**IMPLEMENTING THE DECORATOR PATTERN:**

1. **Creating a New Java Project**
   * A new Java project named DecoratorPatternExample was set up.
2. **Defining the Component Interface**
   * An interface called Notifier was defined with a method send() to serve as the base for notification functionality.
3. **Implementing the Concrete Component**
   * A class named EmailNotifier was created, implementing the Notifier interface to provide basic email notification functionality.
4. **Implementing Decorator Classes**
   * An abstract decorator class called NotifierDecorator was developed, implementing the Notifier interface and holding a reference to a Notifier object to enable dynamic functionality extension.
   * Concrete decorator classes such as SMSNotifierDecorator and SlackNotifierDecorator were created, extending NotifierDecorator to add specific notification channels.
5. **Testing the Decorator Implementation**
   * A test class was created to demonstrate how notifications can be sent through multiple channels using decorators, showcasing the flexibility and extensibility of the system.
6. Output

