Unit 10 PHP-Part – 2

Structure:

10.1 Introduction

In previous unit, you have studied PHP basics programming concepts like variables, operators, control structures; those concepts help you to write programs. One of the PHP most significant features is its wide ranging support for database.

In this unit you will learn the concepts likes,object oriented programming,defining

PHP classes and creating objects, constructor and destructor, inheritance and MYSQL data base concept

Objectives:

After studying this unit, you will be able to:

- describe Concepts of OOP
- define Access modifiers
- describe data base design concepts
- define MYSQL API

10.1.1 Introduction to Object Oriented Programming

OOP stands for Object Oriented Programming. OOP is a programming paradigm wherein you create “objects” to work with. These objects can then be tailored to your specific needs, to serve different types of applications while maintaining the same code base.

Starting with PHP5, the object model was rewritten to allow for better performance and more features. This was a major change from PHP4. PHP 5 has a full object model. Among the features in PHP5 are the inclusions of visibility, abstract and final classes and methods, additional magic methods, interfaces, cloning and type hinting. PHP treats objects in the same way as references or handles, meaning that each variable contains an object reference rather than a copy of the entire object.

An object is simply a copy or instance of a “class”. A class can be defined as a “black box” from where we create our objects and access its attributes (variables) and methods (functions).

10.1.2 Basic Concepts of OOP

The key terms in Object-Oriented Programming.

- **Class** − this data type was created by the programmer and it supports both local functions and local data. A class can be viewed as a template for creating numerous instances of a particular kind (or class) of object.
- **Object** − a unique instance of a class-defined data structure. You can create multiple objects that are members of a class by first defining it. Objects are also referred to as instances.
- **Member Variable** − these are the variables that a class has defined. The class's member functions can be used to access this data, which will be hidden from the outside world. As an object is constructed, these variables are referred to as the object's attributes.

- **Member function** − these functions, which are used to access object data, are defined inside of classes.
- **Inheritance** − Inheritance is the process by which existing functions from a parent class are used to define a new class. Here, a child class may inherit all of a parent class member functions and variables, or only some of them.
- **Parent class** − a class from which another class derives. This is also referred to as a super class or base class.
- **Child Class** − a class that has ancestry in another class. This is referred to as a derived class or subclass.
- **Polymorphism** − the same function can be used for several purposes under this object-oriented notion. For instance, the function name will not change, but it will accept varying numbers of parameters and do various tasks.
- **Overloading** − a sort of polymorphism in which some or all of operators have various implementations based on the types of their arguments. Similar to variables, functions can be overloaded with many implementations.
- **Data Abstraction** − any data representation where the implementation details are concealed
- **Encapsulation** − refers to a technique whereby we create an object by encapsulating all of the data and member functions.
- **Constructor** -When a new object is created from a class, a special kind of function called will be automatically invoked.
- **Destructor** − refers to a unique kind of function that is automatically called if an item is removed or leaves its scope.

10.1.3 Defining PHP Classes

A class can be viewed as a template for creating numerous instances of a particular kind (or class) of object.

The general PHP definition form for a new class is as follows:

```
<?php
  class phpClass {
    var $var1;
    var $var2 = "constant string";

    function myfunc ($arg1, $arg2) {
      ….
    }
    …
  }
?>
```

The explanation for each line is as follows:

- The name of the class you want to define, followed by the special form class.
- A group of brackets that enclose a variable declaration or a function definition in any number.

- Variable declarations begin with the special form var, which is followed by the name of a regular $ variable. They may also contain an initial assignment to a constant value.
- Function definitions, which are used to set and retrieve object data, resemble standalone PHP functions but are local to the class.

Example:

```
<?php
  class Books {
    /* Member variables */
    var $price;
    var $title;

    /* Member functions */
    function setPrice($par){
      $this->price = $par;
    }

    function getPrice(){
      echo $this->price ."<br/>";
    }

    function setTitle($par){
      $this->title = $par;
    }

    function getTitle(){
      echo $this->title ." <br/>";
    }
  }
?>
```

10.1.4 Creating Objects in PHP

Once we defined our class, then we can create as many objects as we like of that class type.

Example how to create object using new operator.

```
$mercedes = new Car;

$ford = new Car;

$ferrari = new Car;
```

Note: Here we have created three objects and these objects are independent of each other and they will have their existence separately.

