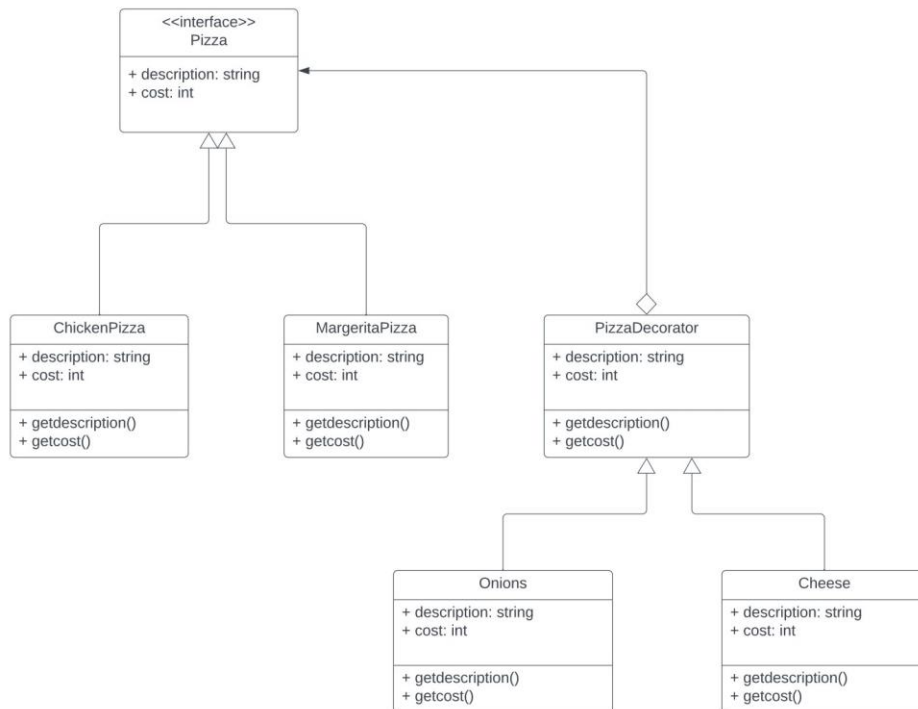# DECORATOR

Design Pattern

**INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD**

## *Report*

**Name:**            Tamoor Ahmad

**Registration No:**   4125-FBAS/BSSE/F20

**Course Code:**     SE322

**Course Title:**     Software Design and Architecture

**Submitted to:**  Dr. Muhammad Nasir

# Class Diagram:



**Consider the real-life example of Decorator design pattern as Pizza with different decorative toppings, i.e., chicken barbeque, grilled chicken, tandoori etc.**

# Source Code:

```csharp
using System;

namespace Decorator.Structural
{   /// <summary>   /// Decorator Design Pattern   /// </summary>

    public class Program
    {
        public static void Main(string[] args)
        {       // Create ConcreteComponent and two Decorators

            PizzaTopping pt = new PizzaTopping();
            TandoriPizzaDecorator p1 = new TandoriPizzaDecorator();
            GrilledPizzaDecorator p2 = new GrilledPizzaDecorator();
```

```csharp
        ChickenBarbeque p3 = new ChickenBarbeque();

        // Link decorators

        p1.SetComponent(pt);
        p2.SetComponent(p1);
        p3.SetComponent(p2);

        p3.Operation();

        // Wait for user

        Console.ReadKey();
    }
}

/// <summary>    /// The 'Component' abstract class    /// </summary>

public abstract class Pizza { public abstract void Operation(); }

/// <summary>    /// The 'ConcreteComponent' class    /// </summary>

public class PizzaTopping : Pizza
{
    public override void Operation()
    {
        Console.WriteLine("PizzaTopping.Operation()");
    }
}

/// <summary>    /// The 'Decorator' abstract class    /// </summary>

public abstract class PizzaDecorator : Pizza
{
    protected Pizza component;

    public void SetComponent(Pizza component) { this.component = component; }

    public override void Operation()
    {
        if (component != null)
        {
            component.Operation();
        }
    }
}    /// <summary>    /// The 'ConcreteDecoratorA' class    /// </summary>
public class TandoriPizzaDecorator : PizzaDecorator
{
    public override void Operation()
    {
        base.Operation();
        Console.WriteLine("TandoriPizza.Operation()");
    }
}
```

```csharp
/// <summary>     /// The 'ConcreteDecoratorB' class     /// </summary>
public class GrilledPizzaDecorator : PizzaDecorator
{
    public override void Operation()
    {
        base.Operation();
        Console.WriteLine("GrilledPizza.Operation()");
    }
}
public class ChickenBarbeque : PizzaDecorator
{
    public override void Operation()
    {
        base.Operation();
        AddedBehavior();
        Console.WriteLine("Chicken Barbeque Pizza.Operation()");
    }

    void AddedBehavior() { }
}
}
```

# Output:

Select C:\Users\ITS\source\repos\Decorator\Decorator\bin\Debug\Decorator.exe

```
PizzaTopping.Operation()
TandoriPizza.Operation()
GrilledPizza.Operation()
Chicken Barbeque Pizza.Operation()
```