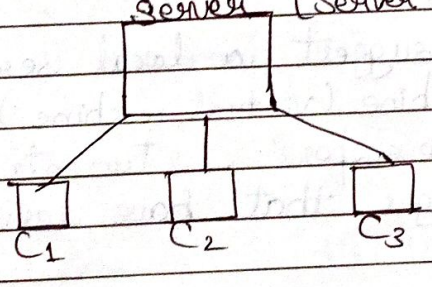
13/3/23

Page :
Date :

Web Based Programming

→ Syntax PHP
→ Web Base App. PHP
→ C R U D
Create Read Updated Delete
→ Database

* Web Based Application	* Standalone Application
→ Internet connectivity must.	→ Internet Not Required.
→ Uses less Number of system resources.	→ Take resources of Operating System RAM.
→ Server required.	→ Eg:- MS office, Calculator.



{Server Site} / Backend

{Client Site} / Frontend

* Web Based Application

→ It is an application that is stored in a remote server and delivered over the internet through web browser.
→ There is no need to download web based application in our system.
→ There can be 'n' numbers of users that can use web based application at a particular time

* Client Side Scripting Language	* Server Side Scripting Language
1) It executes on frontend (client side).	1) It executes on backend.
2) It is visible to user.	2) It is not visible to the user.
3) The data is less secure.	3) The data is more secure.
4) It is highly depend on the web browser.	4) It is not depend on web browser.
5) Eg:- HTML, CSS.	5) Eg:- PHP, JAVA.
6) It is much faster.	6) It is less as compared to client side scripting.

* Local Server

⇒ As name suggest a local server is a privately owned machine (virtual machine) most commonly used by developers', students to store and test webpages that have server side scripting language.

* Remote Server	Local Server
→ Opposite to Local server- the remote server is present remotely (in some other location).	
→ Internet Connectivity is must	→ We don't need internet connectivity.
→ More Expensive.	→ Less expensive.
→ More secure.	→ Less secure.

→ Used by big Organisation.	→ Used by student, developers.
→ Eg:- Google, AJIO.	→ Eg:- WAMP, LAMP, XAMP.

* Dynamic Website	Static Website
→ Back end (executes).	→ Front end.
→ The executive of data can be changed during run time.	→ Can't be changed during run time.
→ Interaction with database.	→ No interaction with database.
→ Content can change during refresh.	→ Content don't change during refresh.
→ Slower that static.	→ Faster than dynamic.
→ More secure.	→ less secure.
→ PHP, node.js are used.	→ HTML, CSS are used.

⇒ PHP (By Rasmus Lerdorf)

* Personal Home Page till 1995 then it becomes Hypertext preprocessor.
* Open source language.
* PHP stands for Hyper Text preprocessor. This is an open source server side scripting language.
* It is used for generating dynamic webpages.
* PHP script resides b/w reserved tags.
* PHP has various inbuilt function.
* Popular website made using PHP are facebook.

Page :
Date :

* <? php
— — —
— — — ?>

* echo → to print.

Eg: <? php
echo "hello"; ?> ⇒ It prints hello

• <? php
$a = 100; ⇒ declaring a variable and printing it
echo "value of A is ". $a;
?>

20/3/23

PHP (Hypertext Preprocessor)

Server Side Open Source scripting language

(loosely Typed language)

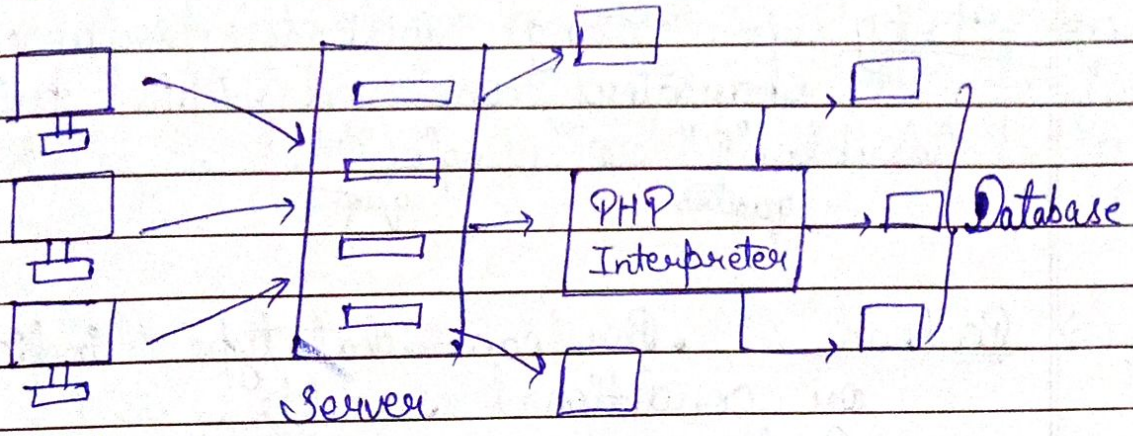
* Variables in PHP

⇒ For naming the variable in php we have to follow some rules:-

1) Our variable name should start from 'A' or 'a' or underscore '_' but not from numeric value.

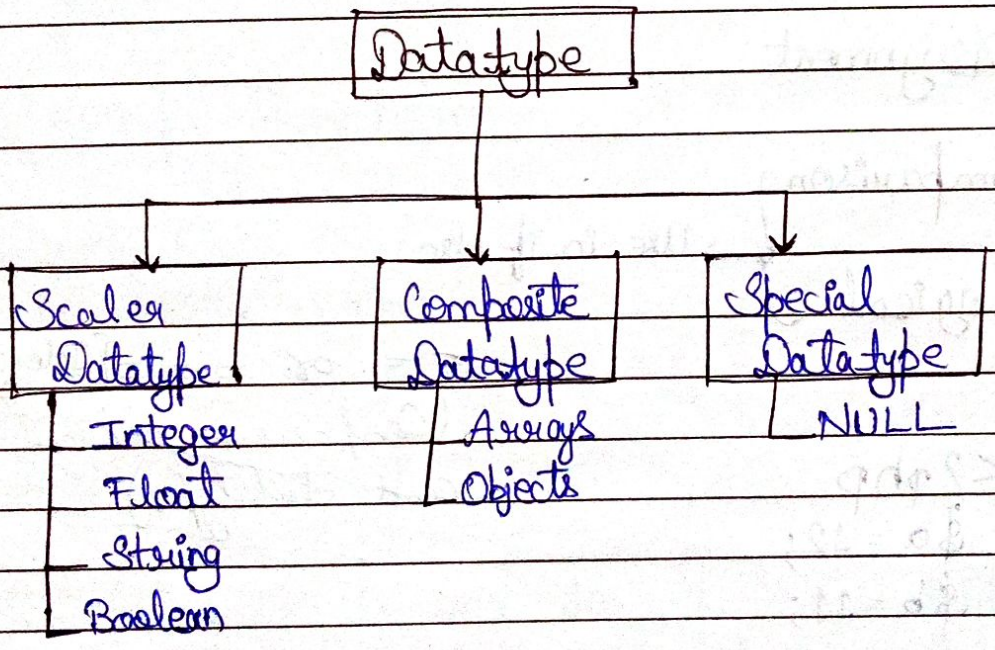
★ In Php we don't specify the datatype of our variable and that's why it is called 'loosely Typed language'.

C:



★ Generally php is not a case sensitive language in case of variable.

