BIT 2nd Year

Semester 3

IT 3505

Web Application Development II

Server Side Web Development (PHP & MySQL) – Part 4

Duration: 30 hours

Instructional Objectives

- Install PHP in a windows environment
- Install PHP in Linux environment
- Explain basic features of PHP
- Articulate MVC architecture
- Differentiate available PHP frameworks
- Explain MVC
- Use web services with PHP
- Develop a web application with PHP

Sub Topics

1.1 Introduction to PHP (Ref 01 Pg:271-278)

1.2. Configuring the environment (Ref 01 Pg : 76 - 85)

1.3. PHP Syntax and Semantics

- 1.3.1. Variables (Ref 01 Pg:281-287)
- 1.3.2. Constants (Ref 01 pg:287 - 296)
- 1.3.3. Conditional Statements (Ref 01 pg:320-335)
- 1.3.4. Loops (Ref 01 Pg:335-346)
- 1.3.5. Functions (Ref 01 Pg: 346-357)

1.4. Arrays and data processing with arrays (Ref 01 Pg: 296-307)

1.5. Form processing with PHP (Ref 02)

1.6. Session control and Cookies (Ref 01 Pg:437-446)

1.7. File system management (Ref 01 Pg: 366-389)

1.8. Email sending using PHP (Ref 03)

1.9. Object Orientation with PHP (Ref 01 pg :397-423)

1.10. Working with MySQL database (Ref 01 PG:515-528)

1.11. Introduction to PHP frameworks (Ref 5)

1.12. Fundamentals of MVC (Ref 6)

1.13. How to call web service using PHP (Ref 01 pg:541-553)

Creating MySQL databases and Tables

Introduction to MySQL

- Released on 23rd May 1993.
- 12+ million web servers use this Database Management System(DBMS)
- Open source

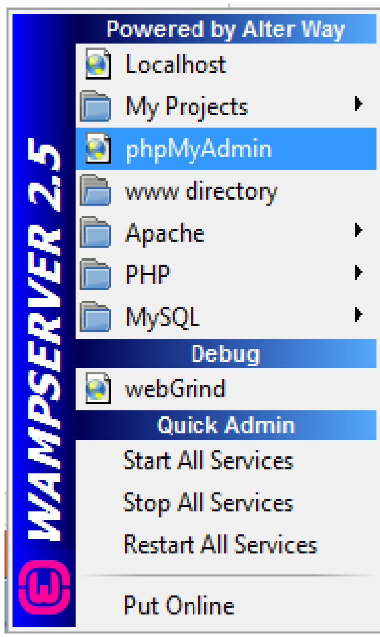
If you don't have a PHP server with MySQL, you can download it for free from the site http://www.mysql.com

Companies like Facebook, Twitter, and Wikipedia use MySQL database as their standard database

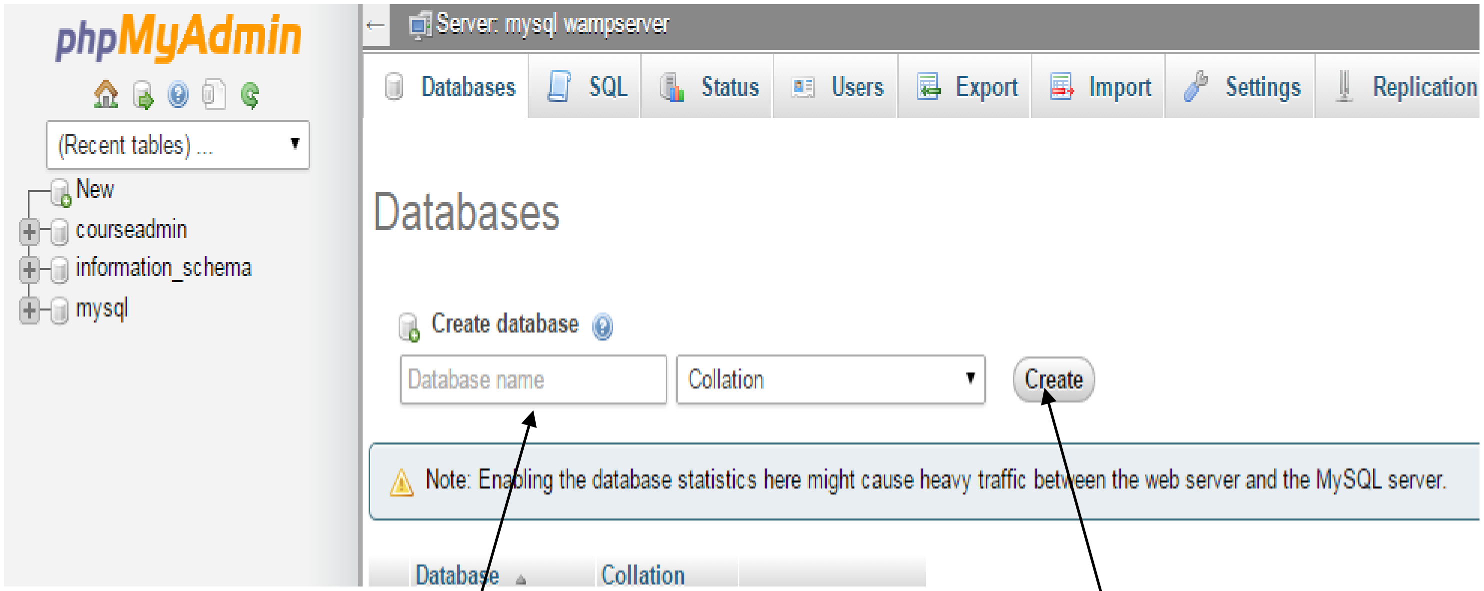
PHP 5 and later can work with a MySQL database using MySQLi extension or PDO (PHP Data Objects)

Creating a Database

- There are 2 main ways of creating a database :
  - With the command line
  - By using a tool such as MySQL workbench , phpMyAdmin
- If you are using PHP , Apache , MySQL package such as wamp or lamp , You already have phpMyAdmin installed.



