



Question 1

Question 1a (i)

Identify the three (3) parts that make up this URL and briefly describe each part.

1. Part 1: Protocol ([https://](#))
 - **Description:** This specifies the protocol used to access the resource on the internet. In this case, 'https' stands for Hypertext Transfer Protocol Secure, which is a secure version of HTTP used for secure communication over a computer network.
2. Part 2: Domain Name and Port ([www.open.ac.mu:80](#))
 - **Description:** The domain name ([www.open.ac.mu](#)) identifies the server where the resource is located. It is a human-readable address that is translated to an IP address by the DNS (Domain Name System). The port number (:80) specifies the specific port on the server where the resource can be accessed. Port 80 is the default port for HTTP traffic.
3. Part 3: Path ([/index.htm](#))
 - **Description:** This specifies the exact resource or file to be retrieved from the server. In this case, 'index.htm' is the file located at the root directory of the server.

Question 1a (ii)

Draw the tree diagram of the domain namespace for the above URL. Show the root and the branches.

Here's a textual representation of the tree diagram:

```
.
└─ mu (root)
    └─ ac
        └─ open
            └─ www
                └─ index.htm
```

Question 1a (iii)

Explain how the webpage is obtained from the webserver (host) when you request the home page of the Open University website (www.open.ac.mu) from your browser, including the role of DNS and ISP in this process.

1. **Browser Request:** The user types the URL `https://www.open.ac.mu:80/index.htm` into the browser and presses Enter.
2. **DNS Lookup:** The browser checks its cache for the IP address of `www.open.ac.mu`. If it's not found, it sends a request to the DNS server to translate the domain name into an IP address. The DNS server responds with the IP address associated with the domain name.
3. **ISP Role:** The Internet Service Provider (ISP) routes the request to the appropriate server using the obtained IP address. The ISP ensures that the data packets travel across the internet to reach the destination server.
4. **Server Response:** The server at `www.open.ac.mu` listens on port 80 for incoming HTTP requests. Upon receiving the request for `/index.htm`, the server processes it and sends back the requested webpage.
5. **Browser Rendering:** The browser receives the HTML content of the `index.htm` file and renders the webpage for the user to view.

Question 1a (iv)

Many computer users often confuse the terms the 'internet' and the 'World Wide Web'. Explain what is meant by the term 'World Wide Web'.

- **Internet:** The internet is a global network of interconnected computers and servers that communicate using standardized protocols. It is the infrastructure that allows for various types of data exchanges and services, such as email, file transfer, and the World Wide Web.
- **World Wide Web (WWW):** The World Wide Web is a system of interlinked hypertext documents and resources, accessible via the internet. It uses HTTP (Hypertext Transfer Protocol) to transmit data. Users can access the web using web browsers to view web pages that may contain text, images, videos, and other multimedia.

The World Wide Web is essentially a subset of the internet, consisting of websites and web

pages, while the internet encompasses a broader range of networking services and protocols.

Question 2

Question 2a

Illustrate with a diagram how the internet is structured based on the client-server architecture.

Below is a textual representation of the client-server architecture:

```
Client Devices (e.g., PC, Smartphone, Tablet)
    |
    v
Internet (Router, ISP, DNS)
    |
    v
Web Server (e.g., Apache, Nginx)
    |
    v
Database Server (e.g., MySQL, PostgreSQL)
```

Question 2b

Web pages can either be static or dynamic. What is the main difference between static and dynamic web pages.

1. Static Web Pages:

- Content is fixed and does not change unless manually updated.
- Typically written in HTML and CSS.
- Suitable for simple websites with a limited amount of content.
- Example: A company's informational page.

2. Dynamic Web Pages:

- Content can change based on user interaction or other factors.
- Often generated in real-time using server-side scripting languages

(e.g., PHP, [ASP.NET](#)) and databases.

- Suitable for complex websites requiring frequent updates or user interactions.
- Example: Social media platforms, e-commerce sites.

Question 2c

A scripting language is a programming language that supports scripts or programs written for a special run-time environment that automates the execution of tasks. The two main types of scripts are Client-side and Server-side Scripting. List two (2) differences between Client-side and Server-side Scripting.

1. Client-side Scripting:

- Executed on the user's browser.
- Example languages: JavaScript, HTML5.
- Used for creating interactive and responsive user interfaces.
- Does not require a server to execute, hence faster for certain operations.

2. Server-side Scripting:

- Executed on the web server.
- Example languages: PHP, Python, Ruby, [ASP.NET](#).
- Used for accessing databases, processing form data, and managing server-side logic.
- Requires a server to execute, which can add some latency.

Question 2d

In PHP script writing, the variables can be declared anywhere in the script. The scope of a variable is the part of the script where the variable can be referenced or used. List three (3) variable scopes in PHP.

1. Global Scope:

- Variables declared outside any function.
- Accessible throughout the entire script, but need the `global` keyword to be accessed inside functions.

2. Local Scope:

- Variables declared within a function.

- Only accessible within that function.

3. Static Scope:

- Variables declared within a function with the `static` keyword.
- Retains its value between function calls.

Question 2e

Explain the following PHP code and what is the output:

```
<?php
$a = 15;
$b = 6;

function myTest() {
    global $a;
    $b = 20;
    $b = $a + $b;
    echo "The value of the operation is ".$b;
}

myTest();
?>
```

Explanation:

- `$a` is declared as a global variable with a value of 15.
- `$b` is declared with a value of 6 but is not used in the function.
- Inside the `myTest` function, the `global` keyword is used to access the global variable `$a`.
- A new local variable `$b` is declared within the function and assigned a value of 20.
- The local `$b` is then updated to the sum of the global `$a` (15) and the local `$b` (20), resulting in `$b` being 35.
- The function outputs: "The value of the operation is 35".