Example:

```
<?php
class Car

{
public function drive(){
echo "driving a car<br/>";
}}
$car=new Car();
$car->drive();
Car::drive();
?>
```

Output:

```
Driving a car driving a car
```

Note: In the above programming we create a class named Car, it has a publically declared function drive(), on the outer side of the class we instantiate the object $car, we call the function drive with - > operator. Any class can also access it's member using:: operator.

Using objects, their properties and methods

As we discussed, a class is a meaningful grouping of various attributes and methods that help us organize our data in a better manner. When we create a class, we store the essential parameters, as properties or attributes of the class. The same parameters can be access directly or through methods and functions of the class.

Example shows how to set title and prices for the three books by calling member functions.

```
$physics->setTitle( "Physics for High School" );
$chemistry->setTitle( "Advanced Chemistry" );
$maths->setTitle( "Algebra" );
$physics->setPrice( 10 );
$chemistry->setPrice( 15 );
$maths->setPrice( 7 );
```

The values set by the set APIs can be retrieved by another simila API:

```
$physics->getTitle();
```

```
$chemistry->getTitle();
$maths->getTitle();
$physics->getPrice();
$chemistry->getPrice();
$maths->getPrice();
```

Output:

```
Physics for High School
Advanced Chemistry
Algebra
10
15
7
```

Note: It is always advisable to access the class properties through functions and not directly, as it offers more flexibility. In the above example, suppose we store the “price” attribute as “price in rupees”, then it would be very difficult for us to show it in “dollars” at a later stage. However, if the same attribute is accessed through a function, then we can always put the conversion logic in the “function” while keeping our base attributes unchanged.

10.1.5 Constructor & Destructor

Constructor

The Constructor functions are a particular class of functions that are automatically called whenever an object is created. By initialising a lot of items using function. PHP has a unique function named __construct(). The function accepts any number of inputs.

Example: the construct function to the Books class and initialise the book's price and title when the object is created.

```
function __construct( $par1, $par2 ) {
   $this->title = $par1;
   $this->price = $par2;
}
```

Now, setting the price and title only requires one call to the set method. These two member variables can only be initialised when an object is created.

```
$physics = new Books( "Physics ", 10 );
$physics->getTitle();
```

$physics->getPrice();

Destructor

The function __**destruct()** can be used to define a destructor function, just as a constructor function . All of the resources can be released.

10.1.6 Access Modifiers

Access modifiers are keywords that let you manage a class's fields, methods, and function visibility. public, protected, and private are the three types of access modifiers.

Public:

A class's methods and properties are public until you indicate otherwise. They can therefore be accessed in one of three ways:

They can therefore be accessed in one of three different ways:

- From outside the class in which they are declared.
- From inside the class in which they are declared.
- From within a different class that uses the declared class's implementation

All members have been visible to the public.

Private:

You can restrict a member's accessibility to the class in which it is declared by making it private. The private member cannot be accessed from outside the class and cannot be referenced from classes that inherit the class in which it is declared.

By using the private keyword in front of a class member, the member can be made private.

Protected:

In the class in which it is declared, as well as in classes that extend that class, a protected property or method is accessible. Outside of these two categories of classes, protected members are not accessible. The protected keyword should be placed in front of the class member to make it protected.

Example:

```
<?php

class Fruit       // classname

{
```

```
Public   $name;          // properties

Protected   $color;

Private   $weight;

}

$apple = new Fruit(  );

$apple -> name = ‘Apple’;     //ok

$apple ->color = ‘red’;        //error

$apple->weight=‘1kg’;         //error

?>
```

The name can be accessed as it is declared public. color and weight cannot be accessed as its declared as protected and private.

10.1.7 Inheritance

Inheritance is the process by which existing functions from a parent class are used to define a new class. Here, a child class may inherit all of a parent class' member functions and variables, or only some of them. The **extends** clause in PHP allows class definitions to potentially derive from a parent class definition.

Syntax:

```
class Child extends Parent {
  <definition body>
}
```

The following qualities of the child class (or subclass or derived class) are a result of inheritance:

- Has all of the parent class's member variable declarations by default.
- Contains all of the parent's member functions automatically. These functions will operate (by default) in the same manner as they do in the parent.