Creating a database by using phpMyAdmin tool

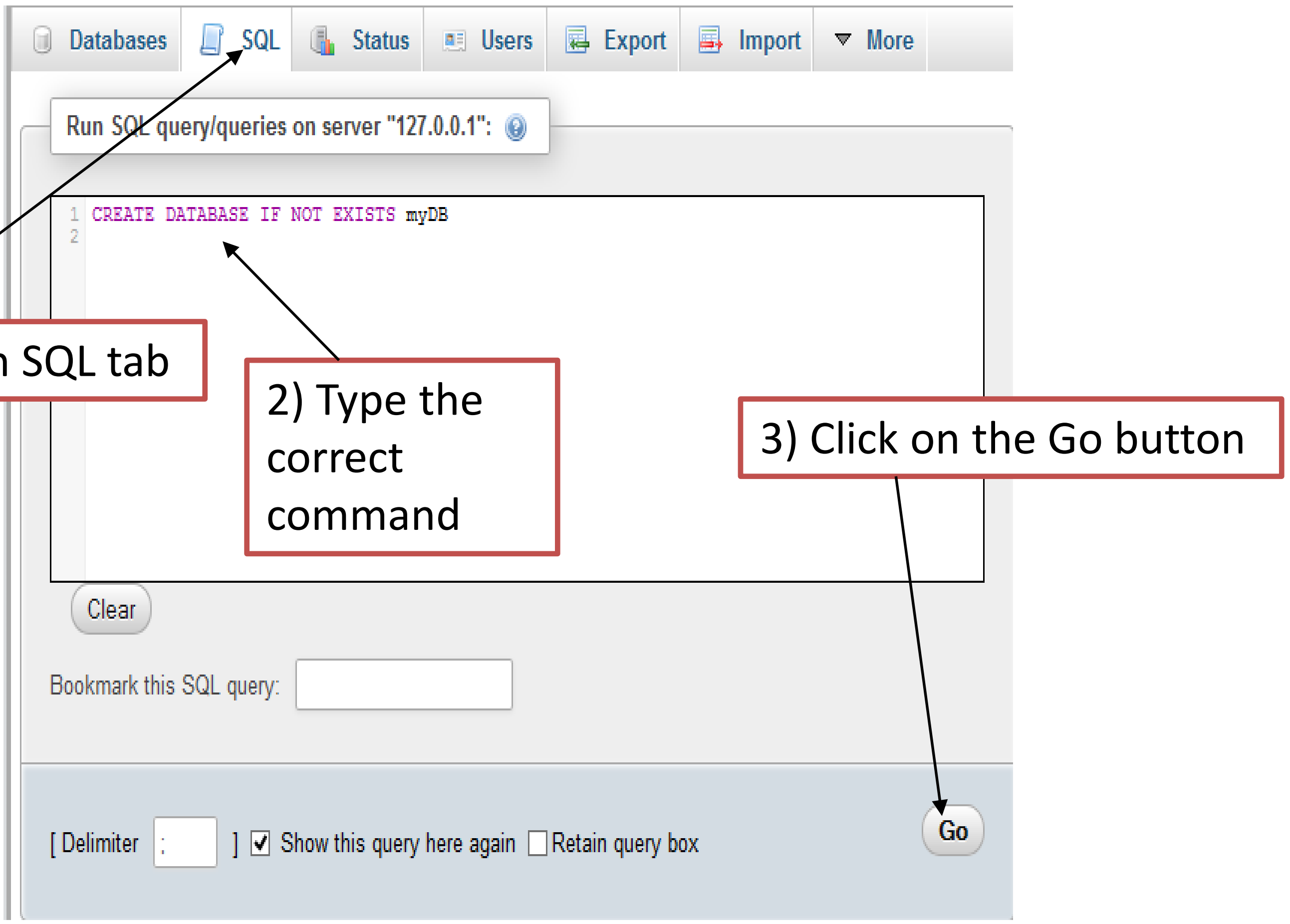


Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

1) Click on databases and give a suitable name . Click ‘(

2) Click ‘Create’

