Week End Exam – 1 (PHP)

1. Define Variable and DataType. Explain types of datatypes.

Ans: **Variable:**

A variable is a name given to the memory location where data is stored and can be accessed or modified during program execution.

Data Type:

Data type specifies the type of data a variable holds, such as integers, strings, booleans, arrays or objects

Types of data types:

In PHP, data types are classified into 3 main categories. They are

1. Scalar data types
2. Compound / Composite data types
3. Special data types

Scalar data types:

These data types are used to store a single value.

a. Integer – Used to store whole numbers.
   Ex: $a = 10;
b. String – Used to store a sequence of characters (text).
   Ex: $s = "Harry"
c. Boolean – Stores either "true" or "false".
   Ex: $b = true;
d. Float – Used to store decimal numbers.
   Ex: $f = 2.196;

Compound data types:

These data types can hold multiple values or grouped data.

a. Array - Stores multiple values in a single variable.
   Ex: $a = [1, 2, 3];
b. Object - Represents instances of classes, used in OOP.
   Ex: $s = new Student ();

Special data types:

These data types are used for special purposes.

a. Null – Represents a variable with no value.
   Ex: $x = NULL;
b. Resource – Refers to external sources such as file handlers or data base connections.
   Ex: $file = fopen("abc.txt", "r");

2. Define Operator and Expression. Explain types of operators.

Ans: **Operator:**

An Operator is a symbol that performs a specific operation on variables or values (operands) to compute a result.

Expression:

An expression is a combination of variables, values and operators that evaluates to a single result.

Types of operators:

There are 7 types of operators in PHP. They are

1. Arithmetic Operators
2. Assignment Operators
3. Logical Operators

4. Comparison Operators
5. Increment / Decrement Operators
6. Array Operators
7. String Operators

Arithmetic Operators:

Used to perform mathematical Operations.

a. Addition (+) – Used to add two values.
   Ex: 2+5 , $a + $b
b. Subtraction (-) – Used to subtract one value from another.
   Ex : 10 – 6 , $x - $y
c. Multiplication (*) – Used to multiply two values.
   Ex: 5 * 8 , $p * $q
d. Division (/) – Used to divide one value by another.
   Ex: 12 / 4, $x / $y
e. Modulus (%) – Used to find the remainder after division.
   Ex: 10 % 3, $a % $b

Assignment Operators:

Used to assign values to variables.

a. Assignment (=) - Used to assign a value to a variable.
   Ex *:* $x = 10
b. Add and assign (+=) - Adds a value to the variable and assigns the result to it.
   Ex*:* $x += 5 (same as $x = $x + 5)
c. Subtract and assign (-=) - Subtracts a value from the variable and stores the result.
   Example: $x -= 2
d. Multiply and assign (*=) - Multiplies the variable with a value and assigns the result.
   Ex: $x *= 3
e. Divide and assign (/=) -Divides the variable by a value and updates the result.
   Ex: $x /= 2

f. Modulus and assign (%=) - Assigns the remainder after dividing the variable.
Ex $x %= 4

Logical Operators:

Used to combine conditional statements.

a. And – Returns true only if both conditionals are true.

   Ex: if ($a > 0 and $a < 10) { echo "Between 1 and 9"; }

b. Or – Returns true if at least one condition is true.

   Ex: if ($a < 0 or $a > 100) { echo "Out of range"; }

c. Xor – Returns true only if one condition is true not both.

   Ex: if ($a == 5 xor $b == 5) { echo "Only one is 5"; }

d. && – Returns true only if both conditionals are true.

   Ex: if ($a > 0 && $a < 5) { echo "Between 1 and 5"; }

e. || – Returns true if atleast one condition is true.

   Ex: if ($a < 0 || $a > 100) { echo "Out of range"; }

f. ! – Reverses the result of the condition.

   Ex: if (!($a > 5)) { echo "a is not greater than 5"; }

Note:

Difference in and vs && and or vs ||

- and and or have **lower precedence** than =
- Use && and || in complex conditions to avoid logic bugs

Example:

```
<?php
$a = false;
$b = true;
$result = $a or $b;     // $result is false because = is evaluated first
$result = ($a or $b);   // $result is true
?>
```

Comparison Operators:

Used to compare two values (number or string).

a. Equal (==) - Checks if two values are equal.
   Ex: $a == $b
b. Not equal (!=) - Checks if two values are not equal.
   Ex: $a != $b
c. Greater than (>) - Checks if the left value is greater than the right.
   Ex: $x > $y
d. Less than (<) - Checks if the left value is less than the right.
   Ex: $x < $y
e. Greater than or equal to (>=) - Checks if a value is greater than or equal to another.
   Ex: $x >= $y
f. Less than or equal to (<=) - Checks if a value is less than or equal to another.
   Ex: $x <= $y
g. Identical (===) - Checks if two values are equal and of the same type.
   Ex: $a === $b

h. Not identical (!==) **-** Checks if values or types are not the same.
Ex: $a !== $b

Increment / Decrement Operators:

Used to increase or decrease the value of a variable by 1.

a. Pre-increment (++$x) – Increases the value **before** it is used.
Ex: ++$x

b. Post-increment ($x++) – Increases the value **after** it is used.
Ex: $x++

c. Pre-decrement (--$x) – Decreases the value **before** it is used.
Ex: --$x

d. Post-decrement ($x--) – Decreases the value **after** it is used.
Ex: $x - -

Array Operators:

Used to compare and combine arrays.

a. Union (+) – Combines two arrays.
Ex: $c = $a + $b

b. Equality (==) – Checks if two arrays have the same key-value pairs.
Ex: $a == $b

c. Identity (===) – Checks if two arrays are equal and have the same order and types.
Ex: $a === $b

d. Not equal (!=) – Checks if two arrays are not equal.
Ex: $a != $b

e. Not identical (!==) – Checks if two arrays are not exactly the same in order or type.
Ex: $a !== $b

String Operators:

Used to join or append strings.

a. Concatenation (.) – Used to join two strings.
Ex: $full = $first . $last

b. Concatenate and assign (.=) – Appends one string to another and stores the result.
Ex: $msg .= " world"