`Computational Sciences

Laboratory Manual

(Bachelor of Computer Application, 4th Semester, PHP and My SQL LAB, BCA49112)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Table of Contents

|  |  |  |
| --- | --- | --- |
| Experiment/Lab Exercise/Activity No. | Name of the Experiment/Activity/Exercise | Page  Number(s) |
| 1 | 1. Create a php webpage and print your name. 2. To write a program gets the name of the user from a form and show greeting text.      1. To write a program gets the name and phone no of the user from a form and show greeting text. | 8-10 |
| 2 | 1. Create a php program to find odd or even number from given number.      1. Write a php program to find maximum of three numbers.      1. Write a PHP program to swap two numbers | 10-12 |
| 3 | 1. Write a PHP program to compute give number multiplication table.      1. Write a program to implement mathematical calculator.      1. Write PHP program to display current date and display using getdate() function. | 12-14 |
| 4 | 1. Write a php program to check whether given number is palindrome or not.      1. Write a php program to check whether given number is Armstrong or not.      1. PHP Program to display digital clock with current time of the Server. | 14-16 |
| 5 | 1. Write a program that displays a different message based on time of day. For example page should display “Good Morning” if it is accessed in the morning.      1. Write a PHP script for the following: Design a form to accept a string. Write a function to count the total number of vowels (a,e,i,o,u) from the string. Show the occurrences of each vowel from the string.      1. Write a php program to check whether given number is String palindrome or not. | 16-18 |
| 6 | 1. Write a program to create Chess board in PHP using for loop      1. Write a program to calculate Electricity bill in PHP   Conditions:   * For first 50 units – Rs. 3.50/unit * For next 100 units – Rs. 4.00/unit * For next 100 units – Rs. 5.20/unit * For units above 250 – Rs. 6.50/unit   You can use conditional statements.     1. Write a PHP Script to find out the Sum of the Individual Digits. | 18-20 |
| 7 | 1. Write a PHP Script to find the factorial of a given number      1. Write a PHP Script to find the Fibonacci series of a given number.      1. Write an Age calculator program | 20-22 |
| 8 | 1. Write a PHP program to create an array and store 10 names in the array. Do the following operations.  * Display the contents using for each statement. * Display the array in a sorted order. * Display the array without the duplicate elements * Remove the last element and display * Display the array in reverse order * Insert a new element in a specified position * Search and find out position of an element from array      1. Write a PHP program to create a String and store value on that string “php Program - the complete SOLUTION for WEB development”. Do the following operations.  * convert lowercase string to uppercase * convert uppercase string to lowercase * convert first character uppercase * convert first character of all the words uppercase * get the last 11 characters of string * replace the first ‘the’ of the following string with ‘best’ * get the Second word of a sentence * insert a string at the specified position in a given string     24.Write a division table program in PHP using for loop. | 22-24 |
| 9 | 25. Write a program in PHP get array key from given array list    26. Write a program in PHP to get current directory, filename and code line number    27. Write a program in PHP to read from directory. | 24-26 |
| 10 | 28. Write a PHP program to Get Server Information.    29. Write a PHP program to detect remote device is mobile device or not.    30. To create a self-processing form, use the $\_SERVER['REQUEST\_METHOD'] that returns the request method e.g., GET or POST.   * If the $\_SERVER['REQUEST\_METHOD'] is GET, you show the form. * And if the $\_SERVER['REQUEST\_METHOD'] is POST, you process it.  |  |  | | --- | --- | |  |  | | 26-28 |
| 11 | 31. Write a PHP program to retrieve given URL information.    32. Write a PHP program to upload a file and get information about the upload file. | 29-31 |
| 12 | 33. Write a PHP program to upload an image file within “upload” folder. | 31-33 |
| 13 | 34. Create a form of four field as name, address, age and password. Input name, address, age and password through form. Then create a file with name and write name, address, age, password in that file. is\_file() function check the existence of file in “data” directory. | 33-35 |
| 14 | 35. Create a form have three input field namely File Name(text field), Message(text area),Task(list box) and a Done (submit button). A user will enter a file name and enter some text in Message (text area) then select Create Task from (list box) and submit the Done button. If file not available in data(directory) then a file will create with file name and content of Message will write in the file. if file not available in the data directory then a message will appear in Message(text area) that “ File Doesn’t exists .”. when user enter a file name and select Read from list box , if file is available then content of existing file will appear in message box. Again user will enter file name and select Append from list box then file will appended with content. | 35-37 |
| 15 | 36. Design a home page in PHP, an about page in PHP and header/menu file in php, home page and about page have same header. In header/menu file two link Home, About. If click on Home link, then open home page if click about page open about page using include function in php. | 37-39 |