Example:

```
class Novel extends Books {
  var $publisher;

  function setPublisher($par){
    $this->publisher = $par;
  }
  function getPublisher(){
    echo $this->publisher. "<br />";
  }
}
```

10.1.8 Abstract Class

A class that is abstract can only be inherited, not created. The word "abstract" is used to declare an abstract class. All methods designated as abstract in the parent's class declaration must be specified by the child when deriving from an abstract class, and they must also be defined with the same visibility.

The following rules to be applied when a child class inherits from an abstract class:

- The child class method must redeclare the parent abstract method and have the same name as the class.
- The access modifier for the child class method needs to be the same or less restrictive.
- The required number of arguments must remain constant. However, the child class could additionally have optional parameters.

Example:

```
<? php

abstract class  parent          // abstract class

{

abstract public function method1(  );                   //abstract property

abstract public function method2($name);

}

}

class child extends parent              //child class derive parent class

{

Public function method1( )
```

```
{

Echo “this is derived from abstract class”;    //implementation of abstract class method

}

}

$class = new child();

echo $class->method1;

?>
```

According to the above example the parent class contains the abstract methods which is declared without definition. The child class inherits the parent class and the definition is specified in the child class.

10.1.9 Function Overriding

The Function Overriding is similar to other OOPs programming languages. Both the parent and child classes should share the same function name, arguments, and parameters when overriding a function. In a child class, it is used to replace the parent method. Overriding is used to modify the behavior of parent class methods.

Example:

```
<?php

  class Base {

    function display() {

      echo " Base class function declared finally";

    }

    function demo() {

      echo "Base class function!";

    }
  }
  class Derived extends Base {
    function demo() {
      echo "Derived class function!";
    }
  }
  $ob = new Base;
  $ob->demo();
  $ob->display();
  $ob2 = new Derived;
  $ob2->demo();
  $ob2->display();
?>
```

In the derived class the function demo() is overridden when the derived object calls the function overridden method will be executed.

Self Assessment Questions

1. _______ is the name of the class that the child class derives from.
2. The act of hiding an application's true internal workings from the user by using well-known user interfaces is referred to as____ .
3. The Constructors are identified by the name ________ in PHP.

10.2 Backend database named MySQL

A database is a place where we can store our data whatever the data might be. It might be a text, number, file, date, time, audio, video, etc.

Typically a database is made up of many linked tables of rows and columns, each containing specific data (E.g. name, address, location, size, weight, dates). The data stored in a table is organized into rows and columns. Each row in a table represents an individual record and each column represents a field.

Database Management system is a collection of programs that enables you to store, modify, and extract information from a database. There are many different types of DBMS, ranging from small systems that run on personal computers to huge systems that run on mainframes.

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model introduced by E. F. Codd and most popular databases currently in use are based on the relational database model.

MySQL is an open source Relational Database Management System developed by Michael Widenius and David Axmark in 1994. MySQL is very popular when it comes to Web development. It is widely-used as the database component of LAMP web application software stack.

SQL stands for “structured Query Language”, and is just a definition of a computer language that can be used to access data stored in relation database. SQL is not itself a standalone computer application, but is rather part of many relation database programs such as Oracle, MYSQL, Sybase, etc.

All of these RDBMS uses SQL as their language. Each of them has minor variations in the “dialect” of SQL that they use, but it’s all still SQL.

MySQL is a powerful Relational Database Management System (RDBMS) which we will use to learn the basic principles of database and data manipulation using Structured Query Language (SQL) statements. SQL is a database language that is used to retrieve, insert, delete and update stored data. This is achieved by constructing conditional statements that conform to a specific syntax (i.e. the strict order required of elements for a statement to work).

10.2.1 MySQL History

A Swidish business named MySQL AB, which was established in 1995, developed MySQL. Sun Microsystems later bought the business for roughly $1 billion in 2008.The first version of MySQL, known as UNIREG, was created internally in 1979 as a database management tool. The inventor of

UNIREG is Michael Widenius. After UNIREG was developed, it was rewritten in a number of other languages to make it capable of handling large databases.

mSQL and UNIREG were combined to produce MySQL, which was given the name "My" after Michael Widenius' daughter. In order to enable indexing in mSQL, Michael Widenius and David Hughes, the creators of mSQL, worked together to connect mSQL to UNIREG's B+ ISAM handler.