C:/XAMPP/htdocs/ → Folder name




```
$ array = ("A", "B", "C");

$ a = 10;
$ b = 103;
$ c = "true";
```

⇒ **String** :- String indicates sequence of characters enclosed within pairs of double quotes "P" or single quote 'R'.

⇒ **Boolean** :- • Boolean datatype indicates logical or conditional results.
• Boolean datatypes has two values either 'true' or 'false'.

21/3/23

* **Operators** :-

1) Arithmetic :- +, −, *, /
2) Assignment
3) Comparison } → Use in If-else
4) Logical }

== or === [Identical Operator]
(Comparison between values and datatypes).

```
<?php
$a = 12;
$b = 11;

echo = "$a + $b = ".$a+$b";
echo "<br>"; → Give the space / next line
?>
```

var_dump()
↓
bool (True) / (False)

22/3/23

* var_dump() :- It is an in-built function which is generally used with boolean datatype values to return the false '(0)' values

```
<?php
$a = 10;
$b = 10.0;

echo "comp". var_dump($a===$b);
?>
```

22/3/23

PHP

* String Functions

- strlen()
- str_word_count()
- strrev()
- strpos()
- str_repeat()

* Strlen() :- It is used to get the string length it return the length of the string on success..
If string is empty then '0' will be return.

```
<?php
$str = "BCA first year is a good batch";
echo $str;  → echo "strlen($str);
echo "<br>";
?>
```

Page :
Date :

23/3/23

- str_word_count():- It is used to count the words which is written in a code.
  → It is an in-built function used to return information about words using in our ~~screen~~ string i.e, total no. of words in our ~~screen~~ string

- strrev():- It is an in-built php function. It is used to reverse the string order.

⇒ echo strrev($str);

- strpos():- It is used to tell the position in ~~on~~ a string.

⇒ echo "Position of sub strings", strpos($str, "BCA");

- str_replace():- Used to replace the string / word in a string.

⇒ echo str_replace("BCA", "MCA", $str);
↓
With what it should be replaced.

- str_repeat():-

⇒ echo str_repeat($str, 131);
↳ It will repeat the string 131 times.

* Php
- Comme
- Synta
- Data
- Oper
- Vari
- Const
- Expr
- Keyw

<?

* Ru

1)
2)
3)
4)

Page :
Date :

23/3/23

* Php :-
- Comments
- Syntax
- Datatypes
- Operators
- Variables → Scopes of variable → Global, Local, Static
- Constants → Super Global Variable
- Expressions → Regular Expression Eg- $x=loo
- Keywords

```
<?php
define ("a", 100);
echo a;
echo "<br>";
$a = 12;
echo $a;
$b = a + 19;
echo "<br>";
echo $b;
?>
```

Without a = value bhele yeh add hogi
(define 'a' = 100) $b = a + 19
= 100 + 19
= 119

With $a = value add hogi
($a = 12) $b = $a + 19;
= 12 + 19
= 31

* Rules of Constants in PHP :-

1) No need to use '$' sign before our const. variable name.
2) We can't start the variable name with 'digits'.
3) We can't use keywords in constant name.
Eg- define ("x", 100); → It can be change afterwards.
4) It's scope is global.

Page :
Date :

⇒ Super Global Variable :-
Some pre-defined variables in Php are "Super global" which means that they are always accessible regardless of scope and you can access them from any function, class or file.

27/3/23

⇒ Local Variable
The variable declared within a function are called local variables to the function and has its scope only limited to that function.

⇒ Global Variable
Variables declared outside the function are called global variables. The variable can be accessed directly outside the function to get access within a function we need to use key word 'Global.'

⇒ Static Variable
It is a characteristic of PHP to delete a variable once it completes its execution and the memory is face but sometimes we need to store the variable each after the completion of function execution.

```
function fun() {
    static $a = 10
    static $b = 11;
    $a++
```

```
    --$b;
    echo "value of B", &b;
    echo "value of A", &b;
    }
fun();
fun();
```

To do this we need to use the 'static' keyword and the variables are called 'static variables.'

29/3/23

* $_GET :- The get method is used to submit the HTML form data. This data is collected by $_GET super global variable.

- ~~He can't se~~ The information send from an HTML form using get method is visible to everyone in the browseer HTML bar

* $_POST :- It is similar to get method The Post method is also used to submit the HTML data. But the data submitted by this method is collected by predefined $_POST super global variable.

- The Information is send using Post method is not visible to user.

Page :
Date :

* **if** :- If statement is one of the most important feature of the many statement languages include PHP. It allows conditional executional of code.

Syntax :-
```
if (exp.) → true
{
  statement 1;
}
?>
```

* **if - else** :- In this often times we want to execute the statement if a certain condition is not true. For this purpose we use else statement.

Syntax :-
```
if (exp) → true
{
  statement 1;
}
else
{
  statement 2;
}
```

* **else - if** :- As name suggest it is a combination of 'if' and 'else' statement. It extends an 'if' statement to execute a different statement In case the original statement is not true then some other condition will be check.

```
<?php
$a = 10;
$b = 20;
if ($a =
{
}
else if (
{
}
else
{
}
```

28/3/23

⇒ Super G

- $ Globals
- $_ Server
- $_POST [
- $_GET [
- $_COOKI
- $_REQUE

Super G
php which
regardl
any fun
Variable

```
<?php
$a = 10;
$b = 20;
if ($a == $b).
{
}
else if ($a > $b)
{
}
else
{
}
```

28/3/23

Php

⇒ Super Global Variable

- $ Globals [ ] → keywords - $GLOBALS[ ]
- $_ Server [ ] → keywords - $_SERVER [ ]
- $_POST [ ] → keywords - $_POST [ ]
- $_GET [ ] → keywords - $_GET [ ]
- $_COOKIES
- $_REQUEST

Super Global Variables are predefined variables in php which means that there are always accessable regardless of scope and we can access them from any function, class or file., examples of Global Variable are $_R.

→ $Globals :-
$Globals is a super global variable which is used to access variable globally in array $GLOBALS[ ] where index holds the global variables name which can be accessed from anywhere without using any special keywords

→ $_SERVER
- Method
- filename → $_SERVER['PHP_SELF']
- Server name / host name
- Web Browser version
- I.P address

It is a php super global variable that store the information about headers, Path, Script, location.

Some of the variable are :-
$_SERVER['PHP_SELF'] → file name
$_SERVER['HTTP']

Example :-
```
echo $_Server['PHP_SELF']; ← file name
echo "<br>";
echo $_Server['HTTP_HOST']; ← host name
echo $_SERVER['HTTP_USER_AGENT']; ← browser
echo $_SERVER['REQUEST_METHOD']; ← Post / Get
echo $_SERVER['REMOTE_ADDR']; ← IPaddress
```

Output
/PH
loca
Mozi
GET
426:

→ $_R
It is
calle

$_R
ser
$

3/4/23

* Swit

⇒ Swit
to
con

Syntax :-

Page :
Date :

Output

/PHP1/Super_global.php
localhost
Mozilla/5.0
GET
426:9694:4902

→ $_REQUEST :-
It is a super global variable which is used to collect data from HTML forms.

$_REQUEST :- It is not so frequently used the same function is for performed by $_POST [] and $_GET[].

3/4/23

* Switch Case

⇒ Switch case statement in PHP are used to execute one statement from multiple conditions.

