



SAVEETHA SCHOOL OF ENGINEERING
SIMATS, CHENNAI



Course Code /Title: CSA4399 – Internet Programming
Programme : Computer Science and Engineering

ASSIGNMENT 4 QUESTIONS

S.No	Questions	Marks	CO	BTL
1	From a developer's perspective, discuss why JDBC is essential in building database-driven applications. How to achieve JDBC CONNECTION pooling using JDBC Data Source and JNDI in Apache Tomcat Server. Provide examples of executing SQL queries using JDBC statements. Discuss the differences between Statement, Prepared Statement, and Callable Statement.	20	CO5	3
2	Describe the lifecycle phases of a JSP page. Explain the significance of each phase in the JSP execution process. Discuss the different ways to embed Java code within a JSP page with examples. Explain the advantages and disadvantages of using scriptlets, declarations, and expressions in JSP.	20	CO4	2
3	You need to develop a PHP program that generates a chessboard using HTML tables. The table should have a total width of 400px, and each cell should have a height and width of 30px. The chessboard should alternate colors between black and white for each cell to represent a typical chessboard layout. How would you write a PHP program using nested for loops to create a chessboard? The chessboard should be displayed using an HTML table with a total width of 400px, and each cell should have a height and width of 30px. Explain how you would use the nested for loops to alternate the cell colors and ensure the chessboard pattern is correctly displayed. Provide the code for this program	20	CO4	2
	You are developing a PHP application that reads content from a text file and uses regular expressions to extract specific patterns, such as email addresses and phone numbers. After extracting the data, the application should store the results in a new XML file following a defined schema for the data. Additionally, you need to compare and contrast DTD (Document Type Definition) and XML Schema for defining the XML structure. How would you create a PHP application that reads content from a text file and uses regular expressions to extract specific patterns (e.g.,	20	CO5	2



SAVEETHA SCHOOL OF ENGINEERING
SIMATS, CHENNAI



email addresses and phone numbers)? After extracting the data, store the results in a new XML file following a defined schema for the data. Additionally, compare and contrast DTD and XML Schema for defining XML structure. Describe the steps and provide the code for the application.

throws Exception {

{Payment Request};

} catch (SOAPFault e) {

// Handle SOAP faults (e.g., invalid request)

} catch (WebServiceException e) {

// Handle Connectivity or Configuration errors

}

Output:

Successful Payment:

Payment successful. Transaction ID: 1987653210

Payment Failure (e.g., Invalid Card Details):

Payment failed. Error: Invalid Credit Card Details.

SOAP fault (e.g., Invalid Request)

Payment failed due to a SOAP fault: Invalid
Request format.

Connectivity Issue (e.g., service unavailable).

Payment failed due to a Connectivity issue.

Cannot connect to the Payment gateway.

④

To integrate an external payment gate we service into your e-commerce application using a WSDL file.

① Generate client code from WSDL

wsimport -keep -s src -d bin -p com.example.

payment -Vbase http://example.com

/payment gate way? wsdl.

② Integrate Generated Code into Application.

- Include Generated code

- Configure Service End point.

③ Involve the payment Service.

- Create Service Instance

```
PaymentService Service = new PaymentService();
```

```
PaymentPortType port = Service.getPaymentPort();
```

- Involve Methods

```
PaymentResponse response = port.ProcessPayment(Pay-  
mentRequest);
```

④ Handle Response and Errors

- Check Response

```
if (response.isSuccess()) {
```

```
// Handle successful payment
```

```
} else {
```

```
// Handle payment failure
```

```
}
```

- Exception Handling

```
try {
```

```
PaymentResponse response = port.ProcessPayment();
```

```
Context initContext = new InitialContext();
```

```
DataSource ds = (DataSource) initContext.
```

```
lookup("java:comp/  
env/jdbc/My DB");
```

```
return ds.getConnection();
```

```
}
```

```
}
```

Executing SQL Queries Using JDBC Statement

① Using a statement

```
try (Connection conn = DatabaseUtil.getConnection();  
Statement stmt = conn.createStatement()) {  
String query = "SELECT * FROM users";  
ResultSet rs = stmt.executeQuery(query);  
while (rs.next()) {  
System.out.println("User ID: " + rs.getInt  
("id") + ", Name: " + rs.getString("name"));  
}
```

② Using a prepared statement.

```
try (Connection conn = DatabaseUtil.getConnection();  
PreparedStatement pstmt = conn.prepareStatement
```

```
("SELECT * FROM users WHERE id=?")) {
```

```
pstmt.setInt(1, 1);
```

```
ResultSet rs = pstmt.executeQuery();  
while (rs.next()) {
```

```
System.out.println("User ID: " + rs.getInt  
("id") + ", Name: " + rs.getString("name"));  
}
```

Output:

Page Display

Stock Market Quotes	
- Apple Inc. (AAPL)	: \$150.00
- Microsoft Corp. (MSFT)	: \$250.00
- Alphabet Inc. (GOOGL)	: \$2800.00

Confirmation Dialog

The page will refresh in 20 seconds.

