a wrapped object. The field's type should be declared as the component interface so it can contain both concrete components and decorators. The base decorator delegates all operations to the wrapped object.

4 **Concrete Decorators** define extra behaviors that can be added to components dynamically. Concrete decorators override methods of the base decorator and execute their behavior either before or after calling the parent method.

5 The **Client** can wrap components in multiple layers of decorators, as long as it works with all objects via the component interface.

```
a = new ConcComponent()
b = new ConcDecorator1(a)
c = new ConcDecorator2(b)
c.execute()
// Decorator -> Decorator -> Component
```

```
wrappee = c
```

```
wrappee.execute()
```

```
super::execute()
extra()
```

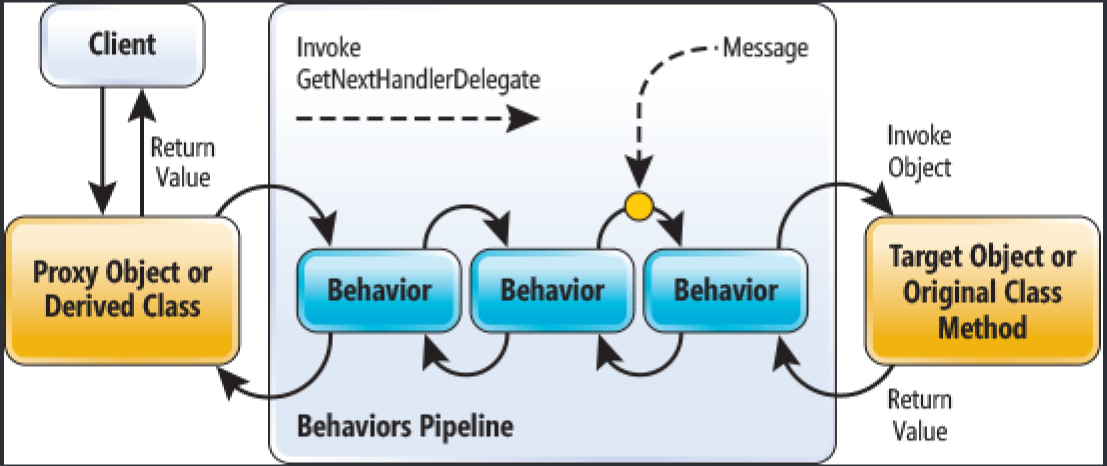
3

# Interceptor

## 3.1 Definition

Interception provides a flexible approach for adding new behaviors to an object at run time. These behaviors typically address some crosscutting concerns,

## 3.2 Mechanism


Client
Return Value
Proxy Object or Derived Class
Invoke GetNextHandlerDelegate
Message
Behavior
Behavior
Behavior
Behaviors Pipeline
Invoke Object
Target Object or Original Class Method
Return Value

## 3.3 Steps

1. Implementing interceptor interface.
2. Adding aspect logic.
3. Attaching interceptor.

4

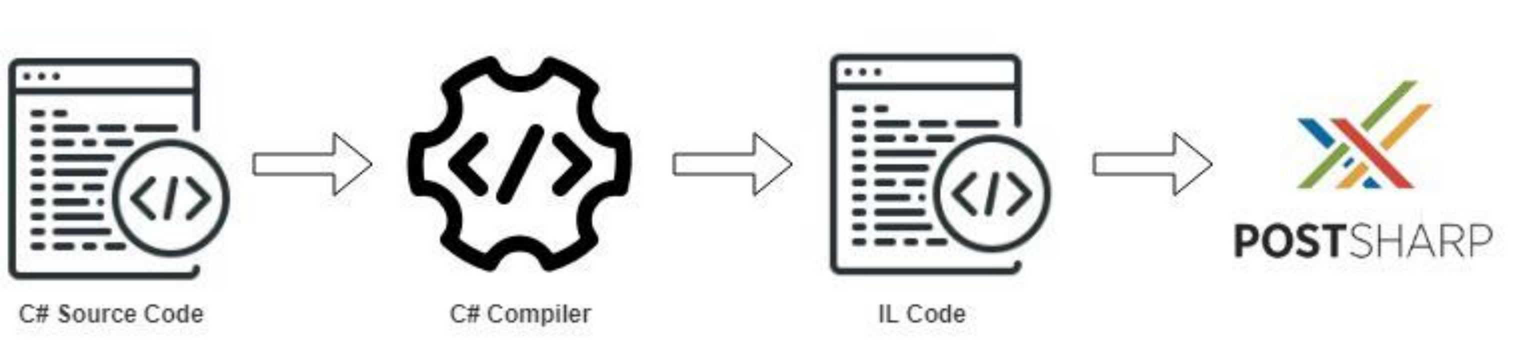
# IL Weaving

## 4.1 Definition

It's the act of manipulating a compiled .dll file and injecting additional instructions in it. This can be used, for example, to reduce the amount of boilerplate in your codebase.

## 4.2 Mechanism

(Using Postsharp)



It extends the build process not the compilation process, as instructions to the compiled binary code using post-processor.

## 4.3 Steps

(Using Postsharp)

1. Implementing the required aspect.
2. Attaching the aspect to its target.

# 5 Application

6

# Task

# References:

- https://en.wikipedia.org/wiki/Aspect-oriented_programming#Comparison_to_other_programming_paradigms
- https://docs.microsoft.com/en-us/archive/msdn-magazine/2014/february/aspect-oriented-programming-aspect-oriented-programming-with-the-realproxy-class#the-decorator-design-pattern
- https://docs.microsoft.com/en-us/archive/msdn-magazine/2014/february/async-programming-intercepting-asynchronous-methods-using-unity-interception
- https://www.youtube.com/watch?v=I4IAhhuGYJo
- https://en.wikipedia.org/wiki/Interceptor_pattern
- https://medium.com/takeaway-tech/how-aop-can-make-our-life-easy-using-postsharp-c4ec31abe0a4
- https://koukia.ca/using-il-weaving-postsharp-for-a-cleaner-code-aop-b909aae2bcd3
- https://www.todaysoftmag.com/article/861/aop-using-unity
- https://www.syncfusion.com/succinctly-free-ebooks/microsoftunity/aspect-oriented-programming-aop

Thanks!

# ANY QUESTIONS?

You can find me at

linkedin.com/in/waleedelwakeel

Waleed_elwakeel13@hotmail.com