**Assignment 1:**

1. Aim/Purpose of the Assignments:

Purpose of this assignment covers the roles of servers, local host servers, PHP, PHP syntax, operators, expressions, conditional branching and looping statements, and arrays in PHP.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory
2. Introduction to Web & Internet:
   * The World Wide Web (WWW) and the Internet are interconnected but distinct concepts.
   * The Internet is a global network of interconnected computers.
   * The Web is a collection of web pages, documents, and multimedia content accessible via the Internet through web browsers.
3. Introduction to Server:
   * A server is a computer or system that provides resources or services to other computers (clients) over a network.
   * Servers respond to requests from clients, delivering web pages, files, or other resources.
4. Local Host Server:
   * Localhost refers to the local machine that a user is currently working on.
   * A local server allows developers to test and develop web applications on their own machine before deploying them to a live server.
5. Starting PHP:
   * PHP (Hypertext Preprocessor) is a server-side scripting language widely used for web development.
   * PHP code is embedded within HTML, and it is executed on the server, producing dynamic web pages.
6. PHP Syntax and Variables:
   * PHP syntax is similar to C, Java, and Perl.
   * Variables in PHP start with a dollar sign ($), followed by the variable name.
   * PHP is loosely typed, meaning variable types are determined by their content.
7. Operators and Expressions:
   * PHP supports various operators, including arithmetic, assignment, comparison, and logical operators.
   * Expressions in PHP are combinations of values and operators that result in a single value.
8. Conditional Branching and Looping Statements:
   * Conditional statements (if, else, elseif) allow you to execute code based on certain conditions.
   * Looping statements (for, while, do-while) enable you to repeat code execution until a certain condition is met.
9. Learning Arrays in PHP:
   * Arrays in PHP are versatile and can store multiple values under a single variable name.
   * There are indexed arrays, associative arrays, and multidimensional arrays in PHP.
10. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Learn HTML and CSS.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
   1. What is the World Wide Web, and how does it differ from the internet?
   2. What is the role of a web server in the context of the internet?
   3. Differentiate between a web server and a database server.
   4. Explain the concept of a URL and its components.
   5. What does PHP stand for, and why is it commonly used in web development?
   6. Briefly explain the process of embedding PHP code in an HTML file.
   7. Describe the basic syntax rules in PHP.
   8. How do you declare and use variables in PHP?
2. Extension and Follow-up Activities (if applicable)
3. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 2:**

1. Aim/Purpose of the Assignments:

Purpose of this assignment covers the roles of servers, local host servers, PHP, PHP syntax, operators, expressions, conditional branching and looping statements, and arrays in PHP.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory
2. Introduction to Web & Internet:
   * The World Wide Web (WWW) and the Internet are interconnected but distinct concepts.
   * The Internet is a global network of interconnected computers.
   * The Web is a collection of web pages, documents, and multimedia content accessible via the Internet through web browsers.
3. Introduction to Server:
   * A server is a computer or system that provides resources or services to other computers (clients) over a network.
   * Servers respond to requests from clients, delivering web pages, files, or other resources.
4. Local Host Server:
   * Localhost refers to the local machine that a user is currently working on.
   * A local server allows developers to test and develop web applications on their own machine before deploying them to a live server.
5. Starting PHP:
   * PHP (Hypertext Preprocessor) is a server-side scripting language widely used for web development.
   * PHP code is embedded within HTML, and it is executed on the server, producing dynamic web pages.
6. PHP Syntax and Variables:
   * PHP syntax is similar to C, Java, and Perl.
   * Variables in PHP start with a dollar sign ($), followed by the variable name.
   * PHP is loosely typed, meaning variable types are determined by their content.
7. Operators and Expressions:
   * PHP supports various operators, including arithmetic, assignment, comparison, and logical operators.
   * Expressions in PHP are combinations of values and operators that result in a single value.
8. Conditional Branching and Looping Statements:
   * Conditional statements (if, else, elseif) allow you to execute code based on certain conditions.
   * Looping statements (for, while, do-while) enable you to repeat code execution until a certain condition is met.
9. Learning Arrays in PHP:
   * Arrays in PHP are versatile and can store multiple values under a single variable name.
   * There are indexed arrays, associative arrays, and multidimensional arrays in PHP.
10. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Learn HTML and CSS.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
   1. What is the World Wide Web, and how does it differ from the internet?
   2. What is the role of a web server in the context of the internet?
   3. Differentiate between a web server and a database server.
   4. Explain the concept of a URL and its components.
   5. What does PHP stand for, and why is it commonly used in web development?
   6. Briefly explain the process of embedding PHP code in an HTML file.
   7. Describe the basic syntax rules in PHP.

h. How do you declare and use variables in PHP?

1. Extension and Follow-up Activities (if applicable)
2. Assessments

As per the assessment and evaluation policy of the university.

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 3:**

1. Aim/Purpose of the Assignments:

Purpose of this assignment covers the roles of servers, local host servers, PHP, PHP syntax, operators, expressions, conditional branching and looping statements, and arrays in PHP.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory
2. Introduction to Web & Internet:
   * The World Wide Web (WWW) and the Internet are interconnected but distinct concepts.
   * The Internet is a global network of interconnected computers.
   * The Web is a collection of web pages, documents, and multimedia content accessible via the Internet through web browsers.
3. Introduction to Server:
   * A server is a computer or system that provides resources or services to other computers (clients) over a network.
   * Servers respond to requests from clients, delivering web pages, files, or other resources.
4. Local Host Server:
   * Localhost refers to the local machine that a user is currently working on.
   * A local server allows developers to test and develop web applications on their own machine before deploying them to a live server.
5. Starting PHP:
   * PHP (Hypertext Preprocessor) is a server-side scripting language widely used for web development.
   * PHP code is embedded within HTML, and it is executed on the server, producing dynamic web pages.
6. PHP Syntax and Variables:
   * PHP syntax is similar to C, Java, and Perl.
   * Variables in PHP start with a dollar sign ($), followed by the variable name.
   * PHP is loosely typed, meaning variable types are determined by their content.
7. Operators and Expressions:
   * PHP supports various operators, including arithmetic, assignment, comparison, and logical operators.
   * Expressions in PHP are combinations of values and operators that result in a single value.
8. Conditional Branching and Looping Statements:
   * Conditional statements (if, else, elseif) allow you to execute code based on certain conditions.
   * Looping statements (for, while, do-while) enable you to repeat code execution until a certain condition is met.
9. Learning Arrays in PHP:
   * Arrays in PHP are versatile and can store multiple values under a single variable name.
   * There are indexed arrays, associative arrays, and multidimensional arrays in PHP.
10. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Learn HTML and CSS.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
   1. What is the World Wide Web, and how does it differ from the internet?
   2. What is the role of a web server in the context of the internet?
   3. Differentiate between a web server and a database server.
   4. Explain the concept of a URL and its components.
   5. What does PHP stand for, and why is it commonly used in web development?
   6. Briefly explain the process of embedding PHP code in an HTML file.
   7. Describe the basic syntax rules in PHP.

h. How do you declare and use variables in PHP?