Syntax :-

```
Switch ($a)
{
 case 1;
  echo --
  break;
default;

}
```

Page :
Date :

* **isset()** :- It checks whether a variable is set, which means that it has to be declared and is not null.
This function returns 'true' if the variable exists and is not null otherwise return 'false'.

5/4/23

* **Php Loops**

⇒ **For Loop**

```
<?php

for($i=1; $i<100; $i=$i+10)
  {
    for($j=$i; $j<=$i+9; $j++)
      {echo $j." ";
    }
  echo " ".<br>";
}

?>
```

⇒ **Do while**

```
$a=1;
do
{
  echo "This is outer do-while".$a."<br>"."<br>";
  $a++;

  $b=1;
  do
  {
```

⇒ **goto**

```
for(
  {
    if
    {
    }
  }
a=
ec
```

```
echo "This is Inner do-while". $b. "<br>";
  $b++;
}
while ($b<=3);
{echo "<br>";
}
  while ($a<=3);

?>
```

goto

```
<?php
for( $i=1; $i<=10; $i++)
{
  if ($i==6)
  {
    goto azx;
  }
  echo $i. "<br>";
}
azx;
echo "out of the loop goto statement";
goto azs;

?>
```

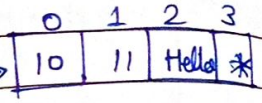
6/4/23

Page :
Date :

PHP Arrays

→ Index Arrays
→ Associative Arrays } → For each Loop

```
$a = array (10, 11, 13, 14);
$a = array (10, 11, "Hello", '*');
$a = [10, 11, "Hello", "*"];
```



→ It is a collection of heterogeneous data type.

* In Php we can define arrays in two types:-

1) Index Arrays
→ It is an array represented by an index number by default.
→ All elements of index array are represented by an index number starts from zero.
→ It can store integer, float, strings etc.

2) Associative Arrays
→ Php allows to associate name with each array element using the symbol '=>'.

→ Ex:- $age = array ("Key" => "value");
= array ("A" => "10", "B" => "11");

* For each loop :- It is a keyword

For each ($age as $b)

⇒ For each loop works only on arrays and is used to loop through each key value pair in and array.

* Code of f

→ <?php
$a = ar
foreac
{
echo $
}
?>

* Code of

⇒ <?ph
$age
foreac
{
echo
}
?>

* print

* Code of foreach loop :-

```
<?php

$a = array("one", "two", "three", "four");

foreach ($a as $b)
{
echo $b. "<br>";
}

?>
```

* Code of Associative Array :-

```
<?php

$age = array("name1" => 31, "name2" => 41, "name3" => 29);

foreach ($age as $key => $val)
{
 echo{ $key. "_____". $val. "<br>";
}
?>
```

* print_r() :- • It is an in-built function that displays information about a variable in a human readable form.
• It shows the information stored in a variable in an easily understandable and easy to read form.

* Multidimensional Array :-

Emp I.D.	Name	Salary
1.	A	10,000
2.	B	10,000
3.	C	10,000

```
$a = array(
        array(1, "A", 10000),
        array(2, "B", 10000),
        array(3, "C", 10000),
    );
```

⇒ It also known as arrays of arrays.
⇒ It allows us to store data in tabular form.

11/4/23

Eg:-

```
$emp = array(
        array(1, "name 1", 1000),
        array(2, "name 2", 2000),
        array(2, "name 3", 3000)
    );
print_r($emp);
```

or.

```
for($row=0; $row<count($emp); $row++){
for($col=0; $col<count($emp); $col++){
    echo $emp[$row][$col];
}
echo "<br>";
}
```

12/4/23

Page :
Date :

* Count() :-
⇒ It is an in-built php function used to count the current elements of arrays.
⇒ For variable which is not set the function returns 0.
⇒ Syntax
Count ($array)
↳ Refers to the array which returns the value

* For each :-
```
For each ($emp as $a)
 {
  foreach ($a as $b)
   {
    echo $b;
   }
 echo "<br>";
}
```

12/4/23

* in_array()
⇒ Return value is 0 or 1
⇒ It is pre-define/in-built function that is used to check whether the given value existing in array or not. It returns either true or false i.e, 0 or 1.

* array_search()
⇒ Widely use for search and locate particular value in an array and if there are more than one value then the key of the first

Page :
Date :

13/4/23

matching value will be returned.
⇒ If it successfully found the specific value it returns corresponding key / index value.

* array_pop()
⇒ Use to delete or pop of the last element from array passed to it as a parameter.
⇒ It reduces the size of array by one.

* array_push()
⇒ It is in-built function use to add one or more element in our array but in last position of array but it allows to add any number of element.

* array_shift()
⇒ Use to remove the first element.
⇒ It reduce the size of array.

* array_unshift()
⇒ It is used to add element in an array at the first position of the array.

• array_s
it take
paramet
array

$1 = [

$ Sub =

$ new

echo
print
echo
echo

• arr
ref
arr

$ a

$

$

13/4/23

- array_replace() :- This is built in PHP function and it takes a list of array separated by commas as parameters and replace all the values of first array that has same key in the other array.

```
$l = ["c++", "c", "c #", "php"];

$sub = ["operating-system", "compiler_design",
        "automata", "compiler_net"];

$new_array = array_replace ($sub, $language);
                    here $s array elements
                    is being replaced by the
                    elements in array $l.
echo "<pre>";
print_r ($new_array);
~~echo "<pre>";~~ print_r ($sub);
echo "</pre>";
```

- array_replace_recursive() :- It is same as replace but it is used for ~~Mat~~ Multidimensional array. Here the both array's key must be same.

```
$a1 = [
        ["a" => "red", "b" => "blue", "c" => "green"]
      ];

$a2 = [
        ["a" => "black", "b" => "white", "c" => "yellow"]
      ];

$new_array = array_replace_recursive ($a1, $a2);
print_r ($new_array);
```

Page :
Date :

* The difference b/w these 2 function is the array_replace_recursive is used with multidimensional array.

* If a key from array 1 exist in array 2 value from array 1 will be replaced by value from array 2.

- array_merge():- It is used to merge 2 or more arrays into a single array. The merging occurs in such a manner that the values of one array appended at the end of another array.

```
$ course = ["bca", "mca", "btech", "mtech", "phd"];
$ class = [12, 11, 10, 9];
$ array_new = array_merge ($course, $ class);
echo "<pre>";
print_r ($array_new);
echo "<pre>";
```

- array_merge_recursive():- It merges 2 array into one. The main difference b/w the 2 function is, the array_merge_recursive is used with multidimensional array. When 2 or more array elements have the same key then instead of overwriting the keys the array_merge_recursive function makes the values as array.

17/4/23

* Function

⇒ Function be

⇒ A functi When

⇒ A funct acco

* Adva
- Code
- Easy
⇒ Time
⇒ Less

* Creati
. Func

- array_com PHP used new one One arr

$ array_

Both the no. of

- array_combine():- It is an in-built function in PHP used to combine 2 array and create new one by using one array for key and one array for values.

$array_new = array_combine [$class, $course];

used as key — used as value

Both the array should have same no. of elements.

17/4/23

* Function in php

⇒ Function is block of statement that can be used again & again in a program

⇒ A function will not execute automatically. When page ~~loades~~ loads.

⇒ A function will execute when we call it according to our convienience.

* Advantages of Function :-

- Code reusuability.
- Easy to maintain.

