Past paper 2015

HNDIT2411- Enterprise Architecture

QUESTION 01

I. State three different computing paradigms currently known in the software industry. [03 Marks]

[1*3 Marks]

a. *Imperative Computing*
b. *Procedural computing*
c. *Structured computing*
d. *Event Driven Computing*
e. *Object Oriented Computing*
f. *Service Oriented Computing*

II. Define "**Service Oriented Architecture**". [03 Marks]

An architectural style of building software applications that promotes loose coupling between components so that you can reuse them and work within a distributed systems architecture

III. List down the key characteristics of Service Oriented architecture [04 Marks]

[1 * 4 Marks]

- *Great interoperability*
- *Reduces cost*
- *Loosely coupled*
- *Increased reusability*
- *Ease of integration*
- *Higher quality of service*

IV. Differentiate between SOA and Web service [04 Marks]

Web services are used to build applications that can send/receive messages using SOAP over HTTP. A web service is a publicized package of functionality offered over the web.

SOA is a set of architectural concepts used for the development and integration of services. Web services can be used to implement SOA. But it is only a single method of realizing SOA based applications.

V. "Service Oriented Computing encompasses revolutionary software solutions for the highly dynamic modern business world" Do you agree with this statement. Explain your answer. [06 Marks]

QUESTION 02

I. What is Concurrency? [02 Marks]

Concurrency is the ability to run several programs or several parts of a program in parallel. If a time consuming task can be performed asynchronously or in parallel, this improve the throughput and the interactivity of the program.

II. Name two methods that you can used to implement a Thread in Java? [02 Marks]

Extending the java.lang.Thread Class

Implementing the java.lang.Runnable Interface

III. Explain what is thread scheduling? [02 Marks]

Execution of multiple threads on a single CPU in some order is called scheduling.

IV. Consider the given scenario and write sample code segments for the following six steps.

Assume that you are using a MySql database for the following operations. Database name is " Library" it has a table named "Book". Detail of Book table as follow: Book (bid: string, bname:string, aid:String, no_page: integer, price:float, isbn:integer). Assume this table already contains records.

a. Load the JDBC Driver [02 Marks]
b. Establish the Database Connection [02 Marks]
c. Create a Statement Object [02 Marks]
d. Insert one data record to book table. [02 Marks]
e. Display values of bname, no_page and price [04 Marks]

f. Close the Connection [02 Marks]

a. Load the JDBC Driver

```
Class.forName("com.mysql.jdbc.Driver").newInstance();
```

b. Establish the Database Connection

```
String url="jdbc:mysql://localhost:3306/ Library ";

Connection con = DriverManager.getConnection(url);
```

c. II. Create a Statement Object

```
Statement stmt = con.createStatement();
```

d. III. Insert one data record to book table. (02 marks) (Any valid code)

```
Stmt.executeUpdate("INSERT INTO Book (b03, Java, a04, 457, 1870.00, 4792132405964);
```

e. Display values of bname, no_page and price (Any valid code)

```
ResultSet results = stmt.executeQuery("SELECT bname, no_page, price FROM Book");

while (results.next()) {

        String b_name = results.getString("bname");

        intno_page= results.getInt("no_page");

        intbook_price= results.getInt("price");

        System.out.print(b_name + "   " +no_page+ " " + book_price);

i.  }
```

f. Close the connection.

```
stmt.close();
```

connection.close();

QUESTION 03

I. What is “Servlet”? [02 Marks]

Model Answer [2 Marks]

A *servlet* is a Java programming language class used to extend the capabilities of servers that host applications accessed via a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by Web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes.

II. What do you mean by Servlet Container? List common tasks performed by Servlet Container. [04 Marks]

Model Answer [2 Marks]

Servlets run inside a special container called the servlet container as part of the same process as the web server itself. It is an application server that provides the facilities for running Java servlets

The servlet container is responsible for initializing, invoking, and destroying each servlet instance.