1. Extension and Follow-up Activities (if applicable)
2. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 4:**

1. Aim/Purpose of the Assignments:

Purpose of PHP offers built-in functions for manipulating strings and performing mathematical operations. HTML forms collect user input and can be accessed using $\_POST and $\_GET super global arrays.

applications.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. Theory of String and Math Functions in PHP:

* Understand and utilize PHP's built-in string functions for manipulating text and math functions for basic arithmetic operations.

2. Introduction to HTML Form Elements and Fields:

* Grasp the fundamentals of HTML form elements and fields, learning how to create and structure forms for user input.

3. Accessing PHP, HTTP Data, Query Strings, and Hyperlinks:

* Learn how PHP accesses data from HTTP requests, including query strings and parameters in hyperlinks.

4. Describing Predefined Variables - Super Global Arrays:

* Understand the concept of super global arrays in PHP (e.g., $\_GET, $\_POST, $\_REQUEST, $\_SESSION) and their significance in handling various types of data.

5. Understanding Functions:

* Gain insights into creating and using functions in PHP, including function declaration, parameters, and return values.

6. Important PHP Functions:

* Familiarize yourself with essential PHP functions, covering areas like string manipulation, array operations, date and time functions, and more.

7. Scope of Variables:

* Understand variable scope in PHP, including global and local scopes, and how it impacts the accessibility of variables.

8. Usage of Include and Require Statements:

* Learn how to modularize code by using include and require statements, allowing the reuse of code across multiple files.

9. Handling Files and Directories in PHP:

* Acquire skills in interacting with files and directories using PHP, including functions for file creation, reading, writing, and directory manipulation.

10. Fetching Information from Files: - Explore methods to retrieve information from files, such as reading and parsing content from text files or extracting data from other file formats.

11. Uploading and Downloading Files: - Learn how to implement file upload functionality in PHP for user-generated content and facilitate file downloads from the server.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Learn HTML and CSS.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
2. Provide examples of at least three string manipulation functions in PHP.
3. How can you concatenate strings in PHP?
4. Explain the usage of the strlen() function in PHP.
5. Discuss the purpose of the round() function in PHP.
6. What is the purpose of HTML form elements, and how are they structured?
7. Differentiate between the "GET" and "POST" methods in HTML forms.
8. Explain the role of the "action" attribute in an HTML form.
9. How can you access data sent through a form using PHP?
10. What is a query string, and how is it commonly used in URLs?
11. Provide an example of how you can pass data between pages using hyperlinks and query strings.
12. Extension and Follow-up Activities (if applicable)
13. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 5:**

1. Aim/Purpose of the Assignments:

Purpose of PHP offers built-in functions for manipulating strings and performing mathematical operations. HTML forms collect user input and can be accessed using $\_POST and $\_GET super global arrays.

applications.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. Theory of String and Math Functions in PHP:

* Understand and utilize PHP's built-in string functions for manipulating text and math functions for basic arithmetic operations.

2. Introduction to HTML Form Elements and Fields:

* Grasp the fundamentals of HTML form elements and fields, learning how to create and structure forms for user input.

3. Accessing PHP, HTTP Data, Query Strings, and Hyperlinks:

* Learn how PHP accesses data from HTTP requests, including query strings and parameters in hyperlinks.

4. Describing Predefined Variables - Super Global Arrays:

* Understand the concept of super global arrays in PHP (e.g., $\_GET, $\_POST, $\_REQUEST, $\_SESSION) and their significance in handling various types of data.

5. Understanding Functions:

* Gain insights into creating and using functions in PHP, including function declaration, parameters, and return values.

6. Important PHP Functions:

* Familiarize yourself with essential PHP functions, covering areas like string manipulation, array operations, date and time functions, and more.

7. Scope of Variables:

* Understand variable scope in PHP, including global and local scopes, and how it impacts the accessibility of variables.

8. Usage of Include and Require Statements:

* Learn how to modularize code by using include and require statements, allowing the reuse of code across multiple files.

9. Handling Files and Directories in PHP:

* Acquire skills in interacting with files and directories using PHP, including functions for file creation, reading, writing, and directory manipulation.

10. Fetching Information from Files: - Explore methods to retrieve information from files, such as reading and parsing content from text files or extracting data from other file formats.

11. Uploading and Downloading Files: - Learn how to implement file upload functionality in PHP for user-generated content and facilitate file downloads from the server.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Learn HTML and CSS.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
2. Provide examples of at least three string manipulation functions in PHP.
3. How can you concatenate strings in PHP?
4. Explain the usage of the strlen() function in PHP.
5. Discuss the purpose of the round() function in PHP.
6. What is the purpose of HTML form elements, and how are they structured?
7. Differentiate between the "GET" and "POST" methods in HTML forms.
8. Explain the role of the "action" attribute in an HTML form.
9. How can you access data sent through a form using PHP?
10. What is a query string, and how is it commonly used in URLs?
11. Provide an example of how you can pass data between pages using hyperlinks and query strings.
12. Extension and Follow-up Activities (if applicable)
13. Assessments

As per the assessment and evaluation policy of university

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 6:**

1. Aim/Purpose of the Assignments:

Purpose of PHP offers built-in functions for manipulating strings and performing mathematical operations. HTML forms collect user input and can be accessed using $\_POST and $\_GET super global arrays.

applications.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. Theory of String and Math Functions in PHP:

* Understand and utilize PHP's built-in string functions for manipulating text and math functions for basic arithmetic operations.

2. Introduction to HTML Form Elements and Fields:

* Grasp the fundamentals of HTML form elements and fields, learning how to create and structure forms for user input.

3. Accessing PHP, HTTP Data, Query Strings, and Hyperlinks:

* Learn how PHP accesses data from HTTP requests, including query strings and parameters in hyperlinks.