⇒ Time Saving
⇒ Less code

* Creating Function

Function ()
always

→ To call a function we just need to write the name of the function following by open and close parenthesis and semi-colon

→ A function name can't start with number with an alphabet or underscore.

<?php

* Function Type

1) Built-In :- PHP provides with huge collection of built-in library function. These functions are already coded and stored in form of function. To use these built-in function we just need to call them according to our requirement.

2) User Define :-

→ PHP allows us to create ~~our~~ customize our own function which is called user-define function. Using this function we can create our own set of code and we get using wherever require.

Program
<?php
echo "
Function
{
$a =
$b =
echo
}
Sub (
echo
echo
Sub (
?>

→ PHP f
insi
para
→ The
func

NOTE:- The
th

Page :
Date :

Program

```
<?php

echo "PHP FUNCTIONS". "<br>";

Function sub()
{
  $a = 10;
  $b = 21;
echo "addition: ---", $a+$b;
}
sub();
echo "test code". "<br>";
echo "test code". "<br>";
sub(12,31);
?>
```

→ PHP provide ~~us function~~ option to pass parameters inside a function. We can pass as much parameter we want.

→ The parameters work like variable inside a function.

NOTE:- The no. of parameters that we pass during our function definition should be equal to the no. of parameters that we pass during our calling function.

18/4/23

Page :
Date :

Functions in PHP

*Important topics

→ Defining & Calling function
→ Parameters passing
  ↳ Passing Array as a parameter.
→ Passing by value and Passing by reference
  ↓ Call by Value
  ↓ Call by reference
→ Return statement in PHP function
→ Static Parameters
→ Variable as a function.

Assignment Questions [Submit it on 28th April (Deadline)]

Q1. Differentiate b/w static and dynamic website.

M.Imp Q2. Diff. b/w Index array and Associative array. Give one example of multidimensional array Associative.

M.Imp Q3. What are super global variables? How $_POST is different from $_GET? with examples?

Q4. What are different types of loop in PHP? How for each loop works?

```
Function [redacted]m($array)
{
  Foreach ($array as $v)
  echo $v;
}
$a = [1,2,3];
[redacted]m($a);
```

* Passing array as a parameter

```
<?php

Function Fun_array($arr) //_____Passing array as a Parameter
{
  $sum = 0;
  Foreach ($arr as $value)
  {
    $sum += $value;
  }
  return $sum;

}
$a = [1,31,51,21];
$b = Fun_array($a);

echo "TOTAL:-----". $b;

?>
```

php.net

Page :
Date :

19/4/23

Example :-

```
Function nm ($Fname = "ram", $lname = "Shyam")
  {
   echo "Hello $fname $lname";
  }
  nm ("John", "Wick");
   nm ();
```

* Return Statement

The Purpose of return statement in PHP is to return control of program execution back to the environment from which it was called.

* Lab Work :-

```
<? php
Function check_prime ($num)
{
  if ($num == 1)
  return 0;
  For ($i = 2; $i <= $num / 2; $i++)
  {
    if ($num % $i == 0)
   return 0;
  }
   return 1;
  }
  $num = 46;
  $Flag_val = check_prime ($num);
if ($Flag_val == 1)
  echo "It is a prime number";
else
   echo "It is a non- prime number"
?>
```

19/4/23

Page :
Date :

PHP Functions

Variable of functions

```
Function nm($a)
  {
  --
  }
$var = "nm";
$ var ();
<?php
Function nm ($ name)
 {
  echo "Hello $name";
 }
$ var = "nm";
$ var ("John");
? >
```

Anonymous Function

```
$ var = Function ($a)
  {
  }
  $ var ();
<?php
$ var = Function ($a)
 {
  echo "hello $a";
 };

$ var ("asd");
 ? >
```

* Pass by value

1) In this parameters are passed as a value that means if the value of parameters within the function is ~~not~~ changed then it will not effect the original value of a parameter outside the function.
2) Makes the copy of actual parameters

Pass by Reference (&) → ampersand

1) In this parameters are passed as a address i.e., if the address of parameters within the function is unchanged then it will not effect the ~~outer~~ original address of a parameter outside the function.
2) Make the copy of actual parameters passed to the address.

2)
```
Function nm ($a)
{
  $a += 7;
}
$b = 100;
nm($b);
echo "Value of B: $b";
```

3.)
```
Function nm2(&$a)
{
  $a += 7
}
$b = 100;
nm2($b);
echo "Value of B: $b";
```

20/4/23

```
<?php
Function first ($num)
{
  $num += 15;
}

Function second (&$num)
{
  $num += 17;
}
$b = 10;
first ($b);
echo "Value of variable B after passing as
      value: $b". "<br>";
second ($b);
echo "Value of variable B after passing as
      reference : $b". "<br>";

?>
```

Page :
Date :

PHP Errors [Unit-2]

- Warning error
- Notice error
- Parse error / syntax
- Fatal error → Critical Error

} → There are 4 types of error

Basically an error is a mistake in a program that may be caused by writing an incorrect code or syntax.

⇒ **Warning Error :-**

- It is in PHP does not stop the script from running. It only warns you that there is a problem and this problem may cause some issues in future.

⇒ **Most common cause of Warning error :-**

- Calling an external file that does not exist in the directory.
- Wrong parameters in function.

```
include ("first.php")
```

# Include Statement :-	# Require Statement
⇒ It takes all the text / code / script HTML that exists in the specified file and copies it into the file that uses the include statement.	⇒ It will not execute the statement after the require statement
⇒ It will show warning error	⇒ It will show fatal error.

Page :
Date :

1 May '23

⇒ Notice Error :-

- Notice Error are also minor error that all similar to warning error. And they also don't stop script execution often the system is uncertain whether it's an actual error or code.

⇒ Parse Error :-

- Also known as syntax error cause by misuse of missing statement of symbol. The compiler catches the error and terminate the script.

Causes

1) Unclosed quotes / parenthesis.
2) Missing or extra semi-colon.
3) Spelling mistake
4) Etc.