Output:

The value of the operation is 35

If you have any more questions or need further assistance, feel free to ask!

Question 3

Question 3a

Explain each of the following terms in computer security:

i. Threat

- **Explanation:** A threat is any potential danger that can exploit a vulnerability to breach security and cause possible harm. It can be natural (e.g., floods), human (e.g., hacking), or environmental (e.g., power failure).

ii. Vulnerability

- **Explanation:** A vulnerability is a weakness or flaw in a system, network, or application that can be exploited by a threat to gain unauthorized access to resources. It can result from improper configuration, lack of updates, or software bugs.

iii. Risk

- **Explanation:** Risk is the potential for loss or damage when a threat exploits a vulnerability. It is typically calculated as a combination of the likelihood of the threat occurring and the potential impact it would have.

iv. Exploit

- **Explanation:** An exploit is a piece of code, software, or sequence of commands that takes advantage of a vulnerability to carry out an attack. It is used by attackers to gain unauthorized access or perform malicious actions.

v. Hacking

- **Explanation:** Hacking refers to the act of gaining unauthorized access to computer systems or networks. It involves using various techniques and tools to exploit vulnerabilities for malicious purposes such as data theft, disrupting services, or causing damage.

Question 3b

What is a Cookie?

- **Explanation:** A cookie is a small piece of data stored on a user's device by a web browser while browsing a website. It is used to remember information about the user, such as login status, preferences, or tracking identifiers, to enhance the browsing experience and enable certain functionalities.

Question 3c

Give two (2) uses of cookies that a Web Developer can apply while developing a website.

1. Session Management:

- Cookies are used to manage user sessions, such as keeping a user logged in across multiple pages of a website or remembering items in a shopping cart.

2. Personalization:

- Cookies help in personalizing the user experience by storing user preferences, themes, language settings, and other customizable elements to provide a tailored experience on subsequent visits.

Question 3d

Explain PHP sessions and provide an example scenario where sessions are commonly used in web development.

- **Explanation:** PHP sessions are a way to store information on the server for use across multiple pages. Unlike cookies, which are stored on the client side, session data is stored on the server, and a unique session ID is used to retrieve the data. Sessions are used to maintain state and track user activity across different pages

of a website.

- **Example Scenario:** In an online shopping site, PHP sessions can be used to store the user's cart contents. When a user adds items to their cart, the session keeps track of these items, allowing the user to navigate to different pages without losing their cart data.

Question 3e

Give two (2) differences between session and cookie.

1. **Storage Location:**

- **Session:** Stored on the server.
- **Cookie:** Stored on the client's browser.

2. **Security:**

- **Session:** Generally more secure since data is stored on the server and less exposed to potential client-side attacks.
 - **Cookie:** Less secure since data is stored on the client-side and can be accessed or manipulated by the user.
-

Question 4

Question 4a

We can use either GET or POST method to send the following form data to server. Give three (3) differences between GET and POST Method in PHP.

1. **Visibility of Data:**

- **GET:** Appends data to the URL, making it visible in the address bar.
- **POST:** Sends data in the request body, making it invisible in the URL.

2. **Data Size:**

- **GET:** Limited to a maximum of about 2048 characters.
- **POST:** No size limitations, allowing larger amounts of data to be sent.

3. **Security:**

- **GET:** Less secure as data is exposed in the URL. Not suitable for

sensitive information.

- **POST:** More secure as data is not exposed in the URL. Suitable for sensitive information.

Question 4b

Give four (4) reasons why PHP is a popular choice when designing dynamic websites.

1. **Ease of Use:**

- PHP is easy to learn and use, making it accessible for beginners and efficient for experienced developers.

2. **Integration:**

- PHP can be easily integrated with various databases, such as MySQL, PostgreSQL, and others, enabling dynamic content generation.

3. **Wide Support:**

- PHP is supported by most web hosting providers and has a large community, providing extensive resources and support.

4. **Flexibility:**

- PHP can be embedded directly into HTML, allowing developers to mix code with content seamlessly.

Question 4c

Write the PHP codes for the “student_reg.php” file. The PHP file should include the following:

- i. PHP code to open a connection to a database named exam.

```
<?php
// Database credentials
$servername = "localhost";
$username = "root";
$password = "pwd";
$dbname = "exam";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

ii. SQL statement used for adding the form data in the table student.

```
<?php
// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Collect form data
    $firstname = $_POST['fname'];
    $lastname = $_POST['lname'];
    $address = $_POST['address'];
    $phone = $_POST['phone'];

    // SQL query to insert data
    $sql = "INSERT INTO student (firstname, lastname, address, phone) VALUES ('$firstname'

    // Execute the query and check if it is successful
    if ($conn->query($sql) === TRUE) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }

    // Close the connection
    $conn->close();
}
?>
```

Complete `student_reg.php` File

```
<?php
// Database credentials
$servername = "localhost";
$username = "root";
$password = "pwd";
$dbname = "exam";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";

// Check if the form is submitted
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Collect form data
    $firstname = $_POST['fname'];
    $lastname = $_POST['lname'];
    $address = $_POST['address'];
    $phone = $_POST['phone'];

    // SQL query to insert data
    $sql = "INSERT INTO student (firstname, lastname, address, phone) VALUES ('$firstname'";

    // Execute the query and check if it is successful
    if ($conn->query($sql) === TRUE) {
        echo "New record created successfully";
    } else {
        echo "Error: " . $sql . "<br>" . $conn->error;
    }

    // Close the connection
    $conn->close();
}
?>
```