# ICS 3104

# Lab 1

**Preliminaries**

We assume that you already know how to set-up a localhost server(s): & MySQL database and Apache web servers. Other tools you’ll need later include postman and engrok.

If not, your instructor will assist you to set it up.

In this lab we will learn how use **Object Orientated PHP** by creating a form, which collects information from a user, which is then stores it in a database as well as read that information from a database.

**Our tasks will be:**

* Create a MySQL database
* Create a table
* Create a form in html
* Connect to the database
* Store to a database
* Read from a database

The objective of this lab is to make clear the concepts object oriented PHP and databases;

* Classes and objects
* Interfaces
* Function and
* Databases

You will do this lab in a folder labelled lab under the htdocs XAMPP folder. You then access your application via the browser on this URL:

<http://localhost/labs/lab1.php>

After this, you be required to practically show how data can be edited/updated/changed and deleted.

# Procedure

Access the XAMPP panel and create a database

[CREATE](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/create-database.html) [DATABASE](http://localhost/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/create-database.html) btc3205; Note that btc3205 is the name of our database

Select the database and then create a table by running this query

[CREATE](http://localhost:8080/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/create-table.html) [TABLE](http://localhost:8080/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/create-table.html) [user](http://localhost:8080/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/information-functions.html#function_user)(

id [int](http://localhost:8080/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/numeric-types.html) AUTO\_INCREMENT PRIMARY KEY,

first\_name [varchar](http://localhost:8080/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/string-types.html)(32) [NOT](http://localhost:8080/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_not) NULL,

last\_name [varchar](http://localhost:8080/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/string-types.html)(32),

user\_city [varchar](http://localhost:8080/phpmyadmin/url.php?url=http://dev.mysql.com/doc/refman/5.5/en/string-types.html)(32) );

In the query above, what is the meaning of AUTO\_INCREMENT and NOT NULL?

AUTO\_INCREMENT:

NOT NULL:

Identify any SQL constraints in the query you just executed.

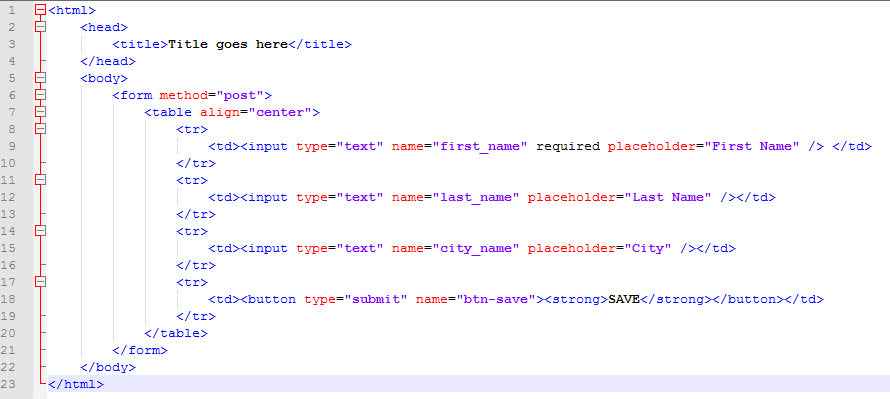
Primary Key Contraint

You just created a database with one table. We will now create an html form lab1.php, a js file to validate the form and PHP classes and an interface with code that will process data inserted in the html form.

All these files should be in the same folder lab.

Let’s go

……

To insert Data into MYSQL table we need to create html form containing all the fields the users table has and HTML code of insert form will be as follows.

Now we create classes and interfaces that we will work with

Class for database connection

Class name: DBConnector in a PHP file called DBConnector.php

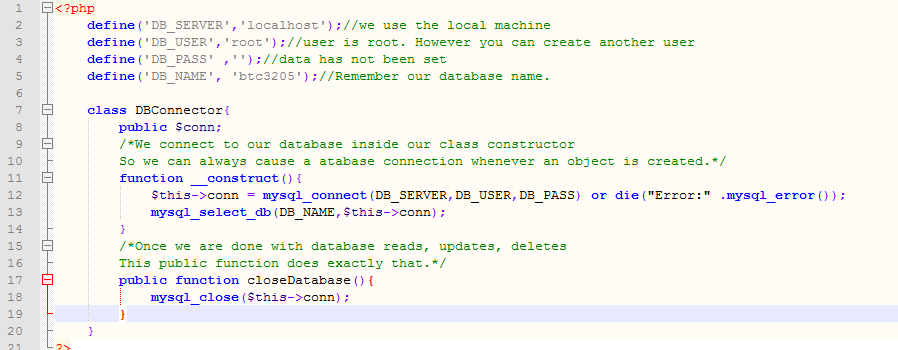
Notice the define() function

## Definition and Usage

The define() function defines a constant.