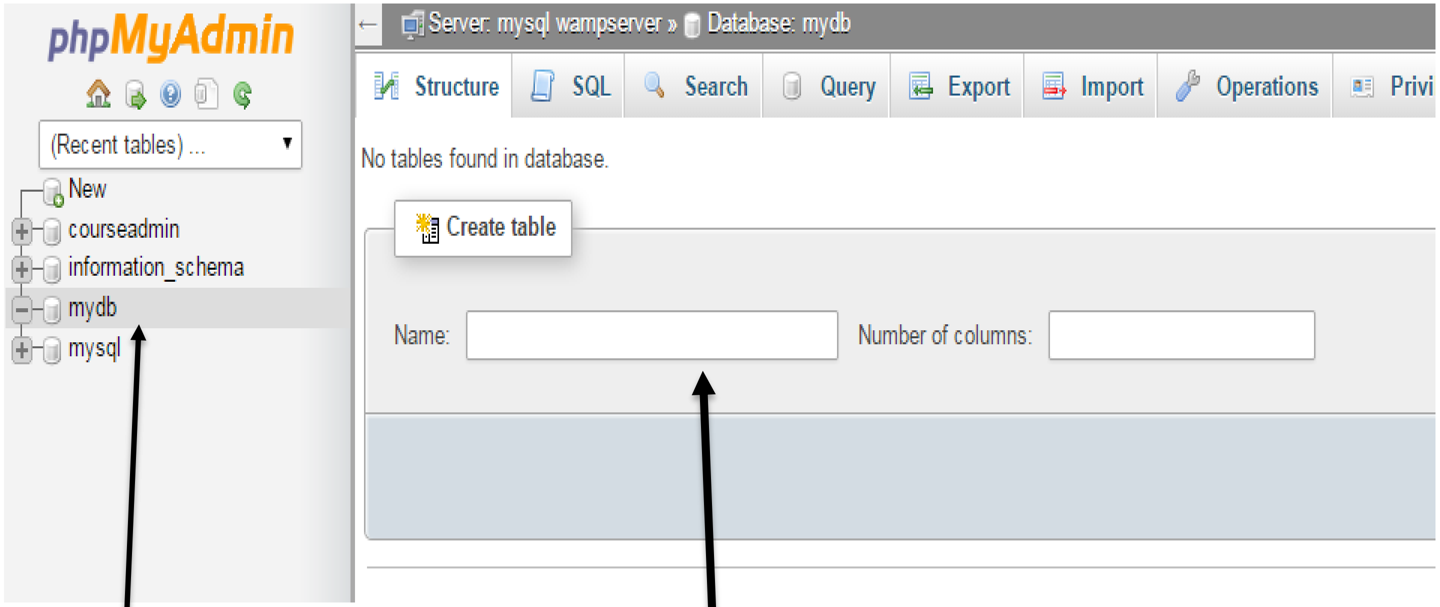
Creating Databases-(with SQL command)


Databases
SQL
Status
Users
Export
Import
More
Run SQL query/queries on server "127.0.0.1":
1) Click on SQL tab
2) Type the correct command
3) Click on the Go button
Clear
Bookmark this SQL query:
[Delimiter ;] ✓ Show this query here again ☐ Retain query box
Go

```
CREATE DATABASE IF NOT EXISTS myDB
```

Creating Table (GUI)

- You can create tables in a selected DB by executing the relavent command or by using the GUI



No tables found in database.

1. Select the database

2. Type a suitable table name and number of columns for the table

Creating Table (GUI) .....

- Columns (Fields) of the table can be created by filling the subsequent form as requited.

