**History**

Knowledge about history of PHP

**Installation + Setup**

Download XAMP and install

**Syntax**

All statement end with semi-colon (;)

$variable\_name ; variable name always start with $ sign

**Comments**

// this is single line comments

# this is also single line comments

/\* this is multiple line comments;

**Data Types**

Data type is not required for declaring variables in PHP

Scalar data types

String

Integer

Boolean

float

Compound data types

Array

Objects

**PHP Variables & Constants**

**Variable**

We are declaring variable with $ sign and not mentioning data types

$sale\_amount;

We are assigning value to variable using = (assignment operator)

$sale\_amount = 1000;

**Constants**

DEFINE(“CONSTANT\_VARIABLE”,”constant variable value”);

Predefine CONSTANT Variables

PHP\_VERSION

\_\_FILE\_\_

\_\_LINE\_\_

E\_ERROR

E\_PARSE

E\_WARNING

E\_ALL

E\_NOTICE

TRUE, FALSE, NULL

Settype() & gettype()

**Variable Scope**

Local scope

Global scope global keyword is used for this for example ( global $x; )

PHP store global variable in $GLOBALS[] array

Static scope static keyword is used for this for example (static $y; )

**Type Casting**

Conversion of one type to another data type

**Environment Variables**

$argv

$argc

$PHP\_SELF

$HTTP\_GET\_VARS

$HTTP\_POST\_VARS

$HTTP\_COOKIE\_VARS

$HTTP\_ENV\_VARS

$HTTP\_POST\_FILES

$HTTP\_SERVER\_VARS

**Operators**

Arithmetic Operators (+,-,\*,/,%)

Assignment Operators( = , +=, -=, \*=, /=, %=, .=)

Comparison Operators (> , < , >=, <=, == , !=, ===, ? ) (exp1 ? exp2:exp3)

Execution Operators (backticks ` `)

Increment / Decrement Operators (++$var, --$var , $var++, $var-- )

Concatenation Operators (.)

String manipulation functions (substr, strstr, str\_replace, )

Logical Operators (||, && , ! )

**Control Structure**

**Conditional structure**

**If statements**

If (condition)

{

Statement;

}

**If – else statements**

If (condition)

{

Statement;

}

Else {

Statement;

}

**If – elseif - else statements**

If (condition)

{

Statement;

}

Elseif (condition ) {

Statement;

}

Else

{

Statement;

}

**Switch statement**

Switch (variable)

{

Case value1: statement;

Case value2: statement;

Default: statement;

}

**Break & Continue Key words**

Break keyword break the execution and come out of the loop while Continue restart the loop execution from start

**Loops / Repeat Statements**

**For loop**

For (startvalue; condition; valueincreament)

{

Statements;

}

Foreach loop

Foreach loop is used with array

$arrayval = array (12,10,20,30);

Foreach ($arrayval as $newvar)

{

Echo “$newvar”;

}

**While loop**

$expression = 10;

While ($expression > 1)

{

Statements;

}

**While …… endwhile loop**

$expression = 10;

While ($expression > 1)

Stetement1;

Statement2;

Endwhile;

**Do -----While loop**

Do {

Statement2;

Statement3;

} while (condition);

**Arrays**

There are three types of array

Enumerated Arrays 🡪 it is numerical index array, we can access the element of

array using index number like arrayame[0] or arrayname[1] etc

$arrayname[“name1”] = “Rizwan”;

$arrayname[“name2”] = “Ismail”;

$arrayname[“name2”] = “Yousaf”;

Associative Arrays

To access the elements of array by name.

$students = array (“name1” => “Rizwan”, “name2” => “Ismail”, “name3” => “Yousaf”)

Multidimensional Arrays

Working with Arrays

Count function show the number of element in array

**Functions**

Function is block of code that accomplish a specific task.

There are two types of functions

Built-in functions

User define / custom functions

Function function\_name ($argument1, $argument2)

{

Statement 1;

Statement 2;

}

Arguments can be passed by three ways to functions

1. Default argument values
2. Passing arguments by values
3. By reference ( & symbol is used to make variable reference var)

Returning values from Function

We using return kework to return value from function like below

Return $return\_variable

Variable Function

If function is define like normal then in main script, if function name is assign to

another variable then that variable is consider function of same function

**Classes**

Class is collection of variables and functions

Object is instance of class to initialize the class

In classes variables are called member or properties and functions are called methods

Declaring class

Class class\_name {

$variable\_name;

Function function\_name ($agruments) {

Statements;

}

}

Accessing class or using class

$obj1 = new class\_name;

$obj1 -> variable\_name;

$obj1->function\_name(hello);

Accessing class method without object . we are using scope resolution operator (::)

$result = class\_name::function\_name();

Accessing variable of class from inside the class, we use this operator like below

$this -> $variable\_name;

Constructors

These the method execute automatically, when class object is created

Constructor method name is same as class

Destructors

In PHP there is not destructor concept.

Class Inheritance

In Inheritance we will extend the base class with child class.

class parant\_class\_name {

$variable\_name;

Function function\_name ($agruments) {

Statements;

}

}

class child\_class\_name extend parent\_class\_name {

$child\_variable\_name;

Function function\_name\_child()

{

Statement;

}

}

Overriding Methods

If in child class if with same name function which define in parent is define child

class is called Overriding methods

Accessing parent class method from child class object using ::

**Exceptional Handling**

Try and Catch block

Connecting to Database

try {

$connection = new mysqlii (‘localhost’, ‘root’,’m@g1t786’,’simpledb’);

Mysqli\_set\_charset($connection, ‘utf-8’);

}

catch (Exception $e)

{

print “exception occur”.$e->message;

}

catch (Error $e)

{

print “exception occur”.$e->message;

}