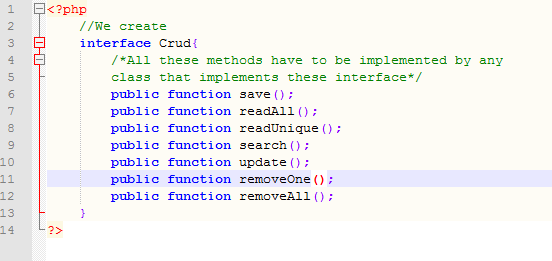
Constants are much like variables, except for the following differences:

* A constant's value cannot be changed after it is set
* Constant names do not need a leading dollar sign ($)
* Constants can be accessed regardless of scope
* Constant values can only be strings and numbers



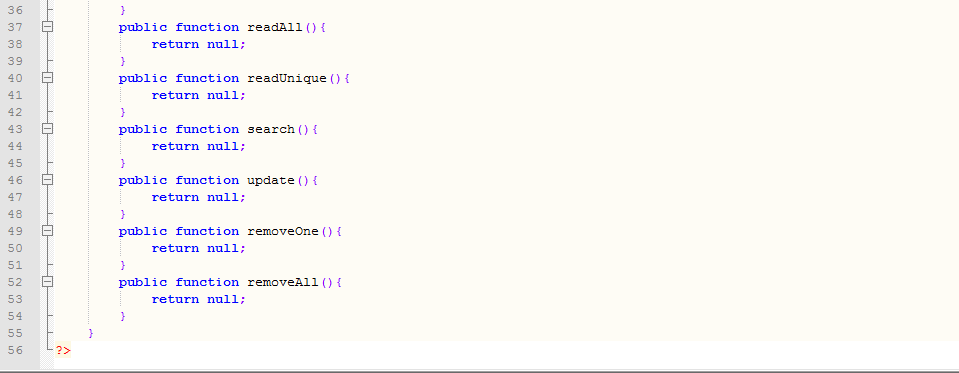
Next we create an interface. In the interface, we have all the operations we may think of, although we may not know what content goes into them.

Crud.php

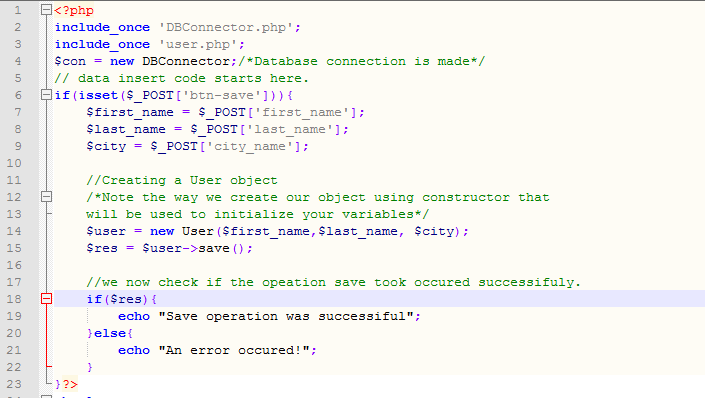


We now create a class User in a file user.php, which implements the interface Crud

As shown:

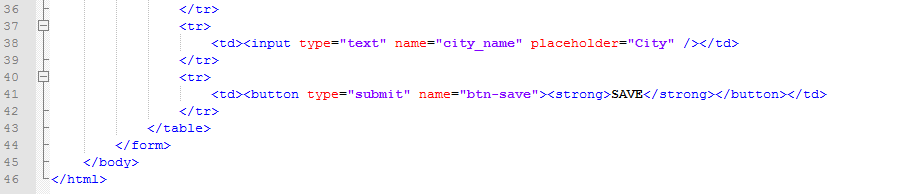
 What does the function save() return;

Now add the following code just before the html tag in lab1.php file.



So now your lab1.php looks like this.





Now you are able to save a records into your database.

Try to delete function readAll() from User class, run the program and make your observation.

Explain what is happening in line 18-21 in the file lab1.php

Remember after the save operation is complete, we need to close our database connection

We call the function closeDatabase() to close the connection. Where do you think this call should be made? Please do that.

# Your task

Now implement the function readAll() and use it to display all the records in your database.

**-This lab is complete-**