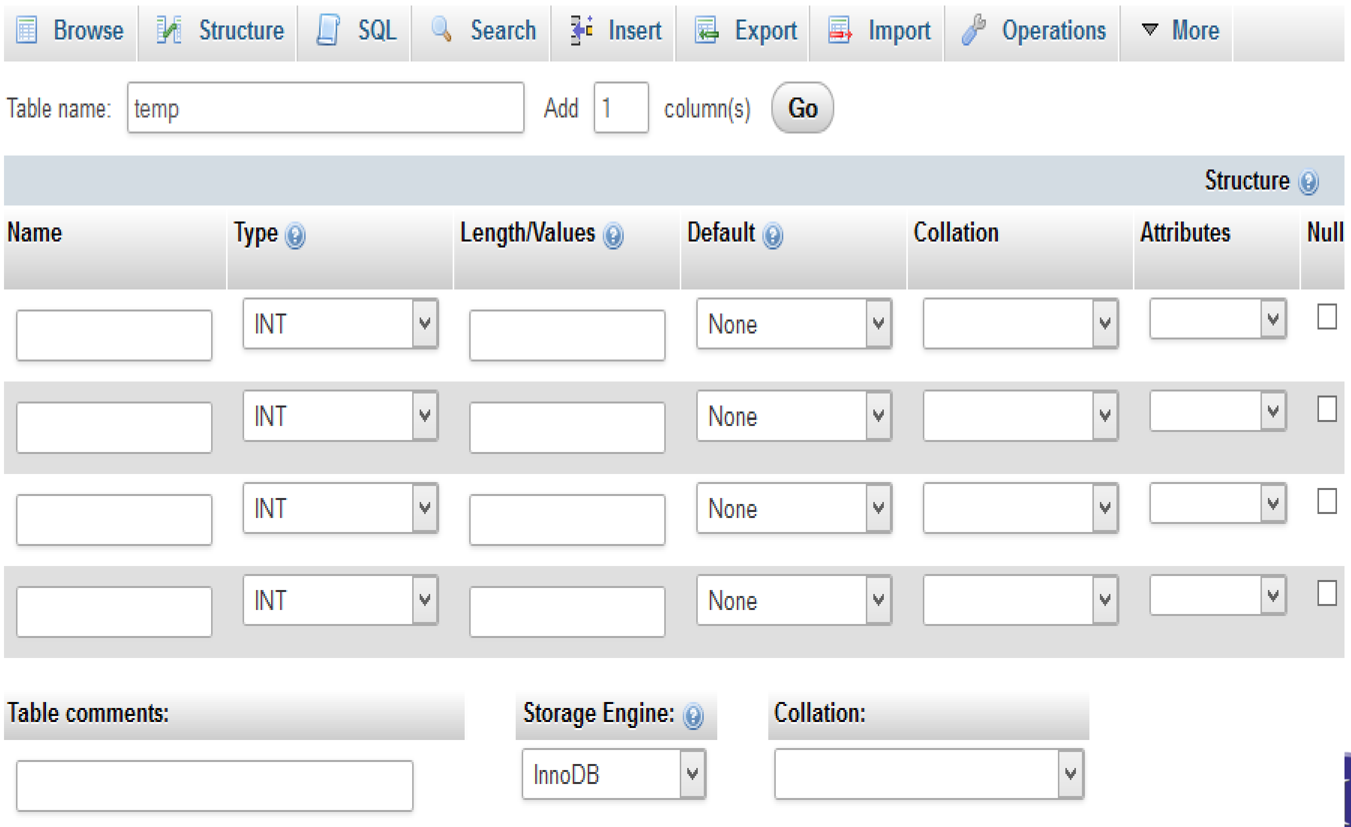


Table name: temp Add 1 column(s) Go

Structure

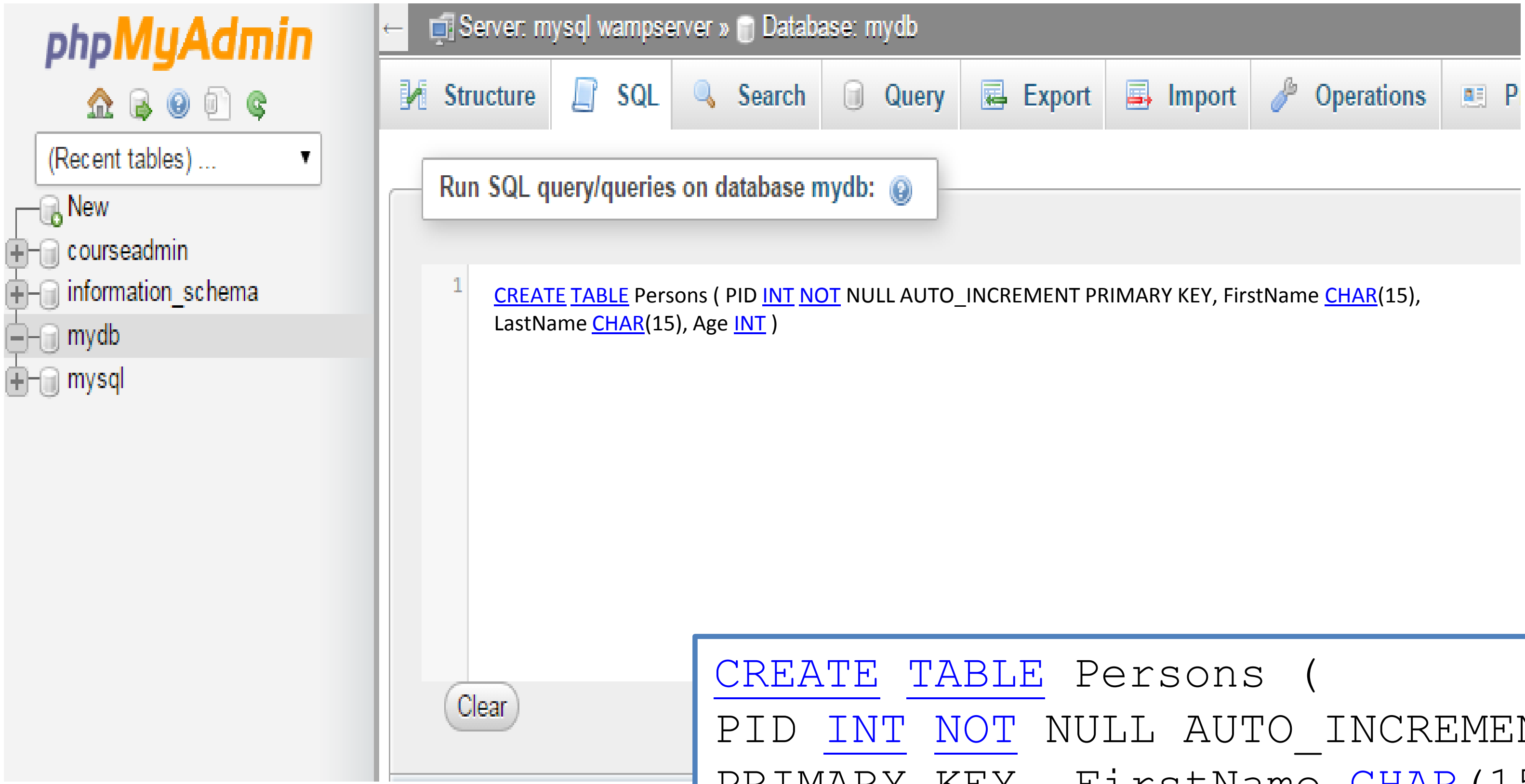
Name	Type	Length/Values	Default	Collation	Attributes	Null
	INT		None			☐
	INT		None			☐
	INT		None			☐
	INT		None			☐

Table comments:

Storage Engine: InnoDB

Collation:

Creating Tables (By SQL command)



```
CREATE TABLE Persons ( PID INT NOT NULL AUTO_INCREMENT PRIMARY KEY, FirstName CHAR(15),
LastName CHAR(15), Age INT )
```

```
CREATE TABLE Persons (
PID INT NOT NULL AUTO_INCREMENT
PRIMARY KEY, FirstName CHAR(15),
LastName CHAR(15),
Age INT )
```