Alert

Page refresh Cancelled

Page Refresh

(The page reloads, displaying updated content)

Stock Market Quotes	
- Apple Inc. (ARPL)	: \$152.00
- Microsoft Corp. (MSFT)	: \$250.00
- Alphabet Inc. (GOOGL)	: \$2850.00

③ Using a callable statement for stored procedures:

```
try (Connection conn = DatabaseUtil.getConnection(),  
     CallableStatement cstmt = conn.prepareCall("{?  
         call getuserById(?)?}") {  
    cstmt.setInt(1, 1);  
    ResultSet rs = cstmt.executeQuery();  
    while (rs.next()) {  
        System.out.println("User ID:" + rs.getInt()  
                           + ", Name:" + rs.getString("name"));  
    }  
}
```

Output:-

Statement Example Output:

User ID: 1, Name: Charith

User ID: 2, Name: Sai

Prepared Statement.

User ID: 1, Name: Charith

Callable Statement

User ID: 1, Name: Charith

② Life cycle phases of a JSP page

- ① Translation phase
- ② Compilation phase
- ③ Initialization phase
- ④ Request processing phase
- ⑤ Destruction phase.

Embedding Java Code in JSP

① Scriptlets

```
<% int sum = 5+10; %>  
<p>The sum is : <% =sum %>
```

Output:

The sum is : 15

② Declarations

```
<%! int add(int a, int b){ return a+b; } %>  
<p>The result is : <% =add(3,7) %></p>
```

Output:-

The result is : 10

③ Expressions

```
<p>Current time : <%= new java.util.Date ()%></p>
```

Output:-

Current time: Mon Sep 09 9:30:00 PDT 2024

Scriptlets:-

Advantages: Easy to use for embedding simple Java logic.

Disadvantages: Leads to messy code, difficult to maintain.

Declarations:-

Advantages: Useful for declaring reusable methods and variables across multiple requests.

Disadvantages:- Can clutter Jsp with Java code, leading to poor separation of concerns.

Expressions:-

Advantages: Simplifies outputting dynamic content directly in JSP.

Disadvantages: limited to expressions.

- ③ Generates a chess board using HTML tables - width of 400px (total) and each cell has a height and width of 30px.

PHP code:-

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Chess board </title>
    <style>
        table{
            width: 400px;
            border-collapse: collapse;
        }
        td{
            width: 30px;
            height: 30px;
        }
    </style>
</head>
<body>
    <table>
        <?php>
            // Loop for 8 rows
            for ($row=0; $row<8; $row++) {
                echo "<tr>";
                for ($col=0; $col<8; $col++) {
                    if ($row+$col)%2 == 0) {
                        echo "<td style='background-color: black;'></td>";
                    } else {
                        echo "<td style='background-color: white;'></td>";
                    }
                }
                echo "</tr>";
            }
        </?php>
    </table>
</body>
</html>
```

```

for ($col=0, $col<8; $col++) {
    if ((($row+$col) % 2 == 0)) {
        echo "<td style='background-color: white; border: 1px solid black;'></td>";
    } else {
        echo "<td style='background-color: black; border: 1px solid black;'></td>";
    }
}
echo "</tr>";
}
? >
</table>
</body>
</html>

```

Output:-

```

[ ][#][ ][#][ ][#][ ][#]
[ ][#][ ][#][ ][#][ ][#][ ]
[ ][#][ ][#][ ][#][ ][#][ ]
[ ][#][ ][#][ ][#][ ][#][ ]
[ ][#][ ][#][ ][#][ ][#][ ]
[ ][#][ ][#][ ][#][ ][#][ ]
[ ][#][ ][#][ ][#][ ][#][ ]
[ ][#][ ][#][ ][#][ ][#][ ]

```

- ④ PHP Application to Extract Data and store in XML

Steps:-

- ① Read content from a Text file
- ② Extract patterns using Regular
- ③ Create and store Results in an XML File.
- ④ Define XML schema.

PHP code:-

```
<?php  
$filename = 'input.txt';  
$content = file_get_contents($filename);  
preg_match_all('/[a-zA-Z0-9.-/.+-] +@ [a-zA-Z0-9.-]+.[a-zA-Z]{2,4}/', $content, $emails);  
preg_match_all('/\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}|\d{1,3}[.]\d{1,2}/', $content, $phones);  
$XML = new SimpleXMLElement('<data></data>');  
$emailElement = $XML->addChild('emails');  
foreach ($emails[0] as $email) {  
    $emailElement->addChild('email', $email);  
}  
$phoneElement = $XML->addChild('phones');  
foreach ($phones[0] as $phone) {  
    $phoneElement->addChild('phone', $phone);  
}  
$XML->asXML('output.xml');  
echo "Data extracted and saved to output.xml";  
?>
```

Output:-

<data>

<emails>

<email>example1@example.com </email>

<email>example 2@example.com </email>

</emails>

<phones>

<phone>+123-456-7890 </phone>

<phone>987-654-3210 </phone>

</phones>

</data>