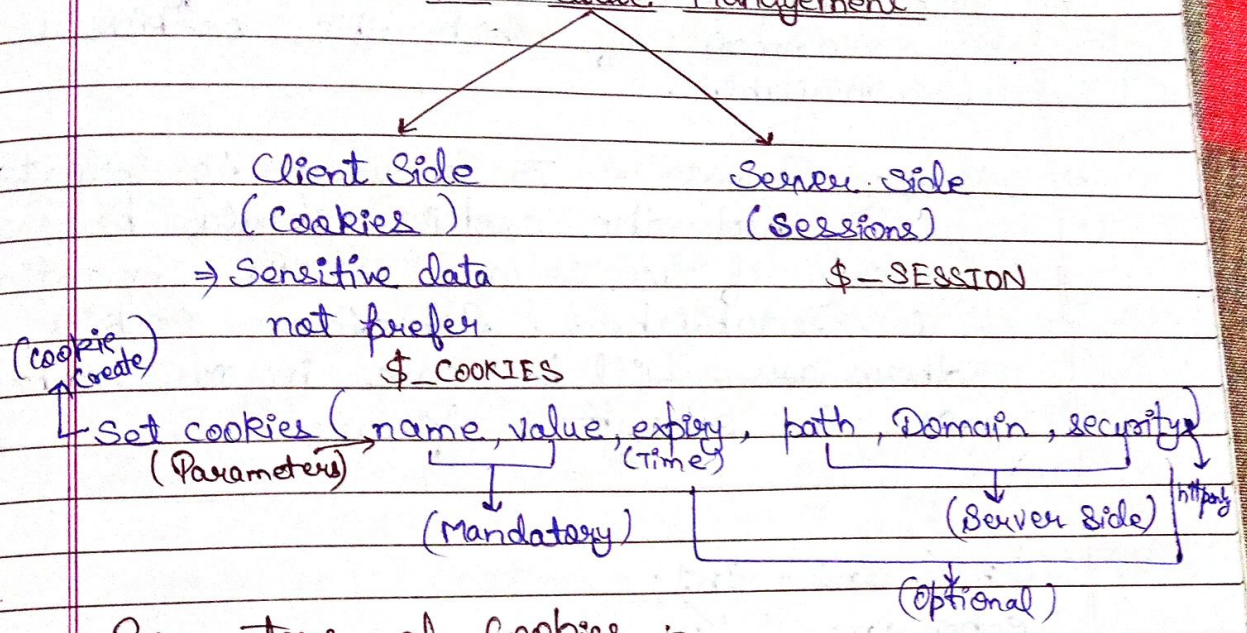
⇒ Fatal Error :-

- They are the one that crashes the program and stops the further execution of our script.

1 May '23

Page :
Date :

PHP State Management



* Parameters of Cookies :-

1) There are various parameters that we used in our set cookies().

⇒ **Name Parameter** :- It is used to set the name of the cookies.

⇒ **Value Parameter** :- It is used to set the values (user name, password, location, web browser).

⇒ **Expiry Parameter** :- It specifies the time period for which the data of our cookies will be stored in our web browser and afterthat specific time those cookies will not be access.

* Session Cookie

	Persistence Cookies

Page :
Date :

9 May'23

⇒ Path Parameter:- It is used to specify the domain of which the cookies is available.

⇒ Security Parameter:- It is used to indicate that the cookies should be sent only if a secure HTTPs connection is established. Sometimes cookies will be save in our webbrowser are in encrypted form.

```
$n3 = $n2 + $n1;
echo $n3, ' ';
$n1 = $n2;
```

```
<?php
Set cookies ("user", "john", time()+86499);
echo "Cookies are set";
?>
```

Q. How
A In
sto
f
be
c
ope
th
u

* S

8 May'23

Page :
Date :

Session in PHP (Server Side State Management)

Q. How sessions are better than cookies?

In general cookies are very useful for storing data related to user but they have serious threat issue / security issue because cookies are stored on user's computer (~~Server~~ Client Side) and thus they are open to attackers to easily & modify the content of cookies and this can result in breakdown the web applications.

* **Session :-**

When we store the user specific information in our server side and this information is available in every webpage of that particular web application then we can say that session has been created.

```
Session_start();
Session_unset();
Session_destroy(); → No argument
Unset($_Session["Pass"]);

Session_start();
$_Session["user"] = "Priya";
$_Session["Pass"] = "***";
echo $_SESSION["user"];
```

8 May'23

Step1:- Creating session [session_start();]
↳ The first step is to start a session with the help of start function. This () is a mandatory function before every step of session such as creating, declaring variable.

[session_start();]

* This is the first thing you mention before any HTML tag.

Step2:- Creating Session

Step2:- Declaring Session Variable [Accessing data]

→ We have to use a super global variable ($_Session).

Eg:- $_SESSION ["user"] = "Priya"; // Declare.
$_SESSION ["Pass"] = "****";

* In each step whether you are declaring or accessing session variable we have to use session_start().

8 May '23

Page :
Date :

Sessions in PHP (Server Side)

→ Mandatory function

Session_start(); #login

```
$_SESSION [ " " ] = " ";
echo ;
```

logout

```
session_unset();
session_destroy();
```

⇒

```
<?php
session_start();
echo "Welcome to my webpage" ;<br>;

$_SESSION ['course'] = "php";
$_SESSION ['Sem'] = "2nd semester";

echo " ";
?>
```

⇒ Session Variables can be easily accessed but for that we have to use SESSION_START() and then by passing the corresponding key (Session name) to the $_SESSION super Global Variable.

⇒ Use for destroying a particular session value :- unset($_SESSION['Sem']);

Page :
Date :

⇒ For destroying completely function:-
SESSION_destroy();

⇒ In Session_destroy() we don't pass arguments / parameters.

9/5/23

PHP State Management

- Query String
- Hidden field

Get Method [Local host / filename / - - -]

< input type = "text" ;
Submit
Password
Hidden → $_SERVER['PHP_SELF']

Eg:- of hidden Field

< input type = "hidden" name = "hname" value = "bca" >

```
<?php
echo "Welcome: ". $_GET['fname']."<br>";
echo "Your course: ". $_GET['hname'];
?>
```

* Hidden Field :-

→ Hidden Field is not displayed on the page It simply stores the text value specified

in value attribute [value = "ka"].

→ Hidden fields are created for passing additional information from the client side to the server side

~~Eg:~~ ~~<Form action = query.~~

* ~~Query_String2.php?~~ ~~fname~~ = 'John' & Roll = 10'

Query String which is shown in URL.

in value attribute [value = "ka"].

→ Hidden fields are created for passing additional information from the client side to the server side

~~Eg:~~ ~~Form action = query.~~

* Query_String2.php? fname = 'John' & Roll = '10'.

Query String which is shown in URL.

10/5/23

* Query String :-

⇒ It is a set of characters attached at the end of ~~its~~ URL.

⇒ It begins with the file name after that '?' and '&' this operator is used to include one or more parameters.

⇒ Each parameter is represented by unique key [name, roll no.] / value. ≡ {.key = value}.

Course = BCA
[Key] [Value]
Unique

1

Key Value
↑ ↑

* parse_str("name = Rahul & Roll = 10", $a);

• parse_str() :-

⇒ It is used to convert the parameters of query string into an array.

Page :
Date :

21/5/23

* File handling in PHP

.txt
- Read
- Write
- append → (Previous info not destroyed)

22/5/23

Type Casting of Variable

```
$a = 14.65;
$a = (int) $a; → Type Cast
$a = 14
```

```
$a = 15.441;
```

23/5/23

File Handling in PHP

- Text
- Binary

$_FILE

Modes :- r, w, a, r+, w+, a+

gets()
getc()

read file()
fopen(___, ___)
fread( )
fclose( )

- OOPs
- (Object Oriented Programming)

* Implicit Typecasting
Automatically change the datatype.

* Explicit Typecasting
Exclusive / Manually change the datatype.
or

* Juggling of variable

Type Casting of Variable in PHP :-

IMPLICIT TYPECASTING	EXPLICIT TYPECASTING
→ $x = 2; $y = 4; echo var_dump($x/$y)." "; echo var_dump($y/$y);	→ $a = -15.441; var_dump($a); echo '=='; $a = (int)$a; var_dump($a); echo ' ';
→ It change the datatype automatically.	→ It change the datatype manually / exclusive.

→ In PHP we don't need to specify datatype. Variable PHP parser ~~sets~~ automatically ~~but~~ set it for us
→ But If we want to change the datatype of our variable at any time according to user convenience then we use the technique of type casting.

* Juggling of Variable :-

→ $x = 2;

```
echo $x : "<br>";
$x = "hello";
echo $x;
```

→ It also change the datatype automatically.