Managing data stored in MySQL DBs Through PHP

Basic Steps in Processing data stored in MySQL through PHP programs

1. Connect to a host server with MySQL installed.
2. Select a database
3. Create a SQL statement
4. Execute the SQL statement.
   – Many SQL statements return the result of a SQL statement as a **record set**
5. Extract data from record set using PHP commands
6. Use the data as required
7. Close the connection

Open a Connection to MySQL

- Opening a connection to a MySQL DB

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "myDB";

// Create connection
$conn = new mysqli($servername, $username, $password,
$dbname);

// Check connection
if ($conn->connect_error) {
    die("Database connection failed: " . $conn-
>connect_error);
}
echo "Success. Connected to database";
?>
```

Usually "localhost"

By default – 'root'

By default – ' '

Close the Connection

- It’s always a best practice to close a connection once you are done with working with the database.
- Can close the connection using this syntax.

```
// if the connection object is
$conn
$conn->close();
```

mysqli_query()

- This is one of the most important and most used function in php when dealing with MySQL.
- mysqli_query() function is used to command PHP to execute a SQL statement.
- It sends a query or command to a MySQL DBMS through the connection object.

Inserting Data Into a Database Table

- You can use INSERT INTO statement to add new records to a database table.
- There are 2 different ways of writing insert quiaries
  - INSERT INTO table_name VALUES (value1, value2, value3,...)
  - INSERT INTO table_name (column1, column2, column3,...) VALUES (value1, value2, value3,...)

- The first form can be used if data is inserted to all columns of the new record.
- The second form can be used if data is inserted only to a selected set of columns in the new record.

Executing a SQL query through PHP

- The following PHP code segment inserts two record to the table ‘Persons’

```
<?php
$con=mysqli_connect("localhost", "root", " ",
"myDB");
if ($con->connect_error)
    die("Database connection failed: " .
             $conn->connect_error);

mysqli_query($con,"INSERT INTO Persons (FirstName,
                   LastName, Age)VALUES
                   ('Nimal', 'Perera',35)");

mysqli_query($con,"INSERT INTO Persons (FirstName,
                   LastName, Age)VALUES
                  ('Amal', 'Silva',33)");

mysqli_close($con);
?>
```

Structure of the table

Persons{

```
PID INT NOT NULL
AUTO_INCREMENT
    PRIMARY KEY,
FirstName CHAR(15),
LastName CHAR(15),
Age INT )
```

Inserting data to a MySQL DB through a HTML form.

- This HTML page requests the web server to execute a PHP script named “insert.php” at the server side.

```
<html>
<body>
<form action=“insert.php" method="post">
    Firstname: <input type="text“
name="firstname"><br>
    Lastname: <input type="text"
name="lastname"><br>
    Age: <input type="text"
name="age"><br>
    <input type="submit">
</form>
</body>
</html>
```

Insert Data Into a Database Table

Content of the PHP script insert.php

```
<?php
$con=mysqli_connect("localhost","root","", "myDB");
if ($con->connect_error) die("Database connection failed:
".$conn->connect_error);
$firstname = $_POST['firstname'];
$lastname = $_POST['lastname'];
$age = $_POST['age'];
$sql = "INSERT INTO persons (FirstName, LastName, Age)"
       . "VALUES ( '$firstname', '$lastname', $age )";
if(mysqli_query($con,$sql)){
    echo "Data inserted to the Table successfully";
}else {
    echo "Error in inserting data". $con->error;
}
mysqli_close($con);
?>
```

Selecting and Displaying Data

```
<?php
$con=mysqli_connect("localhost","root","",“myDB");
if ($con->connect_error) die("Database connection failed: " .
            $conn->connect_error);
$sql = "select * from persons";
$result = mysqli_query($con,$sql);
if(!$result){
    die("Error in executing the SQL" . $con->error);
}
while ($row = mysqli_fetch_array($result)){
    echo $row['FirstName'] . " " . $row['LastName']. "<br>";
}
mysqli_close($con);
?>
```

selects all data stored in the “persons" table and display only the content of the ‘FirstName’ and ‘LastName’ columns.

Select Data satisfying a where clause

- We can use the Where clause to filter records

```
<?php
$con=mysqli_connect("localhost","root","123456","bit
");
if ($con->connect_error) die("Database connection
failed: " .
            $conn->connect_error);
$sql = "select * from persons where
FirstName='Nimal'";
$result = mysqli_query($con,$sql);
if(!$result){
    die("Error in executing the SQL" . $con->error);
}
while ($row = mysqli_fetch_array($result)){
    echo $row['FirstName'] . " " . $row['LastName'].
"<br>";
}
mysqli_close($con);
?>
```

Earlier example selected all the Records from the table , but here we are using a where clause to filter data so that it will only return records where the First name field is ‘Nimal’

MySQL Update

- Whenever you need to update a record which exist in a table , you can use update query.