Model Answer [1 * 2 Marks]

Common tasks of the servlet container:

- Creating the Request Object.
- Creating the Response Object.
- Invoking the service method of the servlet, passing the request and response objects.
- Communication Support.
- Lifecycle and Resource Management
- Multithreading Support
- JSP Support

- Miscellaneous Task: Servlet container manages the resource pool, perform memory optimizations, execute garbage collector, and provides security configurations, support for multiple applications, hot deployment and several other tasks behind the scene that makes a developer life easier.

III. Define "Deployment Descriptor"? [02 Marks]

A deployment descriptor is an Extensible Markup Language (XML) text-based file with an .xml extension that describes a component's deployment settings. A J2EE application and each of its modules has its own deployment descriptor.

Because deployment descriptor information is declarative, it can be changed without modifying the bean source code. At run time, the J2EE server reads the deployment descriptor and acts upon the component accordingly.

IV. Briefly describe the term "Session management in servlet". What are the different methods of session management in servlets? [06 Marks]

Model Answer [02 Marks]

Session is a conversional state between client and server and it can consists of multiple request and response between client and server. Since HTTP and Web Server both are stateless, the only way to maintain a session is when some unique information about the session (session id) is passed between server and client in every request and response.

Model Answer [1 * 4 Marks]

Some of the common ways of session management in servlets are:

a. User Authentication
b. HTML Hidden Field
c. Cookies
d. URL Rewriting
e. Session Management API

V. Assume you have provided following index.html file for collecting user name.

User Name submit	<form method="post" action="Hello"> User Name <input type="text" name="txtUser"> <input type="submit"value="submit"> </form>

Write the code for java Servlet which collect the request and display user name.

[06 Marks]

Model Answer

```
import java.io.*;

importjavax.servlet.*;                    01 Mark

importjavax.servlet.http.*;

public class HelloWorld extends HttpServlet              01 Mark

{

                                          02 Marks

public void doGet(HttpServletRequestreq, HttpServletResponse res) throws ServletException,

IOException

        {

                String username = request.getParameter("txtUser");

               res.setContentType("text/html");

               PrintWriter out=res.getWriter();

               out.println("<HTML>");

               out.println("<HEAD><TITLE>--------</TITLE></HEAD>");
```

```
                out.println("<BODY>");                                    02 Marks
                        out.println("<BIG>txtUser</BIG>");
                out.println("</BODY>");
                out.println("</HTML>");
        }
}
```

QUESTION 04

I. Briefly explain the advantages of using JSP over Servlets in web application development? [Marks 04]

At its most basic, JSP allows for the direct insertion of servlet code into an otherwise static HTML file
Any java syntax is valid within these scriptlet tags
Comment : give marks for any suitable answer

II. What is difference between GET and POST method in HTTP protocol? In which situation does we use doPOST() then? [04 Marks]

GET	Retrieves information identified by a request Uniform Resource Identifier (URI)
POST	Requests that the server pass the body of the request to the resource identified by the request URI for processing.

doPost has no limitations on parameter numbers while doGet has

doGet is faster than doPost.

doPost is secured than doGet.

III. Consider the following index.html file which take user name and password

```
<html>
        <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>JSP Page</title>
        </head>
        <body>
        <form method="post" action="Valiate.jsp">
                        User Name <input type="text" name="txtUser" ><br>
                        Password <input type="text" name="txtPwd" ><br>
                <input type="submit" value="submit"><br>
                </form>
        </body>
</html>
```

JSP Page

User Name

Password

submit

a. Write a JSP code to get the user name and password through request object at the server side [2 marks]

```
<%
String strUsername = request.getParameter("txtUser");
String strPassword = request.getParameter("txtPwd");
%>
```

b. Write a JSP code to check the user name and password are blank or not [02 marks]

```
<%
if (strUsername == null || strPassword == null)
        {
out.print("Blank user name or passeord ");
        }
%>
```

c. Write a JSP code to check the user name = ‘admin’ and password =’abc@123’ [2 marks]

```
        <%
If (strUsername.trim().equals("admin") && password. trim().equals("abc@123"))
{
out.println("Welcome ");
        }
else
{
out.println("Invalid username and password");
        }
%>
```

IV. Briefly describe the three types of Enterprise Java Beans. [06 Marks]

[2 * 3 Marks]

Session beans

Session beans implement business logic. A session bean instance serves one client at a time. There are three types of session beans: stateful, stateless, and singleton.