10.2.2 MySQL Features

SQL (Structured Query Language) queries are the foundation of the relational database management system (RDBMS) known as MySQL. It is one of the most widely used languages for organizing and gaining access to table records. Under the terms of the GNU license, MySQL is free and open-source software.

The most significant characteristics of MySQL are as follows:

- **Relational Database Management System (RDBMS):** A relational database management system is MySQL. To access and manage the table's records, this database language is based on SQL queries.
- **Easy to use:** it's simple to use MySQL. We merely need to learn the fundamentals of SQL. With just a few basic SQL lines, we can create and use MySQL.
- **Secure:** MySQL has a strong data security layer that shields private information from outsiders. Moreover, MySQL encrypts passwords.
- **High performance:** the special storage engine architecture, MySQL is more dependable, quicker, and less expensive. When compared to other databases, it offers results that are extremely high-performance without sacrificing a crucial piece of the software's functionality. The separate cache memory allows for quick loading of tools.
- **Client/ Server Architecture:** MySQL follows the working of a client/server architecture. There is a database server (MySQL) and arbitrarily many clients application programs, which communicate with the server; that is, they can query data, save changes, etc.
- **Free to download:** We can download MySQL from the official website for free because it is free to use.
- **Scalable:** MySQL enables multi-threading, scaling is simple. It can manage practically any volume of data, up to 50 million rows or more. A 4 GB maximum file size is the standard.
- **Compatible on many operating systems:** Several operating systems, including Novell NetWare, Windows, Linux, numerous UNIX and others are compatible with MySQL. Moreover, MySQL offers the option for clients to execute either on the same machine as the server or on a different machine.

10.2.3 Benefits of MySQL

- **Open Source**

  In terms of software as a service, MySQL is among the most widely used options for companies or organisations. The community edition, which provides improved speed, scale, and dependability, is openly accessible for usage and modification by anybody. This can be quite advantageous, particularly for companies trying to avoid paying licence fees.
  Developers can adapt their programme to meet their particular requirements because the source code is open for anybody to read and alter. Businesses with specific needs or those that need to link the programme with other tools or systems can profit from this flexibility.

- **Data Security**

  The world's safest database management system is MySQL. Any firm can greatly benefit from the data security and transactional processing capability provided by the most recent version of MySQL, especially e-commerce companies that conduct lots of financial transactions.

- **Scalability on Demand**

  The distinguishing characteristic of MySQL is scalability on demand. Even in terabyte-sized databases, it manages deeply integrated applications with a small footprint. Also, for e-commerce businesses with unique database needs, MySQL provides tailored solutions.

- **Higher Efficiency**

  It's a good idea to have a backup plan in case something goes wrong, but it's not always possible. It enables system administrators to flawlessly configure the MySQL database server. It makes no difference if it is a high-speed transactional processing system or an e-commerce web application that receives a million queries per day.In addition to ensuring full-text indexes, optimum speed, and different caches for increased performance, MySQL was designed to handle the rising demands of virtually every application.

- **24×7 Server Uptime**

  MySQL promises continuous availability. Also, a vast diversity of high-availability database solutions are provided, such as customised server clusters and master/slave replication configurations.

- **Complete Transactional Support**

  The world's leading transactional database engine is MySQL. It has unconstrained row-level locking in addition to complete atomic, consistent, isolated, durable, and multi-version transaction capabilities. These distinctive attributes make MySQL a one-stop shop for comprehensive data integrity that guarantees immediate deadlock identification through server-enforced referential integrity.

- **Comprehensive Workflow Control**

  MySQL is user-friendly and often takes less than 30 minutes to download and install. No matter what operating system you use—Windows, Mac OS X, Linux, or UNIX—MySQL is a complete solution with self-management tools.

- **Lower Total Cost of Ownership**

  Businesses can significantly reduce their total cost of ownership when they switch from their existing database applications to MySQL. They might spend less on brand-new projects. Because of MySQL's dependability and manageability, you can avoid spending money and effort on troubleshooting in order to fix downtimes and performance problems.