4. Describing Predefined Variables - Super Global Arrays:

* Understand the concept of super global arrays in PHP (e.g., $\_GET, $\_POST, $\_REQUEST, $\_SESSION) and their significance in handling various types of data.

5. Understanding Functions:

* Gain insights into creating and using functions in PHP, including function declaration, parameters, and return values.

6. Important PHP Functions:

* Familiarize yourself with essential PHP functions, covering areas like string manipulation, array operations, date and time functions, and more.

7. Scope of Variables:

* Understand variable scope in PHP, including global and local scopes, and how it impacts the accessibility of variables.

8. Usage of Include and Require Statements:

* Learn how to modularize code by using include and require statements, allowing the reuse of code across multiple files.

9. Handling Files and Directories in PHP:

* Acquire skills in interacting with files and directories using PHP, including functions for file creation, reading, writing, and directory manipulation.

10. Fetching Information from Files: - Explore methods to retrieve information from files, such as reading and parsing content from text files or extracting data from other file formats.

11. Uploading and Downloading Files: - Learn how to implement file upload functionality in PHP for user-generated content and facilitate file downloads from the server.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make simple PHP program.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
2. Provide examples of at least three string manipulation functions in PHP.
3. How can you concatenate strings in PHP?
4. Explain the usage of the strlen() function in PHP.
5. Discuss the purpose of the round() function in PHP.
6. What is the purpose of HTML form elements, and how are they structured?
7. Differentiate between the "GET" and "POST" methods in HTML forms.
8. Explain the role of the "action" attribute in an HTML form.
9. How can you access data sent through a form using PHP?
10. What is a query string, and how is it commonly used in URLs?
11. Provide an example of how you can pass data between pages using hyperlinks and query strings.
12. Extension and Follow-up Activities (if applicable)
13. Assessments

As per the assessment and evaluation policy of university

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 7:**

1. Aim/Purpose of the Assignments:

Aims is that the combination of PHP, MySQL, and HTML forms enables the creation of dynamic web applications where user input is collected through forms and seamlessly integrated with a MySQL database.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. PHP MySQL Connectivity:

Understand the concept of database connectivity in PHP, establishing a connection to MySQL databases. Learn about connection parameters, such as server name, username, password, and database name. Explore error handling and secure practices when connecting to MySQL databases using PHP.

2. Integrating Web Forms and Database:

Grasp the process of integrating HTML web forms with a MySQL database using PHP.

Learn how to capture user input from HTML forms and dynamically insert or update data in the MySQL database.

3. Using PHP’s MySQL Extension:

Familiarize yourself with PHP's MySQL extension for interacting with MySQL databases.

Understand the functions provided by the MySQL extension, such as mysqli\_connect, mysqli\_query, and mysqli\_fetch\_assoc.

4. Querying MySQL Database:

Learn how to construct and execute SQL queries in PHP to perform operations like SELECT, INSERT, UPDATE, and DELETE on the MySQL database. Understand the importance of parameterized queries to prevent SQL injection vulnerabilities.

5. Retrieving and Displaying Data:

Explore methods to retrieve data from MySQL databases using PHP and display it on web pages.

Understand how to iterate through result sets and format data for presentation.

6. Updating and Deleting Records:

Learn how to use PHP to update and delete records in a MySQL database based on user input.

Understand the importance of data validation and error handling during record modification.

7. Error Handling and Security:

Gain insights into handling errors that may occur during MySQL database interactions in PHP.

Implement security measures to protect against common vulnerabilities, such as SQL injection.

8. Connection Pooling and Optimization:

Understand the concept of connection pooling to improve the efficiency of database connections in PHP applications. Explore optimization techniques to enhance the performance of PHP MySQL interactions.

9. Transactions in PHP MySQL:

Learn about the use of transactions in PHP to ensure the atomicity and consistency of database operations. Understand how to begin, commit, and rollback transactions in PHP.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make simple PHP program.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
2. How do you establish a connection between PHP and MySQL?
3. Explain the significance of mysqli and PDO in PHP MySQL connectivity.
4. Provide an example of connecting to a MySQL database using the mysqli extension.
5. Describe the general process of integrating an HTML form with a MySQL database using PHP.
6. How do you secure user inputs from a form before inserting them into a database?
7. Explain the role of SQL injection and how to prevent it when interacting with databases.
8. What is the purpose of the mysql extension in PHP, and why is it deprecated?
9. Compare and contrast the mysql extension with mysqli and PDO.
10. Provide an example of using the mysql extension to perform a basic MySQL query.
11. Why are prepared statements important when working with databases in PHP?
12. How do you use prepared statements to prevent SQL injection in PHP?
13. Extension and Follow-up Activities (if applicable)
14. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 8:**

1. Aim/Purpose of the Assignments:

Aims is that the combination of PHP, MySQL, and HTML forms enables the creation of dynamic web applications where user input is collected through forms and seamlessly integrated with a MySQL database.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. PHP MySQL Connectivity:

Understand the concept of database connectivity in PHP, establishing a connection to MySQL databases. Learn about connection parameters, such as server name, username, password, and database name. Explore error handling and secure practices when connecting to MySQL databases using PHP.

2. Integrating Web Forms and Database:

Grasp the process of integrating HTML web forms with a MySQL database using PHP.

Learn how to capture user input from HTML forms and dynamically insert or update data in the MySQL database.

3. Using PHP’s MySQL Extension:

Familiarize yourself with PHP's MySQL extension for interacting with MySQL databases.

Understand the functions provided by the MySQL extension, such as mysqli\_connect, mysqli\_query, and mysqli\_fetch\_assoc.

4. Querying MySQL Database:

Learn how to construct and execute SQL queries in PHP to perform operations like SELECT, INSERT, UPDATE, and DELETE on the MySQL database. Understand the importance of parameterized queries to prevent SQL injection vulnerabilities.

5. Retrieving and Displaying Data:

Explore methods to retrieve data from MySQL databases using PHP and display it on web pages.