* fopen()

```
<?php
$fo = fopen("textfile.txt", "r");
echo $fo;
?>
```

24/5/23

File Handling in PHP :-

File handling in PHP allows us to create a file, Read a file, write a file, close a file.

```
$a = fopen("file.txt", "r");
echo fread($a, filesize("file.txt"));
fclose($a);
```

⇒ fopen :- It is used to open a file it requires to arguments these are mandatory arguments.

1. File name 2. Mod

Argument in which want to operate a file.

Page :
Date :

MODS

r (read) r+ (read write)
w (write) w+ (write read)
a (append) a+ (Both)

1) Read (r) :- Read mod open the file for reading only. The file pointer points at the beginning of the file.
r+ :- Read and write also.

2) Write (w) :- It help us for to perform only writing operation in the file. The file pointer point at the beginning of the text file. And it overwrite some previous information pre existed information.

→ If the file does not exist the mode then we create a file and then perform the operation.

29/5/23

read file ()
fopen ()
fclose()
fread()
fgets()
fgetc()
file exists()
copy (____) Overwrite previous Information
rename (____, ____)
unlink (To remove a file)

delete ( )

Page :
Date :

→ Folder

Directory Functions

Root Directories

Make:- mkdir ("Foldername")
to create directory

Remove:- rmdir ("____")

File Information function (Getting info on file)

~~file Information~~
filesize ( ); → bytes
realpath ( );
pathinfo ( ); → Array
filetype ( );

⇒ Copy ( ) :-
- If data already exist in that file then overwrite.
- If file does not exist then first it will create that file ~~and~~ after that copy it.

⇒ mkdir ( ) :-
It is used to create directory (folder) in PHP. It is used to create directories with specific path names

⇒ rmdir ( ) :-
It is used to remove directory (Folder) in PHP.

```
echo filesize($file)."<br>";
echo filetype($file);
echo realpath($file);
echo "<pre>";
print_r(pathinfo());
echo "</pre>";
```

realpath($file) → This helps to retrieve the directory name.

Output / array

```
[dirname] => C:\xampp\htdocs
[basename] => txtfile.2.txt
[extension] => txt
[filename] => txtfile2 );
```

30/5/23

→ OOP :- Object Oriented Programming language

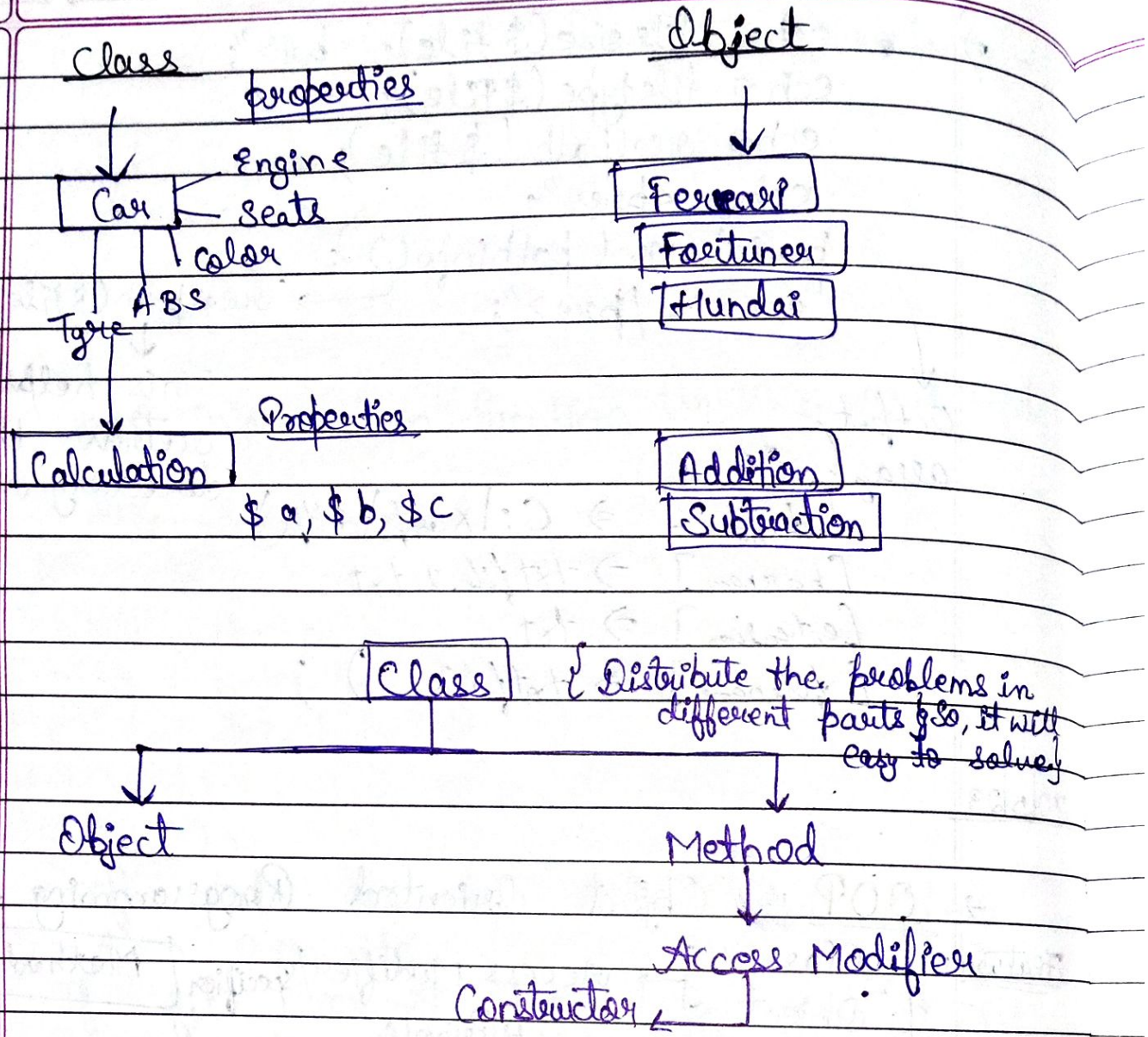
Features
- Class
- Object
- Inheritence
- Encapsulation
- Polymorphism :- Multiple forms

Class, Object → Access Modifier/Specifier
- Private
- Public
- Protected

Method/Function
- Constructor
- Destructor

→ Procedural Oriented
→ Functions, logic

```
$c = $a+$b;
$c = $a-$b;
```
this -> C



* File Information Function :-

⇒ Those predefined functions which help us to retrieve the information about file and these specific function can be size of file [filesize()], realpath(), & pathinfo().

- filesize() :- It returns the file size in bytes.

1 Character = 1 byte

If no character exist it will return false = 0 boolean = 1, 0.

⇒ Syntax of filesize() :-

filesize("text.txt");
↓ → This specify the path of the file.
htdocs/PHP3/text.txt

→ If in some case whose size we want to determine about exist in same folder we have to specify whole path.

- Realpath() :- It is an in-built function ~~in~~ in php which is used to determine the absolute path / actual path.

C:/htdocs/PHP3/text.txt

→ It will remove the unnecessary symbols such as '#', '%', '//'.

- Pathinfo() :- It is an in-built php function which is used to return the information about our file but in associative array form. ~~Key as value~~
→ Gives Directory Name, Extension, Basefile Name, file name.