```
UPDATE table_name
SET column1=value,
column2=value2,...
WHERE some_column=some_value
```

Here the 'Where' clause decide which records to be updated. If you remove the WHERE clause, all records will be updated

Changing Data in the DB

```
<?php
$con=mysqli_connect("localhost","root","",“myDB"
);
if ($con->connect_error) die("Database
connection failed: " .
            $conn->connect_error);
if(mysqli_query($con,"UPDATE Persons SET Age= 50
WHERE FirstName='Nimal'")){
    echo "Record updated successful";
} else {
    echo "Error in executing the SQL" . $con-
>error;
}

mysqli_close($con);
?>
```

This will search for records which have the Firstname as ‘Nimal’ and change the Age attribute of those records to ‘50’

Delete Data In a Database Table

- The delete query is used when you need to remove a record in a table

```
DELETE FROM table_name
WHERE some_column = some_value
```

```
<?php
$con=mysqli_connect("localhost","root","123456","bit");
if ($con->connect_error) die("Database connection failed:
" .
            $conn->connect_error);
if(mysqli_query($con, "DELETE from Persons WHERE
FirstName='Nimal'")){
    echo "Record delete successful";
} else {
    echo "Error in executing the SQL" . $con->error;
}

mysqli_close($con);
?>
```

OOP using PHP

Object-Oriented Programming

- Object-oriented programming (OOP) refers to the creation of reusable software object-types / classes that can be efficiently developed and easily incorporated into multiple programs.
- In OOP an object represents an entity (a student, a desk, a button, a file, a text input area, a loan, a web page, a shopping cart).
- An object oriented application comprises of a collection of objects that interact with each other to solve a particular problem/s.

Object-Oriented Programming

- Objects are self-contained
  - data and operations that pertain to the object are assembled into a single entity.
- In OOP each Object has:
  - An identity
  - State
  - Behavior

Class and Object

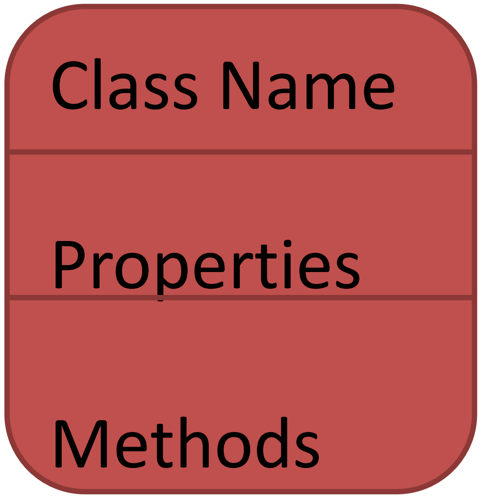
- A “Class” refers to a blueprint. It defines the attributes(variables) and methods the objects of that class should support.

- An “Object” is an instance of a class. Each object should corresponding to a class(es) which defines its attributes and behavior.

The Class

- The basic unit of code in object-oriented PHP is the class. A class provides a mechanism to encapsulate related functionality and data into a single entity.
- In PHP a class can be defined by using the keyword ‘**class**’ as below.
- The class name can be any valid label and it cannot be a PHP reserved word.

```
class Circle
{
// Class properties and
methods
}
```



Properties

- In PHP5, class properties are used as placeholders, to store data associated with the objects of that class. The visibility of a property can be defined by adding one of the following prefixes to the declaration of the property.
  - public : the value of the property can be accessed from everywhere. If no visibility is specified for a method, it defaults to public visibility.
  - protected : the value of the property can be accessed only by the class and the derived classes(child classes).
  - private : the value of the property can be accessed only by the class that defines the member.

A Class with Public and Private Properties - Example

```
class Person{
    public $name;
    public $sex = "m"; //
default value
    public $dob;
    private $bank_account_no;
}
```

Creating objects(Instances) of a class

- In order to access the properties and use the methods of a class, you first need to instantiate, or create an instance(object). This can be done by using the keyword ‘*new*’ as below:

  $c = new **`Person`**();

  Classes should be defined before instantiation.

  $c variable holds a **reference** to an instance (object) of the class ‘Person’.

  Once an object is created, the individual (visible) properties and methods of that object can be accessed by using an arrow (->) operator as given below.

$c->name = “Sunil”;

Object assignments

- When assigned an already created instance of a class to a new variable, the new variable also points to the same instance.

Example :

```
$p1 = new Person();
$p1->name = "Sunil";
$p2 = $p1                    // $p1 and $p2 points to the same object
$p2->name = "Kamal";
echo $p1->name; // This will print the text "Kamal" as $p1 and $p2 points
to the same object
```

Class Methods

- Class properties are used to hold data inside objects. Functions can be created inside a class to manage its property values. Such functions defined inside classes are called its methods.

Syntax for method definitions:

```
visibility function function_name
(parameters)
{
// method implementation here
}
```

Class Methods

```
class Person{
      public $name;
      public $sex = "m"; // default
value
      public $dob;
      private $bank_account_no = ;

Public function set_name($name){
      $this->name = $name;
}

Public function print_name(){
      echo $this->name;
}

}
```

$this is a The pseudo-variable. It is used to refer to the calling object to which the method belongs.

Constructors and Destructors

- In some situations when dealing with classes, you might want to have a way to automatically initialize object variables and/or to perform certain pre-defined actions when the object is created. For such situations, a constructor can be used.
- A constructor is nothing more than a specially named method that is automatically called when an object is instantiated. In PHP5, to implement a constructor, all you need to do is implement a method named   "__construct".

Constructors and Destructors

- PHP5 now includes a special method (destructor) that is called when an object is destroyed.
- An object destructor is called when all references to an object are removed, or it is manually destroyed in your code.
- To create a destructor, add a method to your class, and call it "**__destruct**".

```
Class Object
{
function   __construct()  {}
function   __destruct()  {}
}
$obj= newObject();
unset($obj);
```

Example

```
<?php
class Person{
    public $name = null;
    public $sex = "m";
    public $dob;
    private $bank_account_no;

    function __construct($name,$sex,$dob,$acc){
        $this->name = $name;  $this->sex = $sex;
        $this->dob = new DateTime($dob);
        //$dob should be give as "2015-01-15"
        $this->bank_account_no = $acc;
    }
    public function print_age($toData){
        //$toDate should be give as "2015-01-15"
        $interval =  $this->dob->diff(new DateTime($toDate));
        echo "Years - ". $interval->y . " Months - ".$interval->m ." Days -
".$interval->d ;
    }
}

$p1 = new Person("Saman","m","1960-11-23","123456");
$p1->print_age("2015-02-06");
?>
```

Static Keyword

- Sometimes when using object-oriented programming, you might need to assign properties/methods with a class rather than with its instances. This can be done by using static propertie/methods. Static properties/methods exist only with the class but not with its instances.
- Static Properties/methods of a class can be accessed by using the operator "::".

self::

- In order to access static variables within the same class, you can use the ***self*** keyword followed by the double-colon ("::")
- Using **self::** is similar to $this->, but it is used for static members only.

Example

```
<?php
class Person{
   public $name = null;
   public $sex = "m";
   private static $ObjectCount = 0;

   function __construct($name,$sex){
      $this->name = $name;
      $this->sex = $sex;
      self::$ObjectCount++;
   }

   public function print_object_count(){
      echo "Number of objects instantiated -".
self::$ObjectCount;
   }
}

$p1 = new Person("Saman","m");$p2 = new
Person("Kamala","f");
Person::print_object_count();
?>
```

Class Constants

- It is possible to define class values that are constant for the class. To define a class constant, use the “**const**” keyword before the constant name.
- Constants differ from normal variables in that you don't use the $ symbol in their declaration.

```
<?php
class Person{
    const office = "UCSC";
    public $name = null;
    public $sex = "m";

    function
__construct($name,$sex){
        $this->name = $name;
        $this->sex = $sex;
    }

    public function print_office(){
        echo "Office name -".
self::office;
    }
}

Person::print_office();
?>
```

Example

Inheritance

- Allows you to define a base set of properties and methods that belong to a base class and to extend that class by
  - adding additional properties and methods and/or
  - changing the behavior of existing methods.
- The subclass inherits all of the public and protected properties and methods from the parent class. Unless a subclass overrides a method, the subclass retains its original functionality defined in the parent class.
- Inheritance facilitate the implementation of additional functionality in similar objects without the need of re-implementing all of the shared functionality.
- When defining a subclass the parent class must be defined before defining the child class.

The extends key word

```
<?php
   class Shape
   {
   public $center =
   array(x=>0, y=>0);
   }

   class Circle extends Shape
   {
   public $radius;
   }

   $c = new Circle();
   print_r($c->center);
?>
```

The keyword **extends** is used to build a subclass

Parent Class

The final Keyword

- There are cases where, you want to restrict a subclass from redefining a member that exists in a parent class. You can prevent properties and methods from being redefined(overriding) in a subclass by using the **final** keyword.

Using parent:: References

- In some situations you may want to refer to a property or a method of the parent class, in a subclass. To achieve this, you can use the parent keyword in conjunction with the :: (double colon) you saw in the previous section on static members.

```
<?php
class Shape {
    var $x;
    function getName()
    {
        $this->x = “I’m a shape";
        return;
    }
}
class Circle extends Shape {
    // we have var $x; from the parent already here.
    function getParentName()
    {
        parent:: getName();
        echo $this->x;
    }
}
$b = new  Circle();
$b-> getParentName(); // prints: " I’m a shape "
?>
```

Abstract Classes

- When a class is defined as abstract, other classes can extend it, but it cannot be instantiated. This feature enables you to define classes as templates.
- A class that contains at least one abstract method is treated as an abstract class.
- Abstract methods only defines the signature of the method, but not its implementation.
- When inheriting from an abstract class, all methods declared as abstract in the parent class must be defined by the child.

```php
<?php
abstract class Shape
{
public $origin = array(x=>0,
  y=>0);
}

class Circle extends Shape
{
// Circle implementation
}

$c = new Circle();
echo $c->origin;
$s = new Shape();
echo $s->origin;
?>
```

Interfaces

- Another new object-oriented feature in PHP5 is the ability to create and use interfaces. Interfaces, in a nutshell, are a way to specify what methods a class must explicitly implement. This is useful when dealing with many interconnected objects that rely on the specific methods of one another.
- In PHP5, an interface is defined using the **interface** keyword, and implemented using the **implements** keyword.
- All methods declared in an interface must be public.
- Interfaces can be extended like classes using the **extends** operator.

Interfaces

```
interface TwoDimensionalOperations
{
	public calculateArea() ;
}
class Circle implements
TwoDimensionalOperations
{
	public calculateArea() ;
	{
	// Implementation of calculateArea,
	specific to this Circle class
	}
}
```

Abstract Classes Vs Interfaces

- A child class can extend only one abstract class, whereas a class can implement multiple interfaces.
- An interface does not provide any functionality (method implementations) whereas an abstract class may provide some functionality.

Magic Methods

- Magic methods is a set of methods designed to be executed automatically in response to particular PHP events.
- All names of magic methods starting with two underscores.
- PHP reserves all function names starting with “__” as magical, thus it is recommended not to start any user defined function with “__”.

Eg:

- __call()
- __get and __set
- __toString

__call()

- Allows you to provide actions or return values when undefined methods are called on an object.
- Can be used to simulate method overloading, or even to provide smooth error handling when an undefined method is called on an object.