Understand how to iterate through result sets and format data for presentation.

6. Updating and Deleting Records:

Learn how to use PHP to update and delete records in a MySQL database based on user input.

Understand the importance of data validation and error handling during record modification.

7. Error Handling and Security:

Gain insights into handling errors that may occur during MySQL database interactions in PHP.

Implement security measures to protect against common vulnerabilities, such as SQL injection.

8. Connection Pooling and Optimization:

Understand the concept of connection pooling to improve the efficiency of database connections in PHP applications. Explore optimization techniques to enhance the performance of PHP MySQL interactions.

9. Transactions in PHP MySQL:

Learn about the use of transactions in PHP to ensure the atomicity and consistency of database operations. Understand how to begin, commit, and rollback transactions in PHP.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make simple PHP program.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
2. How do you establish a connection between PHP and MySQL?
3. Explain the significance of mysqli and PDO in PHP MySQL connectivity.
4. Provide an example of connecting to a MySQL database using the mysqli extension.
5. Describe the general process of integrating an HTML form with a MySQL database using PHP.
6. How do you secure user inputs from a form before inserting them into a database?
7. Explain the role of SQL injection and how to prevent it when interacting with databases.
8. What is the purpose of the mysql extension in PHP, and why is it deprecated?
9. Compare and contrast the mysql extension with mysqli and PDO.
10. Provide an example of using the mysql extension to perform a basic MySQL query.
11. Why are prepared statements important when working with databases in PHP?
12. How do you use prepared statements to prevent SQL injection in PHP?
13. Extension and Follow-up Activities (if applicable)
14. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 9:**

1. Aim/Purpose of the Assignments:

Aims is that the combination of PHP, MySQL, and HTML forms enables the creation of dynamic web applications where user input is collected through forms and seamlessly integrated with a MySQL database.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. PHP MySQL Connectivity:

Understand the concept of database connectivity in PHP, establishing a connection to MySQL databases. Learn about connection parameters, such as server name, username, password, and database name. Explore error handling and secure practices when connecting to MySQL databases using PHP.

2. Integrating Web Forms and Database:

Grasp the process of integrating HTML web forms with a MySQL database using PHP.

Learn how to capture user input from HTML forms and dynamically insert or update data in the MySQL database.

3. Using PHP’s MySQL Extension:

Familiarize yourself with PHP's MySQL extension for interacting with MySQL databases.

Understand the functions provided by the MySQL extension, such as mysqli\_connect, mysqli\_query, and mysqli\_fetch\_assoc.

4. Querying MySQL Database:

Learn how to construct and execute SQL queries in PHP to perform operations like SELECT, INSERT, UPDATE, and DELETE on the MySQL database. Understand the importance of parameterized queries to prevent SQL injection vulnerabilities.

5. Retrieving and Displaying Data:

Explore methods to retrieve data from MySQL databases using PHP and display it on web pages.

Understand how to iterate through result sets and format data for presentation.

6. Updating and Deleting Records:

Learn how to use PHP to update and delete records in a MySQL database based on user input.

Understand the importance of data validation and error handling during record modification.

7. Error Handling and Security:

Gain insights into handling errors that may occur during MySQL database interactions in PHP.

Implement security measures to protect against common vulnerabilities, such as SQL injection.

8. Connection Pooling and Optimization:

Understand the concept of connection pooling to improve the efficiency of database connections in PHP applications. Explore optimization techniques to enhance the performance of PHP MySQL interactions.

9. Transactions in PHP MySQL:

Learn about the use of transactions in PHP to ensure the atomicity and consistency of database operations. Understand how to begin, commit, and rollback transactions in PHP.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make simple PHP program.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
2. How do you establish a connection between PHP and MySQL?
3. Explain the significance of mysqli and PDO in PHP MySQL connectivity.
4. Provide an example of connecting to a MySQL database using the mysqli extension.
5. Describe the general process of integrating an HTML form with a MySQL database using PHP.
6. How do you secure user inputs from a form before inserting them into a database?
7. Explain the role of SQL injection and how to prevent it when interacting with databases.
8. What is the purpose of the mysql extension in PHP, and why is it deprecated?
9. Compare and contrast the mysql extension with mysqli and PDO.
10. Provide an example of using the mysql extension to perform a basic MySQL query.
11. Why are prepared statements important when working with databases in PHP?
12. How do you use prepared statements to prevent SQL injection in PHP?
13. Extension and Follow-up Activities (if applicable)
14. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 10:**

1. Aim/Purpose of the Assignments:

Understanding and applying these OOP concepts leads to more modular, maintainable, and scalable code. They are fundamental in modern programming paradigms and are widely used in languages like Java, C++, Python, and PHP

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. Introduction to Object-Oriented Programming (OOP):

Understand the core principles of OOP, which include encapsulation, inheritance, and polymorphism.

Grasp the concept of organizing code around objects that represent real-world entities.

2. Classes and Objects:

Learn the distinction between classes (blueprints for objects) and objects (instances of classes).

Understand how classes define attributes (properties) and behaviors (methods) of objects.

3. Abstraction:

Recognize the concept of abstraction in OOP, where essential features are captured while unnecessary details are hidden. Understand the use of abstract classes and interfaces to achieve abstraction.

4. Encapsulation:

Understand encapsulation as the bundling of data (attributes) and methods (functions) that operate on the data within a single unit (class). Learn how access modifiers like public, private, and protected control the visibility of class members.

5. Inheritance:

Grasp the concept of inheritance, allowing a class (subclass or derived class) to inherit properties and behaviors from another class (base class or parent class). Understand the benefits of code reusability and the "is-a" relationship between classes.

6. Polymorphism:

Understand polymorphism as the ability of objects of different classes to respond to the same method name. Explore method overloading and method overriding as forms of polymorphism.