Message-Driven Beans

A message-driven bean implements loosely coupled or asynchronous business logic in which the response to a request need not be immediate. A message-driven bean receives messages from a JMS Queue or Topic, and performs business logic based on the message contents. It is an asynchronous interface between EJBs and JMS.

Entity Bean

An "**Entity Bean**" is a type of Enterprise JavaBean, a server-side Java EE component that represents persistent data maintained in a database. An **entity bean** can manage its own persistence (**Bean**managed persistence) or can delegate this function to its EJB Container (Container managed persistence).

QUESTION 05

I. What are the major the differences between XML and HTML [02 Marks]

XML was designed to store and transport data, HTML was designed to display data

II. Why do we use a DTD in XML? [02 Marks]

With a DTD, independent groups of people can agree on a standard for interchanging data.

With a DTD, you can verify that the data you receive from the outside world is valid.

III. Briefly explain the term "XML Parser". Give 2 examples for Java XML parsers. [04 Marks]

A **parser** is a program that reads a document , checks whether it is syntactically correct, and takes some actions as it processes the document.

Java Parsers

- DOM
  - Sun JAXP
  - IBM XML4J

- Apache Xerces
- Resin (Caucho)
- DXP (DataChannel)

- SAX
  - Sun JAXP
  - SAXON
- JDOM

IV. What do you mean by Valid XML Document [02 Marks]

A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a DTD

V.

a. Write XML code that describes the given scenario.

The root element is Vehicles. The VEHICLE element must contain one each of items MODEL, REGNO, COLOR, and BRAND in order. The VEHICLE element may contain an attribute called "transmission type" which may have a value of "Auto" or "Manual". The elements MODEL, REGNO, COLOR, and BRAND all contain PCDATA which is parsed character data.

Note: Vehicle may have multiple colors

[05 Marks]

Model Answer – any correct xml code eg:

```
<VEHICLES>
    <VEHICLE transmissiontype = “Auto”>
        <MODEL>Corolla</ MODEL>
        < REGNO >GA-1210</ REGNO >
        <COLOR> Blue</ COLOR>
        <BRAND>Toyota</ BRAND>
    </ VEHICLE>
    <VEHICLE transmissiontype = “Manual”>
```

```
                    <MODEL>Sunny</ MODEL>
                    < REGNO >GA-1010</ REGNO >
                    <COLOR> Silver</ COLOR>
                    <BRAND>Nissan</ BRAND>
                </ VEHICLE>
</VEHICLES>
```

b. Design the DTD file for the above XML document. [05 Marks]

ANSWER

```
<!DOCTYPE VEHICLES [

    <!ELEMENTVEHICLES (VEHICLE*)>
    <!ELEMENTVEHICLE ( MODEL, REGNO, COLOR+, BRAND)+>
    <!ATTLISTVEHICLE
    transmissiontype (auto|manual) #IMPLIED>
    <!ELEMENTMODEL (#PCDATA)>
    <!ELEMENTREGNO (#PCDATA)>
    <!ELEMENTCOLOR (#PCDATA)>
    <!ELEMENTBRAND (#PCDATA)>
]>
```

QUESTION 06

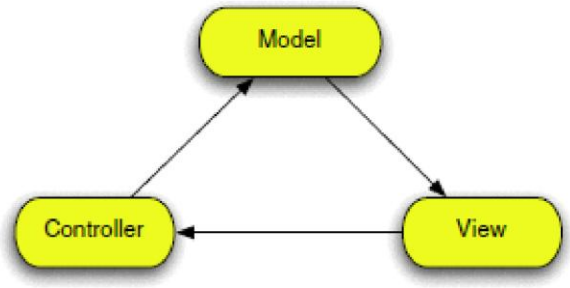
I. Briefly describe the MVC Architecture. [05 Marks]

Model Answer

Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts:

- Model - The lowest level of the pattern which is responsible for maintaining data.Contains no information about the user interface.
- View - This is responsible for displaying all or a portion of the data to the user.
- Controller - Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the Controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows.



II. Define the term "Web Application Framework" [03 Marks]

Model Answer

A *web application framework* is a piece of structural software that provides automation of common tasks of the domain as well as a built-in architectural solution that can be easily inherited by applications implemented on the framework.

III. Why do you use frameworks in web application development? [04 Marks]

any suitable answer

IV. Briefly explain any two frameworks given below. Indicate at least two benefits of each. [2*4=08 Marks]

A. Spring

B. Struts 2

C. Hibernate

a. Spring

Spring framework is an open source Java platform. Spring is the most popular application development framework for enterprise Java. Millions of developers around the world use Spring Framework to create high performing, easily testable, reusable code. Spring is lightweight when it comes to size and transparency. Spring framework targets to make J2EE development easier to use and promote good programming practice by enabling a POJO-based programming model.

Benefits

- Works on POJOs. Hence easier for dependency injection / injection of test data.
- With the Dependency Injection(DI) approach, dependencies are explicit and evident in constructor or JavaBean properties
- Enhances modularity. Provides more readable codes.
- Provides loose coupling between different modules.
- Effective in organizing the middle-tier applications.
- Flexible use of Dependency injection. Can be configured by XML based schema or annotation-based style.
- Supports declarative transaction, caching, validation and formatting.

a. Struts 2

Apache Struts 2 is an open-source web application framework for developing Java EE web applications.

Struts is a framework based on set of Java technologies like Servlet, JSP, JSTL, XML etc which provides implementation of MVC architecture. The framework also provides ready to use validation framework. The power of Struts lies in its model layer by which Struts can be integrated with other Java technologies like JDBC, EJB, Spring,

Hibernate and many more. Struts is an open source framework developed by Apache Software foundation.

Benefits

Simplified Design: Code is not tightly coupled to Struts framework or Servlet API.

Easy plug-in: Developers can use other technologies plug-in easily. It includes SiteMesh, Spring, Tiles, etc.

Simplified ActionForm: ActionForms are POJOs, we do not need to implement any interface or extend from any class.

Annotations introduced: Use of annotation results in reduction in length and complexity of code. It is also used in configuration file for simplicity.

Better tag features: It includes theme based tags and Ajax enabled tags.

Simplified Testability: Unit testing of Struts 2 Action class is very easy because it doesn't need complexHttpServletRequest and HttpServletResponse objects.

Simplified Action: Similar to ActionForms, Actions are also simple POJOs and they do not need to implement any interface or extend any class.

OGNL integration: It uses OGNL to fetch data from ValueStack and type conversion which reduces code.

Ajax support: Struts2 tags are Ajax enabled.

Multiple View options: View is not restricted to JSP. Freemarker, velocity templates can also be used as a view.

a. Hibernate

Hibernate ORM (Hibernate in short) is an object-relational mapping framework for the Java language, providing a framework for mapping an object-oriented domain model to a traditional

relational database. Hibernate not only takes care of the mapping from Java classes to database tables (and from Java data types to SQL data types), but also provides data query and retrieval facilities. Hibernate sits between traditional Java objects and database server to handle all the work in persisting those objects based on the appropriate O/R mechanisms and patterns.

Benefits

- Hibernate takes care of mapping Java classes to database tables using XML files and without writing any line of code.
- Provides simple APIs for storing and retrieving Java objects directly to and from the database.
- If there is change in Database or in any table then the only need to change XML file properties.
- Abstract away the unfamiliar SQL types and provide us to work around familiar Java Objects.
- Hibernate does not require an application server to operate.
- Manipulates Complex associations of objects of your database.
- Minimize database access with smart fetching strategies.
- Provides Simple querying of data.

a.