```
public function __call($m, $a){
echo “The method ” . $m . “ was called.<BR> The arguments
were as follows:<BR>;
print_r($a);
}
```

__get and __set

- __get makes properties which actually don't exist in a class to appear as if they do.
- __get takes one argument
  - the name of the property,

- __set requires two:
  - the name of the property and the new value.

__toString

- __toString returns a custom string value that is automatically used when the object is converted to a string.
- Only called when used directly with echo or print. If not implemented in a class the object id will be returned by default.

PHP Frameworks

What is a framework ?

- A software framework is a re-usable design that can be used to build a software system (or subsystem).

A framework for a house

A framework for a house is a structure that engineers use to build a house . Likewise ,a software framework is a structure with core functionalities and data structures that enable developers to build their applications with ease.

Allows developers to develop applications faster

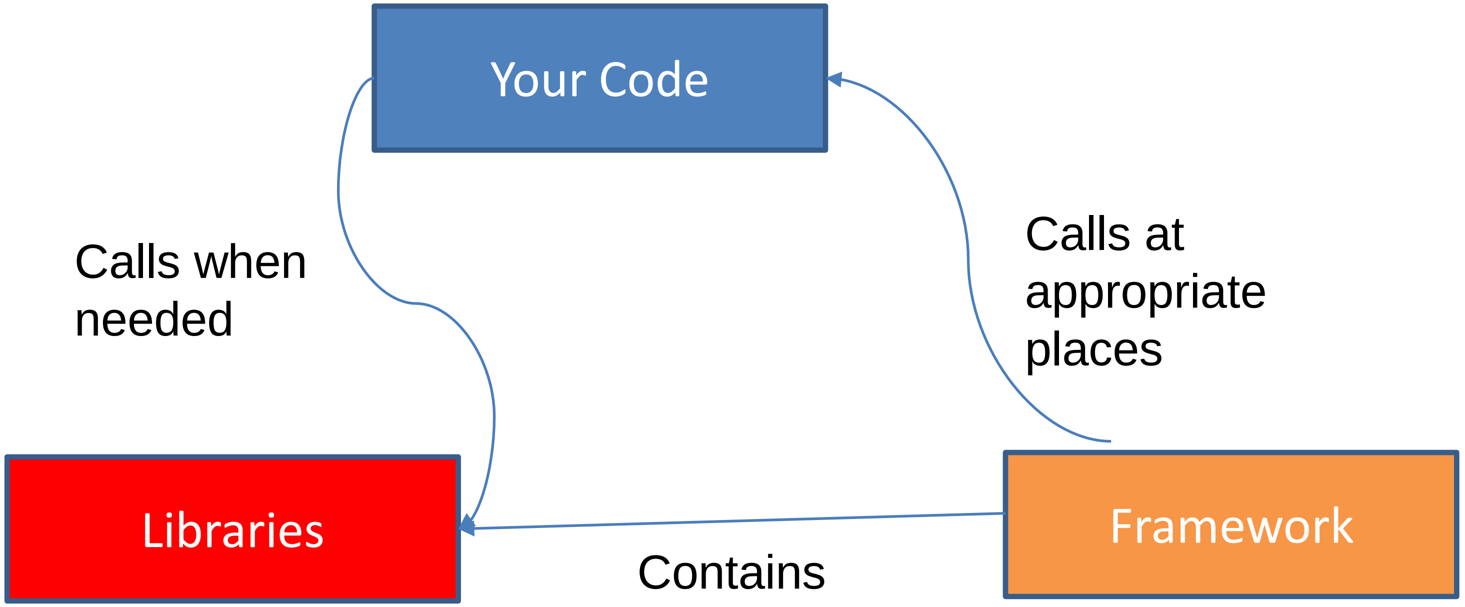
What is a framework ?

- Frameworks support for the development of large-scale software systems. They provide **higher productivity** and **shorter** development time through **design** and **code** reuse.
- Software frameworks include support programs, compilers, code libraries, tool sets, and application programming interfaces (APIs) that bring together all the different components to enable development of a project or solution.

Library vs. Framework

- A **library** performs specific, well-defined operations whereas a **framework** is a skeleton (abstract design) where the application defines what exactly to be done by filling out the skeleton.
- The main objective of a library is the code reuse.
- Typically, in a framework there is a defined control flow with predefined spots that you should fill out with our code. Your inserted code will be called by the framework appropriately.

Library vs. Framework



Why Frameworks ?

- Raw PHP, works very well with small applications. HTML files can be easily extended with dynamic content from the database, form processing, etc.
- But , when applications grows, lots of code repetition occurs across multiple pages.
- Its hard for a new developer to work on a code someone else has written.
  - It takes a long time to get familiar with the code.

Model-View-Controller design pattern

- Most common and popular Web application development frameworks are based on the Model-View-Controller design pattern.
- Typically, application frameworks provide basic building blocks needed by most applications such as
  - Database connections
  - Business logic
  - Form handling

Features of a good framework

- Supports a design pattern
- Provide libraries , plugins to make application development easier.
- Support abstract layer for database interactions
  - Ability to work with a database without writing queries
- A strong community
  - If something goes wrong , a place to get support.

PHP Frameworks

There are many PHP framework. A number of them are listed below

- CakePHP
- Symfony
- CodeIgniter
- Zend Framework
- Yii Framework

In this course emphasis is given on the CodeIgniter framework.