7. Constructors and Destructors:

Learn the role of constructors in initializing object properties and performing setup tasks when an object is created. Understand destructors for performing cleanup tasks when an object is destroyed or goes out of scope.

8. Overloading:

Understand method overloading, where a class can have multiple methods with the same name but different parameter lists. Recognize the flexibility it provides when working with varying numbers or types of parameters.

9. Overriding of Functions:

Grasp the concept of method overriding, allowing a subclass to provide a specific implementation of a method defined in its superclass. Understand how overriding supports polymorphism and enhances flexibility in the inheritance hierarchy.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make simple PHP program.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions

Assignment 10-12:

1. What is Object-Oriented Programming (OOP), and how does it differ from procedural programming?
2. Name and explain the four main principles of OOP.
3. Define a class and an object in the context of OOP.
4. How are properties and methods related to classes and objects in PHP?
5. Explain the concept of abstraction in OOP.
6. Provide an example of abstraction in a PHP class.
7. Define encapsulation and describe its benefits in OOP.
8. How does encapsulation contribute to information hiding in a class?
9. What is inheritance, and how does it promote code reuse in OOP?
10. Explain polymorphism and provide an example of its usage in PHP.
11. What is the purpose of a constructor in a PHP class?
12. Discuss the role of destructors in PHP classes and when they are invoked.
13. Extension and Follow-up Activities (if applicable)
14. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 11:**

1. Aim/Purpose of the Assignments:

Understanding and applying these OOP concepts leads to more modular, maintainable, and scalable code. They are fundamental in modern programming paradigms and are widely used in languages like Java, C++, Python, and PHP

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. Introduction to Object-Oriented Programming (OOP):

Understand the core principles of OOP, which include encapsulation, inheritance, and polymorphism.

Grasp the concept of organizing code around objects that represent real-world entities.

2. Classes and Objects:

Learn the distinction between classes (blueprints for objects) and objects (instances of classes).

Understand how classes define attributes (properties) and behaviors (methods) of objects.

3. Abstraction:

Recognize the concept of abstraction in OOP, where essential features are captured while unnecessary details are hidden. Understand the use of abstract classes and interfaces to achieve abstraction.

4. Encapsulation:

Understand encapsulation as the bundling of data (attributes) and methods (functions) that operate on the data within a single unit (class). Learn how access modifiers like public, private, and protected control the visibility of class members.

5. Inheritance:

Grasp the concept of inheritance, allowing a class (subclass or derived class) to inherit properties and behaviors from another class (base class or parent class). Understand the benefits of code reusability and the "is-a" relationship between classes.

6. Polymorphism:

Understand polymorphism as the ability of objects of different classes to respond to the same method name. Explore method overloading and method overriding as forms of polymorphism.

7. Constructors and Destructors:

Learn the role of constructors in initializing object properties and performing setup tasks when an object is created. Understand destructors for performing cleanup tasks when an object is destroyed or goes out of scope.

8. Overloading:

Understand method overloading, where a class can have multiple methods with the same name but different parameter lists. Recognize the flexibility it provides when working with varying numbers or types of parameters.

9. Overriding of Functions:

Grasp the concept of method overriding, allowing a subclass to provide a specific implementation of a method defined in its superclass. Understand how overriding supports polymorphism and enhances flexibility in the inheritance hierarchy.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make database connection.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions

Assignment 10-12:

1. What is Object-Oriented Programming (OOP), and how does it differ from procedural programming?
2. Name and explain the four main principles of OOP.
3. Define a class and an object in the context of OOP.
4. How are properties and methods related to classes and objects in PHP?
5. Explain the concept of abstraction in OOP.
6. Provide an example of abstraction in a PHP class.
7. Define encapsulation and describe its benefits in OOP.
8. How does encapsulation contribute to information hiding in a class?
9. What is inheritance, and how does it promote code reuse in OOP?
10. Explain polymorphism and provide an example of its usage in PHP.
11. What is the purpose of a constructor in a PHP class?
12. Discuss the role of destructors in PHP classes and when they are invoked.
13. Extension and Follow-up Activities (if applicable)
14. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 12:**

1. Aim/Purpose of the Assignments:

Understanding and applying these OOP concepts leads to more modular, maintainable, and scalable code. They are fundamental in modern programming paradigms and are widely used in languages like Java, C++, Python, and PHP

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. Introduction to Object-Oriented Programming (OOP):

Understand the core principles of OOP, which include encapsulation, inheritance, and polymorphism.

Grasp the concept of organizing code around objects that represent real-world entities.

2. Classes and Objects:

Learn the distinction between classes (blueprints for objects) and objects (instances of classes).

Understand how classes define attributes (properties) and behaviors (methods) of objects.

3. Abstraction:

Recognize the concept of abstraction in OOP, where essential features are captured while unnecessary details are hidden. Understand the use of abstract classes and interfaces to achieve abstraction.

4. Encapsulation:

Understand encapsulation as the bundling of data (attributes) and methods (functions) that operate on the data within a single unit (class). Learn how access modifiers like public, private, and protected control the visibility of class members.

5. Inheritance:

Grasp the concept of inheritance, allowing a class (subclass or derived class) to inherit properties and behaviors from another class (base class or parent class). Understand the benefits of code reusability and the "is-a" relationship between classes.

6. Polymorphism:

Understand polymorphism as the ability of objects of different classes to respond to the same method name. Explore method overloading and method overriding as forms of polymorphism.

7. Constructors and Destructors:

Learn the role of constructors in initializing object properties and performing setup tasks when an object is created. Understand destructors for performing cleanup tasks when an object is destroyed or goes out of scope.

8. Overloading:

Understand method overloading, where a class can have multiple methods with the same name but different parameter lists. Recognize the flexibility it provides when working with varying numbers or types of parameters.

9. Overriding of Functions:

Grasp the concept of method overriding, allowing a subclass to provide a specific implementation of a method defined in its superclass. Understand how overriding supports polymorphism and enhances flexibility in the inheritance hierarchy.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make database connection.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions

Assignment 10-12:

1. What is Object-Oriented Programming (OOP), and how does it differ from procedural programming?
2. Name and explain the four main principles of OOP.
3. Define a class and an object in the context of OOP.
4. How are properties and methods related to classes and objects in PHP?
5. Explain the concept of abstraction in OOP.
6. Provide an example of abstraction in a PHP class.
7. Define encapsulation and describe its benefits in OOP.
8. How does encapsulation contribute to information hiding in a class?
9. What is inheritance, and how does it promote code reuse in OOP?
10. Explain polymorphism and provide an example of its usage in PHP.
11. What is the purpose of a constructor in a PHP class?
12. Discuss the role of destructors in PHP classes and when they are invoked.
13. Extension and Follow-up Activities (if applicable)
14. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 13:**

1. Aim/Purpose of the Assignments:

Understanding these MySQL concepts and commands is crucial for developers and database administrators to interact effectively with databases, ensuring proper data management and retrieval in various scenarios. These skills are essential for building robust and efficient database-driven applications.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. MySQL Database Architecture:

* Understand the MySQL database architecture, consisting of storage engines, query processing, and how data is organized in tables.

2. System-Defined Tables:

* Explore system-defined tables (e.g., information\_schema) that store metadata about the database, tables, and other objects.

3. CREATE, ALTER, DELETE, DROP Tables:

* Learn SQL commands for creating tables (CREATE), altering table structure (ALTER), deleting table data (DELETE), and dropping tables (DROP) altogether.

4. INSERT, UPDATE, DELETE Table Data:

* Understand how to insert new records (INSERT), update existing records (UPDATE), and delete records (DELETE) from tables.

5. WHERE Conditions:

* Grasp the use of the WHERE clause to filter rows based on specified conditions in SELECT, UPDATE, and DELETE statements.

6. AND, OR, IN, BETWEEN, LIKE, DISTINCT:

* Understand logical operators (AND, OR) and use of IN, BETWEEN, LIKE, and DISTINCT to filter and manipulate data in SQL queries.

7. ORDER BY:

* Learn the ORDER BY clause to sort query results in ascending or descending order based on specified columns.

8. GROUP BY:

* Understand the GROUP BY clause to group rows based on common values, often used with aggregate functions like COUNT, SUM, AVG.

9. UNION:

* Explore the UNION operator to combine results of multiple SELECT statements, ensuring the results are distinct.

10. Sub-queries: - Grasp the concept of sub-queries, allowing nested queries within other queries to perform complex operations.

11. LEFT JOIN, RIGHT JOIN, INNER JOIN: - Understand different types of joins (LEFT JOIN, RIGHT JOIN, INNER JOIN) to retrieve data from multiple tables based on specified conditions.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make database connection.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
2. Explain the basic architecture of a MySQL database.
3. What is the role of the MySQL Storage Engine, and name a few popular storage engines?
4. What are system-defined tables in MySQL, and why are they important?
5. Provide examples of commonly used system-defined tables.
6. How do you create a new table in MySQL using the CREATE TABLE statement?
7. Explain the purpose of the ALTER TABLE statement in MySQL.
8. Differentiate between the DELETE and DROP statements in MySQL.
9. Provide examples of using the INSERT INTO, UPDATE, and DELETE statements in MySQL.
10. How can you use the DELETE statement with a WHERE clause to selectively remove records?
11. Explain the significance of the WHERE clause in MySQL.
12. Differentiate between the AND and OR operators in WHERE conditions.
13. How does the IN clause work in MySQL, and provide an example.
14. Give an example of using the BETWEEN operator in a WHERE condition.
15. How is the LIKE operator used in MySQL queries?
16. Explain the purpose of the DISTINCT keyword in MySQL.
17. How can you use the ORDER BY clause to sort query results?
18. Discuss the usage of the GROUP BY clause in MySQL.
19. Extension and Follow-up Activities (if applicable)
20. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 14:**

1. Aim/Purpose of the Assignments:

Understanding these MySQL concepts and commands is crucial for developers and database administrators to interact effectively with databases, ensuring proper data management and retrieval in various scenarios. These skills are essential for building robust and efficient database-driven applications.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. MySQL Database Architecture:

* Understand the MySQL database architecture, consisting of storage engines, query processing, and how data is organized in tables.

2. System-Defined Tables:

* Explore system-defined tables (e.g., information\_schema) that store metadata about the database, tables, and other objects.

3. CREATE, ALTER, DELETE, DROP Tables:

* Learn SQL commands for creating tables (CREATE), altering table structure (ALTER), deleting table data (DELETE), and dropping tables (DROP) altogether.

4. INSERT, UPDATE, DELETE Table Data:

* Understand how to insert new records (INSERT), update existing records (UPDATE), and delete records (DELETE) from tables.

5. WHERE Conditions:

* Grasp the use of the WHERE clause to filter rows based on specified conditions in SELECT, UPDATE, and DELETE statements.

6. AND, OR, IN, BETWEEN, LIKE, DISTINCT:

* Understand logical operators (AND, OR) and use of IN, BETWEEN, LIKE, and DISTINCT to filter and manipulate data in SQL queries.

7. ORDER BY:

* Learn the ORDER BY clause to sort query results in ascending or descending order based on specified columns.

8. GROUP BY:

* Understand the GROUP BY clause to group rows based on common values, often used with aggregate functions like COUNT, SUM, AVG.

9. UNION:

* Explore the UNION operator to combine results of multiple SELECT statements, ensuring the results are distinct.

10. Sub-queries: - Grasp the concept of sub-queries, allowing nested queries within other queries to perform complex operations.

11. LEFT JOIN, RIGHT JOIN, INNER JOIN: - Understand different types of joins (LEFT JOIN, RIGHT JOIN, INNER JOIN) to retrieve data from multiple tables based on specified conditions.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make database connection.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
2. Explain the basic architecture of a MySQL database.
3. What is the role of the MySQL Storage Engine, and name a few popular storage engines?
4. What are system-defined tables in MySQL, and why are they important?
5. Provide examples of commonly used system-defined tables.
6. How do you create a new table in MySQL using the CREATE TABLE statement?
7. Explain the purpose of the ALTER TABLE statement in MySQL.
8. Differentiate between the DELETE and DROP statements in MySQL.
9. Provide examples of using the INSERT INTO, UPDATE, and DELETE statements in MySQL.
10. How can you use the DELETE statement with a WHERE clause to selectively remove records?
11. Explain the significance of the WHERE clause in MySQL.
12. Differentiate between the AND and OR operators in WHERE conditions.
13. How does the IN clause work in MySQL, and provide an example.
14. Give an example of using the BETWEEN operator in a WHERE condition.
15. How is the LIKE operator used in MySQL queries?
16. Explain the purpose of the DISTINCT keyword in MySQL.
17. How can you use the ORDER BY clause to sort query results?
18. Discuss the usage of the GROUP BY clause in MySQL.
19. Extension and Follow-up Activities (if applicable)
20. Assessments

As per the assessment and evaluation policy of University

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013

**Assignment 15:**

1. Aim/Purpose of the Assignments:

Understanding these MySQL concepts and commands is crucial for developers and database administrators to interact effectively with databases, ensuring proper data management and retrieval in various scenarios. These skills are essential for building robust and efficient database-driven applications.

1. Learning Outcomes

* Develop a foundational understanding of PHP as a server-side scripting language for web development.
* Be familiar with proficient in handling PHP functions for string and math operations, designing HTML forms, accessing and manipulating PHP and HTTP.
* To understand and participants should be well-equipped to design, develop, and maintain dynamic web applications that interact with MySQL databases using PHP.
* Understanding these concepts is essential for modern software development in languages that support object-oriented programming paradigms such as Java, C++, Python, and PHP.
* Be familiar to design and manipulate database tables, write optimized queries for data retrieval, and perform complex operations such as joins and sub-queries.

1. Prerequisites:

Able to making design web pages and connect to database server.

1. Software required:

XAMPP

1. Introduction and Theory

1. MySQL Database Architecture:

* Understand the MySQL database architecture, consisting of storage engines, query processing, and how data is organized in tables.

2. System-Defined Tables:

* Explore system-defined tables (e.g., information\_schema) that store metadata about the database, tables, and other objects.

3. CREATE, ALTER, DELETE, DROP Tables:

* Learn SQL commands for creating tables (CREATE), altering table structure (ALTER), deleting table data (DELETE), and dropping tables (DROP) altogether.

4. INSERT, UPDATE, DELETE Table Data:

* Understand how to insert new records (INSERT), update existing records (UPDATE), and delete records (DELETE) from tables.

5. WHERE Conditions:

* Grasp the use of the WHERE clause to filter rows based on specified conditions in SELECT, UPDATE, and DELETE statements.

6. AND, OR, IN, BETWEEN, LIKE, DISTINCT:

* Understand logical operators (AND, OR) and use of IN, BETWEEN, LIKE, and DISTINCT to filter and manipulate data in SQL queries.

7. ORDER BY:

* Learn the ORDER BY clause to sort query results in ascending or descending order based on specified columns.

8. GROUP BY:

* Understand the GROUP BY clause to group rows based on common values, often used with aggregate functions like COUNT, SUM, AVG.

9. UNION:

* Explore the UNION operator to combine results of multiple SELECT statements, ensuring the results are distinct.

10. Sub-queries: - Grasp the concept of sub-queries, allowing nested queries within other queries to perform complex operations.

11. LEFT JOIN, RIGHT JOIN, INNER JOIN: - Understand different types of joins (LEFT JOIN, RIGHT JOIN, INNER JOIN) to retrieve data from multiple tables based on specified conditions.

1. Operating Procedure:

Standard operating procedure for using XAMPP as per the instruction of the faculty member

1. Precautions and/or Troubleshooting

Debugging while running the program

1. Observations

Take different possible inputs and observe the output

1. Calculations & Analysis

Able to make database connection.

1. Result & Interpretation

The result or output have to be shown in computer and write it in the assignment copy

1. Follow-up Questions
2. Explain the basic architecture of a MySQL database.
3. What is the role of the MySQL Storage Engine, and name a few popular storage engines?
4. What is system-defined tables in MySQL, and why are they important?
5. Provide examples of commonly used system-defined tables.
6. How do you create a new table in MySQL using the CREATE TABLE statement?
7. Explain the purpose of the ALTER TABLE statement in MySQL.
8. Differentiate between the DELETE and DROP statements in MySQL.
9. Provide examples of using the INSERT INTO, UPDATE, and DELETE statements in MySQL.
10. How can you use the DELETE statement with a WHERE clause to selectively remove records?
11. Explain the significance of the WHERE clause in MySQL.
12. Differentiate between the AND and OR operators in WHERE conditions.
13. How does the IN-clause work in MySQL, and provide an example?
14. Give an example of using the BETWEEN operator in a WHERE condition.
15. How is the LIKE operator used in MySQL queries?
16. Explain the purpose of the DISTINCT keyword in MySQL.
17. How can you use the ORDER BY clause to sort query results?
18. Discuss the usage of the GROUP BY clause in MySQL.
19. Extension and Follow-up Activities (if applicable)
20. Assessments

As per the assessment and evaluation policy of university

1. Suggested readings

1. PHP and MySQL Web Development Luke Welling, Laura Thomson, Pearson, 5th Edition, 2016

2. Programming PHP: Creating Dynamic Web Pages Kevin Tatroe,Peter MacIntyre, Rasmus Lerdorf, O'Reilly, 3rd Edition, 2013