UNIT 4:

Server Site Programming:

Introduction to Active Server Pages (ASP) and ASP.NET:

1. **Active Server Pages (ASP):**

- ASP is a server-side scripting technology developed by Microsoft for building dynamic web pages.

- It allows embedding server-side code (usually VBScript or JScript) directly into HTML pages, enabling dynamic content generation.

2. **ASP.NET:**

- ASP.NET is an evolution of ASP and a part of the Microsoft .NET framework.

- It provides a more robust and object-oriented approach to web development, supporting multiple languages such as C# and VB.NET.

****JavaServer Pages (JSP) and JSP Application Design:****

1. **JavaServer Pages (JSP):**

- JSP is a server-side technology developed by Sun Microsystems (now Oracle) for building dynamic web pages using Java.

- It allows embedding Java code into HTML pages, making it easier to write and maintain server-side logic.

2. **JSP Application Design:**

- JSP applications are designed based on the Model-View-Controller (MVC) architecture, separating the business logic (Model), user interface (View), and application flow (Controller).

- The Model represents the data and application logic, the View handles the presentation of data to users, and the Controller manages the user input and application flow.

****Tomcat Server and JSP Objects:****

1. **Tomcat Server:**

- Apache Tomcat is an open-source web server and servlet container used to host Java web applications, including JSP applications.

- It provides a runtime environment for JSP pages and servlets to be executed.

2. **JSP Objects:**

- JSP provides several predefined objects, collectively known as implicit objects, that can be accessed directly in JSP pages without explicit declaration.

- Common implicit objects include `request`, `response`, `session`, `application`, `out`, `config`, and `pageContext`.

Declaring Variables and Methods, Debugging, and Sharing Data between JSP Pages:

1. **Declaring Variables and Methods:**

- In JSP, you can declare variables using scriptlet tags (`<% %>`) and expression tags (`<%= %>`).

- Methods can be defined within scriptlet tags using Java syntax.

2. **Debugging:**

- Debugging JSP can be challenging since it involves both Java and HTML code.

- Logging and error handling techniques can be used to identify and fix issues in JSP applications.

3. **Sharing Data between JSP Pages:**

- Data can be shared between JSP pages using request attributes, session attributes, and application attributes.

- Request attributes are used to pass data between JSP pages during a single request, while session attributes maintain data across multiple requests from the same user. Application attributes are shared across all users.

****Session and Application: Database Actions, Development of JavaBeans in JSP:****

1. **Session and Application:**

- Session and application objects allow storing data in memory on the server to be shared among multiple users.

- Sessions are user-specific and maintain data during a user's visit, while application objects store data that is accessible by all users of the application.

2. **Database Actions:**

- JSP applications often interact with databases to store and retrieve data.

- JDBC (Java Database Connectivity) is commonly used to connect and perform database operations in JSP.

3. **Development of JavaBeans in JSP:**

- JavaBeans are reusable Java components that follow specific conventions for property accessors and mutators.

- They can be developed and utilized in JSP applications to encapsulate business logic and promote code reusability.

****Introduction to COM/DCOM:****

1. **COM (Component Object Model):**

- COM is a Microsoft technology that enables software components to communicate and interact with each other in a distributed environment.

- It provides a standard way for software components to be reused and integrated into different applications.

2. **DCOM (Distributed Component Object Model):**

- DCOM extends COM to support communication between software components over a network in a distributed computing environment.

- DCOM enables components running on different machines to interact transparently as if they were local components.

In summary, ASP and ASP.NET are Microsoft's server-side scripting technologies, while JSP is a Java-based server-side technology. Both ASP.NET and JSP follow the

MVC architecture for designing web applications. Tomcat is a popular servlet container used to host JSP applications. JSP applications can use various objects to handle data sharing and can interact with databases through JDBC. Additionally, JavaBeans can be employed for encapsulating business logic and promoting reusability in JSP applications. Finally, COM/DCOM are Microsoft technologies for facilitating component-based communication in distributed environments.

UNIT 5: PHP

(Hypertext Preprocessor):

Introduction, syntax, variables, strings, operators, if-else, loop, switch, array, function, form ,mail, file upload, session, error, exception, filter, PHP-ODBC.

****Introduction to PHP:****

PHP (Hypertext Preprocessor) is a widely used server-side scripting language designed for web development. It is embedded within HTML code and executed on the server, generating dynamic web pages. PHP is open-source, easy to learn, and has a large community of developers, making it a popular choice for web application development.

****Syntax:****

PHP code is enclosed within `<?php ?>` tags. For example:

```php
<?php
  // PHP code goes here
?>
```

Variables:

Variables in PHP start with a dollar sign (`$`) followed by the variable name. They do not require explicit data type declaration. Example:

```php
$name = "John Doe";
$age = 30;
```

Strings:

Strings are sequences of characters and can be enclosed in single or double quotes. Example:

```php
$greeting = "Hello, world!";
```

Operators:

PHP supports various types of operators, including arithmetic, assignment, comparison, logical, etc. Example:

```php
$sum = 5 + 3;      // Arithmetic operator
$isTrue = true;    // Assignment operator
$isEqual = 10 == 5; // Comparison operator
```

if-else:

The `if-else` statement is used for conditional execution in PHP. Example:

```php
if ($age >= 18) {
  echo "You are an adult.";
} else {
  echo "You are a minor.";
}
```

Loop:

PHP provides various loop constructs like `for`, `while`, `do-while`, and `foreach`. Example:

```php
for ($i = 1; $i <= 5; $i++) {
  echo $i . " ";
}
```

Switch:

The `switch` statement allows for multi-way conditional branching based on a variable's value. Example:

```php
$day = "Monday";

switch ($day) {
  case "Monday":
    echo "Today is Monday.";
    break;
  case "Tuesday":
```

```
    echo "Today is Tuesday.";
    break;
  // ... Other cases ...
  default:
    echo "It's another day.";
}
```

Array:

Arrays in PHP can store multiple values in a single variable. Example:

```php
$fruits = array("apple", "banana", "orange");
```

Function:

Functions in PHP are blocks of code that can be called multiple times. Example:

```php
function add($a, $b) {
  return $a + $b;
```

}

$result = add(5, 3); // Result will be 8

```

****Form Handling and Mail:****

PHP can process data submitted from HTML forms using `$_POST` or `$_GET` superglobal arrays. Example:

```php

if ($_SERVER["REQUEST_METHOD"] == "POST") {

$name = $_POST["name"];

$email = $_POST["email"];

// Process form data

// Send email

}

```

****File Upload:****

PHP allows users to upload files through HTML forms. Uploaded files can be accessed using `$_FILES` superglobal array. Example:

```php
$targetDir = "uploads/";
$uploadedFile = $targetDir . basename($_FILES["file"]["name"]);

if (move_uploaded_file($_FILES["file"]["tmp_name"], $uploadedFile)) {
  echo "File uploaded successfully.";
} else {
  echo "Error uploading file.";
}
```

Session, Error, Exception, and Filter:

- Session: PHP sessions allow persisting data across multiple requests for a specific user.
- Error: PHP provides error handling functions to customize error reporting and handling.
- Exception: PHP supports exception handling using `try`, `catch`, and `throw`.
- Filter: PHP filters allow data validation and sanitization to prevent malicious inputs.

PHP-ODBC: