

Date  
07/06/2017

ons', this snippet above violates this principle as the PasswordReminder class is being forced to depend on the MySQLConnection class.

Later if you were to change the database engine (for ex. from MySQL to Oracle), you would also have to edit the PasswordReminder class and thus violates Open-closed principle.

The PasswordReminder class should not care what database your application uses, to fix this we "code to an interface", since high level and low level modules <sup>(both)</sup> should depend on abstraction, we can create an interface:

```
interface DBConnectionInterface  
{  
    public function connect();  
}
```

The interface has a connect method and the MySQLConnection class implements this interface, also instead of directly type-hinting MySQLConnection class in the constructor of the PasswordReminder, we instead type-hint the interface and no matter the type of database your application uses, the PasswordReminder class can easily connect to the database without any problems and OCP is not violated.

*low level code*

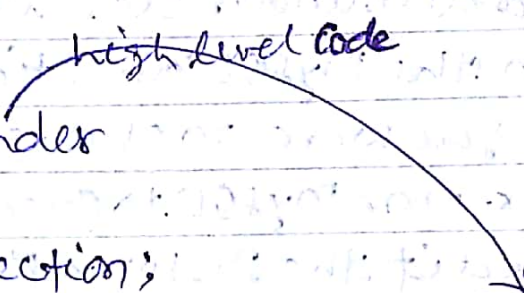
```
class MySQLConnection implements DBConnectionInterface  
{  
    public function connect  
    {  
        - - - - -  
    }
```



Date  
07/06/2017

```
return "Database connection";  
}  
}  
class PasswordReminder  
{  
    private $dbconnection;  
    public function __construct(DBConnectionInterface  
    {  
        $this->dbconnection = $dbconnection;  
    }  
}
```

high level code



According to the little snippet above, you can now see that both the high level and low level modules/code depend on abstraction.

High level code is not as concerned with details. Low level code is more concerned with details and specifics.

Date  
28/06/2017

~~When you install Xampp~~, Cautions:

- ① Set max-execution-time  $\Rightarrow$  30 minutes i.e. 1800 in xampp/php/php.ini.
- ② In xampp/apache/conf/httpd.conf, set AllowOverride  $\Rightarrow$  All instead of none.
- ③ In xampp/apache/conf/extra/httpd-xampp.conf, set AllowOverride  $\Rightarrow$  All instead of none.

Note: Always use xampp interface to open above configuration files & edit.

Use xampp Shell editor when you want to execute any command on cmd with "php".

Ex: php xyz.php ↵

php -v ↵

Otherwise, you will have to set environment variable for PHP:

Window  $\rightarrow$  search "system"  $\rightarrow$  Advanced system settings  $\rightarrow$  Environment Variables  $\rightarrow$  system variables  $\rightarrow$  edit "PATH" variable or create new one with name "PATH" if this variable does not exist. Add the value for it is:

D:\xampp\php;

and restart the system to make visible the changes.

To terminate a command execution in window cmd, use ctrl+c (same as in linux) or ctrl+Pause/Break



Date  
04/01/2018

## SymLink (Symbolic link or soft link)

In computing, a symLink is the nickname for any file that contains a reference to another file or directory in the form of an absolute or relative path and that affects path name resolution.

The following command creates a symbolic link at the command-line interface (shell):

```
ln -s target_path link_path
```

target\_path is the relative or absolute path to which the symbolic link should point. Usually the target will exist, although symbolic links may be created to non-existent targets. link\_path is the path of the symbolic link.

Reference : [https://en.wikipedia.org/wiki/Symbolic\\_link](https://en.wikipedia.org/wiki/Symbolic_link)  
<https://www.cyberciti.biz/faq/creating-soft-link-or-symbolic-link/>