10.2.4 Drawbacks of MySQL

The following are a few drawbacks of MySQL:

- ROLE, COMMIT, and stored procedures are not supported by MySQL versions below than 5.0.
- A extremely high database size is not supported by MySQL as effectively.
- MySQL is susceptible to data corruption and does not manage transactions particularly effectively.
- MySQL is criticized for lacking an effective debugging and development tool in comparison to commercial databases.
- MySQL is not compatible with SQL check limitations.

10.2.5 How MySQL works?

MySQL has a client-server architecture and may be utilised in a networked setting, like the majority of database management systems available today. The database files are kept on the same physical or virtual system as the server programme, which controls all communications with the databases.

The server can be accessed by a variety of client programmes, including MySQL tools for database administration and any programmes created in other programming languages. Requests from clients are handled by the server, which then sends the responses back to the client.

A client can communicate with a server via using a network or internet connection, or it can be located on a different host and send database requests there. The MySQL server must be functioning for clients to connect to it, which is important.

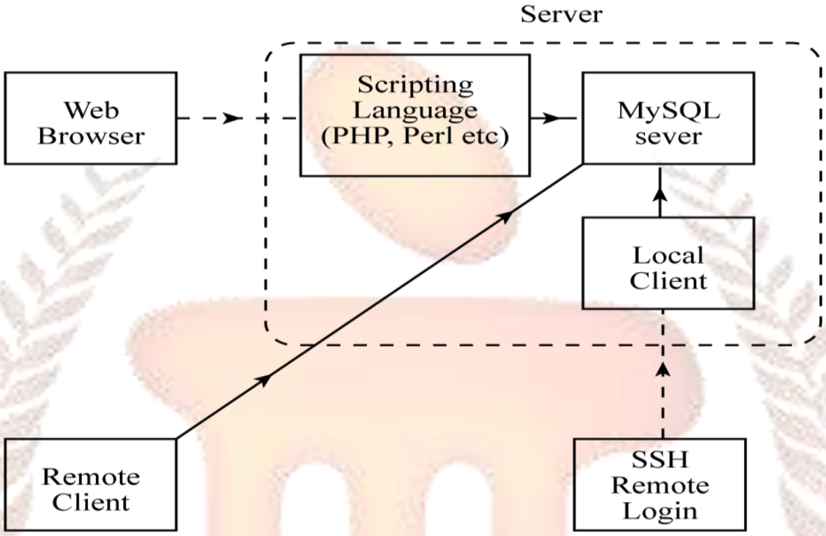

Figure10.1 How MySQL works?

10.2.6 Database Design Concepts

SQL supports many different types of statements and the most of these statements include a number of options. SQL is generally broken down into three categories of statements: data definition languages (DDL), data manipulation language (DML), and data control language (DCL).

- **Using DDL Statements**

In MYSQL, DDL statements create, alter, and delete data structures within the database. DDL statements define the structure of a MySQL database and determine the type of data that can be stored in the database and how to store that data. Specially, DDL statements allow you to do following:

- Create and remove database (the create database and drop database statements)
- Create, alter, rename, and remove tables (the create table, alter table, rename table, and drop table)

- **Using DML statements**

Unlike DDL statements, DML statements are more concerned with the data stored in the database than the database structure itself. For example, you can use DDL statements to request information from the database as well as insert, update, and delete data. Especially DML statements allow you to do the following:

- Query a database for specific types of information from one or more tables (the SELECT statement).
- Insert data into table (the Insert, Replace and Load data in file statement)
- Update existing data in a table (the Insert and Replace statement)
- Delete data from table (the delete from and truncate table statements)

➢ **Using DCL statements**

As you learned above, DDL statements allow you to define the structure of a MYSQL database, and DML statements allow you to access and manipulate data within the database. DCL statements represent yet another function supported by SQL statements: controlling access to database. Specially, DCL statements allow you to do the following:

- Grant access privileges to users (the Grant statement)
- Revoke access privileges to users (the Revoke statement)

10.2.7 Connecting to MySQL

PHP has a lot of inbuilt functions you can use to manipulate databases. In PHP version 5, a lot more were added as well! Here, we'll stay with the inbuilt functions for versions earlier than PHP 5. But if you have version 5,it's well worth researching the newer database functions. A good place to

start is php.net. To open our Address Book database, we'll use the following inbuilt functions:

- **mysql_connect()**
- **mysql_select_db()**
- **mysql_close()**

The approached we'll take has three steps:

1. Open a connection to MySQL itself
2. Specify the database we want to open
3. Close the connection

Step 1 – Open a connection to MySQL

The first job is to actually connect to MySQL. As its name suggests, **mysql_connect( )** does exactly that. Here's the code we're going to be using.