31/5/23

Page :
Date :

* $_FILE [' '] [' ']

Field Name — Attribute — file name

file name:
- Pdf & Extension
- Size of our file
- Temporary Name

→ Mandatory

1) Input type = "file" name = "myfile".
2) Enctype = "Multipart / form-data"
   After Method
3) Input type = "Submit" name = "upload" value = "upload file"

* ENCTYPE :-
→ This attribute of form tag specific the method of encoding for form data.
→ This attribute can only be used with post method.

* Multipart / form-data :-
→ This value is necessary if user will be upload a file through a form.
→ Method always post.

* $_FILE :- It is a super global variable is an associative array containing items uploaded via a post method.

→ Properties :- $_FILE [name_of_field] ['name'] ['size']
['temp_name']
['Type'] ['error'] → 404

⇒ move_uploaded_file ( )

1 June '23

Page :
Date :

→ File Uploading & Downloading

HTML Form ← [- type = 'file'
- enctype = Multipart/form-data

Php ← [- $_FILES[ ]
- Move_uploaded_file(tmp_name, ____ . $_fname);
[download attribute]

```
Array
(
    [name] => Som.jpg
    [type] => image/jpeg
    [tmp_name] => C:\Xampp\tmp\phpDC5B.tmp
    [error] => 0
    [size] = 15174
)
```

file uploaded successfully
Click here to download a file

⇒ move_uploaded_file() :-
This function is used to upload a file to a new location. We have to pass two arguments in this function i.e.,
1.) tmp_name
2.) location of our file (or) folder name [Where we want to upload our file]

⇒ Download Syntax :-
<a download href = "uploads/<?php echo $f_name; ?>"> click here to download a file .

=> Download Attribute :-
It specifies that the target (Link in href) will be downloaded when user click on that link.

* Procedural Oriented Programming	Object Oriented Programming
1.) Program is divided into small parts called ~~logics~~ functions.	1.) Program is divided into small parts called objects.
2.) Top-down Approach.	2.) Bottom-up Approach.
3.) Programming is less secure.	3.) More Secure.
4.) We don't have the feature of access modifier.	4.) We have the feature of access modifier.
5.) They have no any feature.	5.) ~~Featt~~ Features are class, object, Inheritence, Encapsulution, Polymorphism

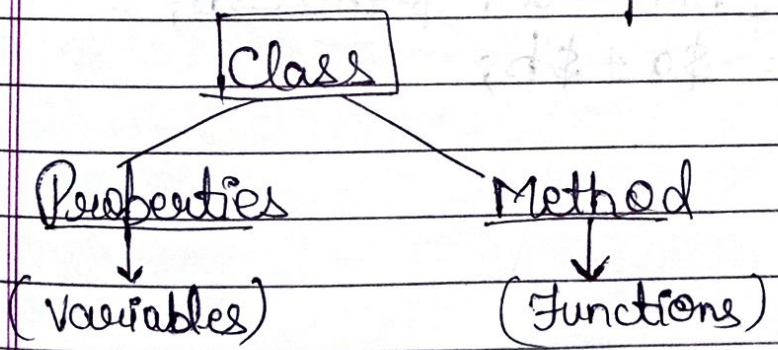
5/6/23

* Class :- It act like ~~as~~ a blueprint or a template for a object.

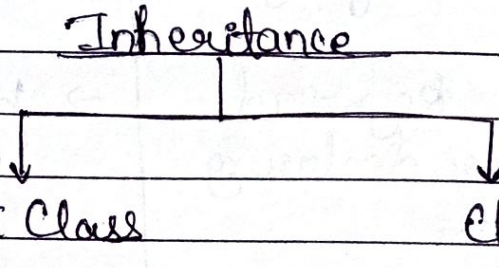
* Object :- It is an instance of class.

Page :
Date :

Class	Object
→ It is the blueprint or a ~~object of class~~ template of object.	→ It is the instance of class.
→ We use keyword 'class' for declaring our class.	→ We use keyword "New' for declaring our object.
→ It is the collection of similar objects.	→ It is an real world entity.
→ Classes are declared only once.	→ Object declared multiple times.
→ Class do not allocate memory when it created.	→ Object allocates memory when it declared.
→ Eg:- Fruit is a class.	→ ~~Eg:- If Fruit is an object.~~ → Eg:- Mango, Apple are an objects.



* Inheritance :-
When a class is define by inheriting by existing function and properties from parent class then it is called Inheritance.



* Polymorphism
It is an object oriented programming feature where same function can be used for different purposes.
For eg:- A function in which the name of function remain same but the arguments that we passed to that functions can do different tasks.

⇒
```
Class Calculation {
  Public $a, $b, $c;
  function Sum()
    {
$this-><= $this->a + $this->b;
  } $c = $a + $b;
```

6/6/23

Page :
Date :

OOPS Features

- reusability
- Debugging
- Understandable
- Inheritence

$a=10, $b=20;
$c = $a+$b

* **To Create a Class :-** → Class

Syntax:-
```
Class Calculation {
    Public $a, $b, $c;          → Properties
    function sum() {            Method
        $this->c = $this->a + $this->b;
        return $this->c;
    }
}
```

* **To Create an Object :-** 'New' keyword should be use

Syntax:-
```
$obj1 = new calculation();
$obj1->a = 10;
$obj1->b = 50;
echo $obj1->Sum();
```

* **$this :-**

⇒ This keyword '$this' refer to the current object and is available only in the method.

7/6/23

```
<?php
class calculation
{
  public $a, $b, $c;
    function sum()
    {

      $this->c = $this->a + $this->b;
       return $this->c;
    }
  function sub()
    {
     $this->c = $this->a - $this->b;
       return $this->c;
    }
     $x1 = new calculation(); //--- Object declaration
                (calling)
      $x1 =>a = 110;
      $x1->b = 11;
     $x2 = new calculation();
      $x2->a = 100;
       $x2->a = 10;
   echo "sum value of x1 - - ". $x1->sum();

  ?>
```

Output

$x1 = 121.

Page :
Date :

* Access Modifier

Access Modifier of properties & method is used to control the access of variables functions.

Access Modifier	Class Itself	Outside Class	Derived Class
Public	✓	✓	✓
Private	✓	✗	✗
Protected	✓	✗	✓

Public :- In Public access modifier the properties & method can accessed with in the class or outside the class or derived class.

declaration

Private :- In Private access modifier the properties & method can accessed with in the class itself not in outside & derived class.

();

Protected :- In Protected access modifier the properties & method can accessed with in class itself and in derived class but not in outside the class.

* Constructor	Destructor
→ Accepts one or more arguments.	→ No arguments are passed. Its void.
→ Function name is __construct().	→ Function name is __destruct().
→ Constructor is involved automatically when the object is created.	→ Destructor is involved automatically when the object is destroyed.
→ It can be overloaded.	→ It cannot be overloaded

8/6/23

OOP Features

- Class & Object
- Access Modifier (only with ~~the~~ in class)
- Construct
  - Object declare with argument
  - Function __Construct
    {
    echo "This is construct Function
";
    }
- Destruct
  - Function __destruct
    {
    echo "This is destruct function
";
    }

Page :
Date :

```php
<?php
class abc
{ public $a, $b;
  function xyz()
    {
      echo "This is xyz function <br>";
    }
}
$obj = new abc();
echo $obj -> xyz();

?>
```

Output

⇒ This is xyz function.

