**Exercise 5: Implementing the Decorator Pattern**

**Scenario:**

You are developing a notification system where notifications can be sent via multiple channels (e.g., Email, SMS). Use the Decorator Pattern to add functionalities dynamically.

**Steps:**

1. **Create a New Java Project:**

A new Java project is created and is named as **DecoratorPatternExample**.

1. **Define Component Interface:**

An interface **Notifier** is created with a method **send()**.

1. **Implement Concrete Component:**

A class **EmailNotifier** is created that implements Notifier. The source code is provided in the wordpad.

1. **Implement Decorator Classes:**
   * An abstract decorator class **NotifierDecorator** is created that implements **Notifier** and holds a reference to a **Notifier** object.
   * A concrete decorator classes like **SMSNotifierDecorator**, **SlackNotifierDecorator** is created that extend **NotifierDecorator**. The source code is provided in the wordpad.
2. **Test the Decorator Implementation:**
   * A test class is created to demonstrate sending notifications via multiple channels using decorators.The source code is provided in the wordpad.