

4.1 How PHP Fits into a Web Page

Explanation

- PHP code is embedded inside HTML.
- PHP runs on the **server**, not in the browser.
- The browser receives only **HTML output**, not PHP code.

Complete Program Example

```
<!DOCTYPE html>
<html>
<head>
  <title>PHP in Web Page</title>
</head>
<body>

<h1>Welcome to My Website</h1>

<?php
  echo "This content is generated using PHP.<br>";
  echo "Server date is: " . date("d-m-Y");
?>

</body>
</html>
```

Output

```
Welcome to My Website
This content is generated using PHP.
Server date is: 15-01-2026
```

4.2 Variables and Constants

Explanation

- Variables store values temporarily.

- Constants store fixed values.
- PHP is loosely typed (no data type declaration needed).

Complete Program Example

```
<?php
// Variables
$name = "Amit";
$age = 21;
$percentage = 78.5;

// Constant
define("COLLEGE", "ABC Institute");

// Display values
echo "Name: $name <br>";
echo "Age: $age <br>";
echo "Percentage: $percentage <br>";
echo "College: " . COLLEGE;
?>
```

Output

```
Name: Amit
Age: 21
Percentage: 78.5
College: ABC Institute
```

4.3 Operators

Explanation

Operators are used to perform calculations and comparisons.

Complete Program Example

```
<?php
$a = 20;
$b = 10;

// Arithmetic operators
echo "Addition: " . ($a + $b) . "<br>";
echo "Subtraction: " . ($a - $b) . "<br>";
echo "Multiplication: " . ($a * $b) . "<br>";
echo "Division: " . ($a / $b) . "<br>";

// Comparison operator
if ($a > $b) {
    echo "$a is greater than $b <br>";
}

// Logical operator
if ($a > 5 && $b > 5) {
    echo "Both values are greater than 5";
}
?>
```

4.4 Working with Text and Numbers

Explanation

- PHP provides built-in functions for string and number manipulation.

Complete Program Example

```
<?php
$text = "PHP Programming";
$number = -25.7;

// String functions
echo "Length: " . strlen($text) . "<br>";
echo "Uppercase: " . strtoupper($text) . "<br>";
echo "Lowercase: " . strtolower($text) . "<br>";
```

```
// Number functions
echo "Absolute value: " . abs($number) . "<br>";
echo "Square root of 16: " . sqrt(16) . "<br>";
echo "2 power 3: " . pow(2, 3) . "<br>";
echo "Random number: " . rand(1, 100);
?>
```

4.5 Making Decisions with Control Statements

Explanation

Control statements control program flow using conditions and loops.

Complete Program Example

```
<?php
$marks = 68;

// if-elseif-else
if ($marks >= 75) {
    echo "Distinction<br>";
} elseif ($marks >= 60) {
    echo "First Class<br>";
} elseif ($marks >= 40) {
    echo "Pass<br>";
} else {
    echo "Fail<br>";
}

// switch statement
$day = 3;
switch ($day) {
    case 1: echo "Monday<br>"; break;
    case 2: echo "Tuesday<br>"; break;
    case 3: echo "Wednesday<br>"; break;
    default: echo "Invalid Day<br>";
}
}
```

```
// loop example
echo "Numbers from 1 to 5:<br>";
for ($i = 1; $i <= 5; $i++) {
    echo $i . "<br>";
}
?>
```

4.6 Working with Arrays, Strings, DateTime and Files

Arrays – Complete Program

```
<?php
// Indexed array
$fruits = array("Apple", "Banana", "Orange");

// Associative array
$marks = array("Math" => 90, "Science" => 85);

// Display arrays
foreach ($fruits as $fruit) {
    echo $fruit . "<br>";
}

echo "Math Marks: " . $marks["Math"];
?>
```

Date and Time – Complete Program

```
<?php
echo "Today Date: " . date("d-m-Y") . "<br>";
echo "Current Time: " . date("h:i:s A");
?>
```

File Handling – Complete Program

```
<?php
$file = fopen("sample.txt", "w");
fwrite($file, "This is file handling in PHP.");
fclose($file);

echo "File created and written successfully.";
?>
```

4.7 Functions

Explanation

- Functions are reusable blocks of code.
- They reduce duplication and improve readability.

Complete Program Example

```
<?php
// Function definition
function calculateSum($a, $b) {
    return $a + $b;
}

// Function call
$result = calculateSum(10, 20);
echo "Sum is: " . $result;
?>
```

Function with Default Argument

```
<?php
function greet($name = "User") {
    echo "Hello, $name!";
}
```

```
greet();  
greet("Rahul");  
?>
```