Example:

```
<?PHP

$user_name = "root";
```

```
 $password = ""; $database = "addressbook";
$server = "127.0.0.1";
mysql_connect($server, $user_name, $password);
print "Connection to the Server opened";
?>
```

Save your work and try it out on your server (this assumes that you have the Address Book database created, and that it is in the data folder of MySQL. If you don't, you can download all the files here.).

The first four lines are just setting up variables, and putting something in them

```
$user_name = "root"; $password = ""; $database = "addressbook"; $server = "127.0.0.1";
```

The username we're trying here is "root" and the password is blank. These are the MySQL defaults. You don't need to change these, in most cases.

Hopefully, you won't have any errors. But the line that connects to MySQL is this:

```
mysql_connect($server, $user_name, $password);
```

So you type the name of the function first (mysql_connect), followed by the round brackets. In between the round brackets, you need three things: the name of your server, your MySQL username, and your MySQL password. These can be entered directly, like this:

```
mysql_connect( '127.0.0.1', 'root', '' );
```

Or as variables, like we did at first:

```
$user_name = "root";
$password = ""; $server = "127.0.0.1";
mysql_connect($server, $user_name, $password);
```

And that's all you need to get you connected to MySQL. But we haven't connected to the database yet. That's Step 2 on our slant.

Step 2 - Specify the database you want to open

In our code, we set up a variable with the name of our database: **$database = "addressbook";**

Example:

```
$user_name = "root"; $password = ""; $database = "addressbook"; $server = "127.0.0.1";
```

```
mysql_connect($server, $user_name, $password);
$db_found = mysql_select_db($database);
print "Connection to the Server opened";
```

You use the **mysql_select_db( )** function to specify which database you want to open. The function then returns a true/false value. If it finds your database, a value of true is returned; if your database can't be found then a value of false is returned. You can use some logic to test if the database was found.

Example:

```
$db_found = mysql_select_db($database);
if ($db_found) {
print "Database Found";
} else {
print "Database NOT Found";
}
```

Now change the database name from this:

```
$database = "addressbook";
```

to something like this:

```
$database = "addressbook2";
```

Run your code again, and you should see Database NOT Found printed out (unless you have a database called addressbook2). Change the database name back to address book.

But there's another option you can use for mysql_select_db -something called a resource link identifier. It's just a file handle that you used in an earlier section (opening text files). You use it like this:

Example:

```
$user_name = "root"; $password = ""; $database = "addressbook"; $server = "127.0.0.1";
$db_handle = mysql_connect($server, $user_name, $password);
$db_found = mysql_select_db($database, $db_handle);
if ($db_found) {
print "Database Found " . $db_handle;
} else {
print "Database NOT Found " . $db_handle;
}
```

So when we connect to the database, we're now using this:

```
$db_handle = mysql_connect($server, $user_name, $password);
```

It's just the same as before, except we're returning a value from the mysql_connect function, and putting it into a variable called **$db_handle**. When we connect to the database, we can use this file handle:

```
$db_found = mysql_select_db($database, $db_handle);
```

The resource link identifier (file handle) goes after the name of the database you want to open. You can then use this file handle to refer to your database connection. Now that we've connected to MySQL, and connected to a database, it's time to close the connection.

Step 3 – Close the connection

Closing a connection to a database is quite easy. If you've used a file handle, as above, you just do this:

```
mysql_close( $db_handle );
```

Otherwise, you don't need to bother. It's recommended that you take the file handle approach, though. That's what we'll be doing from now on.

So, we'll add a line to close our connection. Here what your code should now look like:

Example:

```
<?PHP
$user_name = "root"; $password = ""; $database = "addressbook"; $server
= "127.0.0.1";
$db_handle = mysql_connect($server, $user_name, $password);
$db_found = mysql_select_db($database, $db_handle);
if ($db_found) {
print "Database Found "; mysql_close($db_handle);
} else {
print "Database NOT Found ";
}
?>
```

Now that we've got a connection to the database, it's time to look at how you can access the data in the database.

