**Software entities like classes, modules, and functions should be open for extension but closed for modifications.**

The Open Closed Principle represents the “O” of the five SOLID Principles of object-oriented programming to write well-designed code that is more readable, maintainable, and easier to upgrade and modify.

Wikipedia Says In object-oriented programming, **the open/closed principle states "software entities (classes, modules, functions, etc.) should be open for extension but closed for modification" that is, such an entity can allow its behavior to be extended without modifying its source code.**

Let's learn this design principle with class diagrams, hands-on source code, thumb rules, benefits etc.

Rules of Thumb?

**Open to an extension - you should design your classes so that new functionality can be added as new requirements are generated.**

**Closed for modification - Once you have developed a class you should never modify it, except to correct bugs.**

**Design and code should be done in a way that new functionality should be added with minimum or no changes in the existing code**

**When needs to extend functionality - avoid tight coupling, don't use if-else/switch-case logic, do code refactoring as required.**

**Techniques to achieve - Inheritance, Polymorphism, Generics**

**Pattern to apply – Strategy Pattern, Template Method**

Benefit

**The benefit of this Object oriented design principle is, which prevents someone from changing already tried and tested code.**

Open Closed Principle Example

Let's take the same example of Database Connection Provider, ideal we can have a connection to any RDBMS data source as per our requirements.

Let's design or develop source code using this design principle.

Code for the interface so first create IConnetionProvider interface and create a separate class to implement this interface for each database connection like H2ConnectionProvider, MySQLConnectionProvider.

In future, if Client wants to connect MS-Server or Oracle then we need to create the class like MsServerConnectionProvider or OracleConnectionProvider and just implement IConnectionProviderinterface that's it.

we are not modifying existing code here, we are just extending as per our requirement.

Class diagram

IConnetionProvider.java

public interface IConnetionProvider {

public Connection establishconnection();

}

H2ConnectionProvider.java

public class H2ConnectionProvider implements IConnetionProvider{

public Connection establishconnection() {

// TODO : provide connection for H2 database and return the connection object

return null;

}

}

MySQLConnectionProvider.java

public class MySQLConnectionProvider implements IConnetionProvider{

public Connection establishconnection() {

// TODO : provide connection for MySQL database

return null;

}

}

OracleConnectionProvider.java

public class OracleConnectionProvider implements IConnetionProvider{

public Connection establishconnection() {

// TODO : provide connection for Oracle database and return the connection object

return null;

}

}