NOTE :-

→ Construct function run at first time.

→ Destruct function will run at last time when all in-built function are called.

```php
<?php
class abc
{
public $val1, $val2, $c;
function sum()
{
echo "Addition of two numbers". $this->c-
$this->val1 + $this->val2. "<br>";
}
```

```
function __construct ($v1, $v2)
{
    $this -> val1 - $v1;
    $this -> val2 - $v2;
    echo "This is construct function <br>";
}
    → ~~echo~~ __destruct()

$obj = new abc(200,900);
// $obj -> val1 - 100;
// $obj -> val2 - 200;

echo $obj -> sum();
// $obj -> $a-10;
?>
```

```
function __destruct()
{
echo "This is destruct function <br>";
}
```

Output :-

This is construct function
Addition of two numbers 1100.
This is destruct function.

* Construct

→ In OOP the keyword construct [Function/Method] define inside the class and is called automatically at the time of object declaration

→ The Purpose of construct method is to initialize the object.

Syntax :-

```
function __construct()
{

}
```

Page :
Date :

* Destruct

→ Destruct method is opposite of construct method, Destruct method also called automatically at the time of object declaration but it will be called at last after construct and any other user-define function.

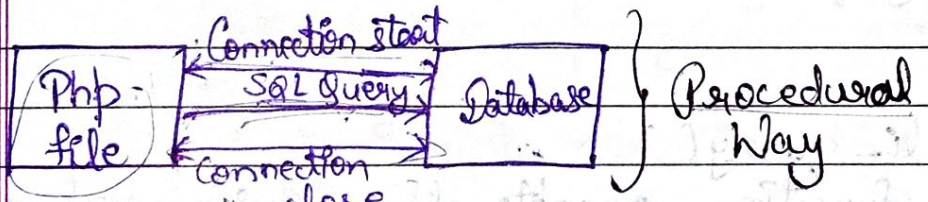
→ It is just called before de-allocation of memory.

Syntax:- function __ destruct ( )
{

}

12/6/23

MySQL



S ⇒ mysqli_connect ([Connection name] host name, username, Password) ;
Improved version

$-

→ mysqli_query (1, 2) ;
Connection
Query

→ mysqli_close ( ) ;
Connection name

* mysqli_connect()

⇒ This is the pre-defined function in Php which is used to connect our PHP file with database. This is the improved version compared to mysql_connect().

Parameters of mysqli_connect() :-

⇒ host :- It specifies the host name & IP address. But in case of local server host name will be 'local host'.

⇒ Username :- It specifies the username of mysql. The username By default is 'root'.

⇒ Password :- By default :- Blank.

⇒ Database

Syntax :-

```
mysqli_connect("localhost", "root", "", database name);
```

* mysqli_query()

⇒ This function accepts the string value representing a query as one of the parameters and execute given query on the database.

Syntax :-

```
mysqli_query($connection, $query);

$query = "create database college");
```

Both of these

⇒ It has 2 parameters are connection & query and these are mandatory function.

Page :
Date :

* mysqli_close() is used to
=> This function close the connection between user name and database.

Syntax :-
mysqli_close("Connection name");

=> It has only one parameter i.e, "Connection name".

13/6/23

PDO [PHP Data Object]
→ SQLite

PHP My Admin is an Interface where we can create our database or Table without writing (not always) Query.
It make the program very easy to create.

Php Script for Creating Database

Example of

```
<?php
$servername = "local host";
$username = "root";
$password = "";

$con = mysqle_connect ($servername, $username
                       $password);
$sql = "create database databases2";

mysqli_query($con, $sql);
```

#C

Page :
Date :

```
if( !$con)
{
   die("sorry fail to connect".mysqli_connect
                              _error());
}
else{
echo "Connection Successful";   // Not Mandatory
}
```

⇒ die() function is used to just show popup in the web page
This condition will run only if when the condition is false and it will not run if the condition is true.
⇒ exist() function is used to bring the control out of loop.

14/6/23

Php Script For Creating Table

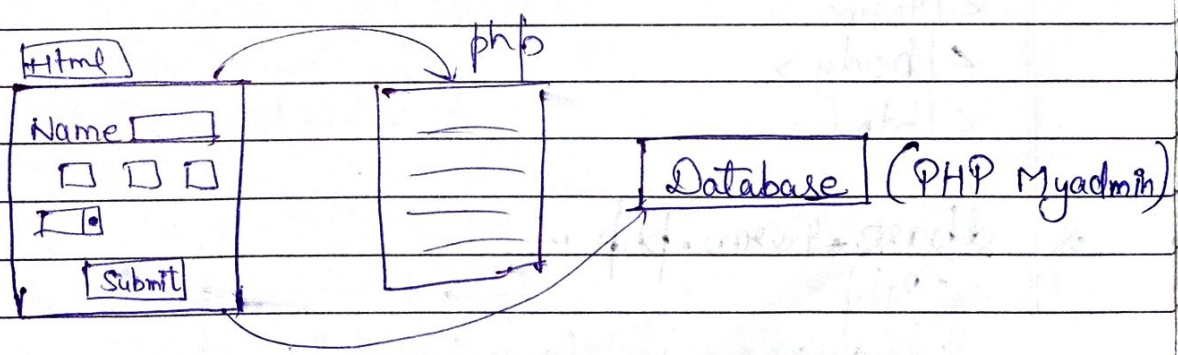
```
<?php

$servername = "local host";
$username = "root";
$password = "";
$database = "text 5";

$con = mysqli_connect($servername, $username,
                      $password, $database);
if(!$con)
{
die("Failed to connect");
}
```

```
$sql = "Create Table phpTable( Sno Int(6),
        Sname Varchar(20), branch Varchar(10),
        gender varchar(2));

$table = mysqli_query( $con, $sql);

echo var_dump($table);

if ($table)
{
echo "Table Created Successfully";
}
else
{
echo "Table not created successfully";
}
```

15/6/23

UNIT-4

Database → mysql_connect(), mysqli_query()
└→ Table →



Page :
Date :

* Form.PHP :-

```
<!DOCTYPE html>
<html lang="en">
 <head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content=
                                    "IE=edge">
 <meta name="viewport" content="width=device-
                                width, initial-scale=1.0">
   <title> Document </title>
</head>
<body>
<Form method="post" action="dbase_Form.php">
 serial no. : <input type="text" name="sn"><br>
 Name : <input type="text" name="fname">
                                    <br>
 Branch : <input type="text" name="Bname"><br>

 <input type="submit" value="click here"><br>

</Form>
</body>
</html>
```

* dbase_Form.php :-

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
$database = "test 5";
```

```
$con = mysqli_connect($servername, $username,
                      $password, $database);
if (!$con)
{
  die("Failed to connect");
}
$sql = "Insert Into phptable (Sno, Sname, branch)
VALUES ('$_POST[Sn]', '$_POST[Fname]', '$_POST
                                  [bname]')";

$insert = mysqli_query($con, $sql);
if($insert)
{
  echo "The record has been inserted <br>";
}
else
{
  echo "Record not Inserted";
}
?>
```

19/6/23

MySQL

PHP ↔ Database

- MySqli (Procedural Way)
  - → mysqli_connect()
  - → mysqli_query()
  - → mysqli_close()
- MySqli (Object Oriented)
  - → Class / Object
- PhP Data Object (P.D.O.)