When the PHP script and MySQL are on the same machine, we can use localhost as the address we wish to connect to. Localhost is a shortcut to just have the machine connect to itself. If our MySQL service is running at a separate location we will need to insert the IP address or URL in place of localhost.

10.2.8 PHP & MySQL Code:

Before we can do anything with MySQL in PHP we must first establish a connection to our web host's MySQL database. This is done with the MySQL connect function.

Example:

```
<?php
mysql_connect("localhost", "admin", "1admin") or die(mysql_error());
echo "Connected to MySQL<br />";
?>
```

The above code if succeeded, will display like-

Output: Connected to MySQL

Note: The mysql_connect function takes three arguments -- Server, username, and password. In our example above these arguments were:

Server - localhost

Username - admin

Password - ******

The "or die(mysql..." code displays an error message in our browser if -- you've probably guessed it -- there is an error in processing the connection!

Choosing the working Database

After establishing a MySQL connection with the code above, we then need to choose which database we will be using with this connection. This is done with the mysql_select_db function.

PHP & MySQL Code:

```
<?php
mysql_connect("localhost", "admin", "1admin") or die(mysql_error());
echo "Connected to MySQL<br />";
mysql_select_db("test") or die(mysql_error());
echo "Connected to Database";
?>
```

10.2.9 MYSQL API

There are several inbuilt PHP functions which are specified for MYSQL. Below table gives the list of them.

Function	**Description**
Mysql_affected_rows()	Returns the number of affected rows in the previous MySQL operation
mysql_change_u ser()	Changes the user of the current MySQL connection
mysql_create_db( )	Creates a new MySQL database. Use mysql_query() instead
Mysql_error	Returns the error description of the last MySQL operation
Mysql_fetch_array	Returns a row from a recordset as an associative array and/or a numeric array

Mysql_field_name()	Returns the name of a field in a recordset
Mysql_field_seek()	Moves the result pointer to a specified field
Mysql_field_type()	Returns the type of field in a recordset
Mysql_insert_id()	Returns the AUTO_INCREMENT ID generated from the previous INSERT operation
Mysql_list_dbs()	Lists available database on a MySQL server
Mysql_list_process()	Lists MYSQL processes

Self Assessment Questions

4. ____________ function, returns the number of affected rows in the previous MySQL operation.
5. Is it required to use a semicolon after each query?(True/False)
6. SQL command falls under the category of ALTER.

10.3 Summary

- PHP Object-oriented programming was created to solve the drawbacks of traditional programming paradigms.
- In object-oriented programming, the accessibility of methods, classes, constructors, and other class members is specified using the Access Modifiers keyword.
- MySQL is a powerful Relational Database Management System(RDBMS) which we will use to learn the basic principles of database and data manipulation using Structured Query Language (SQL) statements.
- MySQL operates in accordance with client-server architecture. This paradigm is made to allow end users, referred to as clients, to use network services to obtain resources from a single machine known as a server.

Database connections and manipulation are both possible with PHP. The most common database used with PHP is MySQL.

10.4 Terminal Questions

1. Write a brief note on inheritance?
2. Briefly explain how MySQL works?
3. Write a brief note on database design concepts?

4. Explain the PHP Class with an example program?

10.5 Answers

Self Assessment Questions

1. Parent class
2. Encapsulation
3. _construct()
4. Mysql_affected_rows()
5. True
6. DDL

Terminal Questions

1. Inheritance is a class derives from a different class. All of the parent's public and protected properties and methods will be inherited by the child class.For more details refer section 10.1.7.
2. MySQL builds a database that enables you to construct numerous tables to store and edit data as well as specify the relationships between each table.For more details refer section 10.2.5.
3. SQL is generally broken down into three categories of statements:

   Data Definition Languages (DDL), Data Manipulation Language (DML), and Data Control Language (DCL). For more details refer section 10.2.6.
4. Once we defined our class, then we can create as many objects as we like of that class type. For more details refer section 10.1.3.

10.6 References

- Larry Edward Ullman (2003). *PHP for the World Wide Web*. Third edition. Peachpit press
- Leon Atkinson, Zeev suraski. *Core PHP Programming.* Third edition
- Jesus castragentto (2004). *Professional Php programming*