**Experiment Title:** Program based on PHP variables, Expression, arrays, control structure

**Objective:**

To learn and implement Program based on PHP variables, Expression, arrays, control structure

# Theory:Basic PHP

Variables are "containers" for storing information.

Creating (Declaring) PHP Variables

In PHP, a variable starts with the $ sign, followed by the name of the variable:

Example

<?php

$txt = "Hello world!";

$x = 5;

$y = 10.5;

?>

PHP Variables

A variable can have a short name (like x and y) or a more descriptive name (age, carname, total\_volume).

Rules for PHP variables:

A variable starts with the $ sign, followed by the name of the variable

A variable name must start with a letter or the underscore character

A variable name cannot start with a number

A variable name can only contain alpha-numeric characters and underscores(A-z, 0-9, and \_)

Variable names are case-sensitive ($age and $AGE are two different variables)

Remember that PHP variable names are case-sensitive!

**PHP Data Types**

Variables can store data of different types, and different data types can do different things.

PHP supports the following data types:

String Integer Float (floating point numbers - also called double) Boolean Array

Object NULL Resource

PHP Object

An object is a data type which stores data and information on how to process that data.

In PHP, an object must be explicitly declared.

First we must declare a class of object. For this, we use the class keyword. A class is a structure that can contain properties and methods:

Example

<?php

class Car { function Car() {

$this->model = "VW";

}}

// create an object

$herbie = new Car();

// show object properties

echo $herbie->model;?>

What is an Array?

An array is a special variable, which can hold more than one value at a time.

If you have a list of items (a list of car names, for example), storing the cars in single variables could look like this:

$cars1 = "Volvo";

$cars2 = "BMW";

$cars3 = "Toyota";

However, what if you want to loop through the cars and find a specific one? And what if you had not 3 cars, but 300?The solution is to create an array!

An array can hold many values under a single name, and you can access the values by referring to an index number

In PHP, there are three types of arrays:

Indexed arrays - Arrays with a numeric index

Associative arrays - Arrays with named keys

Multidimensional arrays - Arrays containing one or more arrays

**PHP Indexed Arrays**

There are two ways to create indexed arrays:

The index can be assigned automatically (index always starts at 0), like this:

$cars = array("Volvo", "BMW", "Toyota");

or the index can be assigned manually:

$cars[0] = "Volvo";

$cars[1] = "BMW";

$cars[2] = "Toyota";

The following example creates an indexed array named $cars, assigns three elements to it, and then prints a text containing the array values:

Example

<?php

$cars = array("Volvo", "BMW", "Toyota");

echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . "."; ?>

Get The Length of an Array - The count() Function

The count() function is used to return the length (the number of elements) of an array:

Example

<?php $cars = array("Volvo", "BMW", "Toyota"); echo count($cars); ?>

# Procedure :Write web application implementing above codes with PHP basics.

# Conclusion: