



**NEERAJ<sup>®</sup>**

# M.C.S.-51

## ADVANCED INTERNET TECHNOLOGIES

By: *Nirman Kaur*, M.C.A.

*Question Bank cum Chapterwise Reference Book*  
**Including Many Solved Question Papers**



**NEERAJ  
PUBLICATIONS**

*(Publishers of Educational Books)*  
( An ISO 9001 : 2008 Certified Company )

Sales Office:  
1507, 1st Floor, Nai Sarak, Delhi - 6  
Ph.: 011-23260329, 45704411,  
23244362, 23285501  
E-mail: [info@neerajignoubooks.com](mailto:info@neerajignoubooks.com)  
Website: [www.neerajignoubooks.com](http://www.neerajignoubooks.com)

MRP ` 240/-

**Published by:**

**NEERAJ PUBLICATIONS**

Sales Office : 1507, 1st Floor, Nai Sarak, Delhi-110 006

E-mail: [info@neerajignoubooks.com](mailto:info@neerajignoubooks.com)

Website: [www.neerajignoubooks.com](http://www.neerajignoubooks.com)

**Reprint Edition with Updation of Sample Question Paper Only**

Typesetting by: *Competent Computers*

Printed at: *Novelty Printer*

**Notes:**

1. For the best & upto-date study & results, please prefer the recommended textbooks/study material only.
2. This book is just a Guide Book/Reference Book published by NEERAJ PUBLICATIONS based on the suggested syllabus by a particular Board /University.
3. The information and data etc. given in this Book are from the best of the data arranged by the Author, but for the complete and upto-date information and data etc. see the Govt. of India Publications/textbooks recommended by the Board/University.
4. Publisher is not responsible for any omission or error though every care has been taken while preparing, printing, composing and proof reading of the Book. As all the Composing, Printing, Publishing and Proof Reading etc. are done by Human only and chances of Human Error could not be denied. If any reader is not satisfied, then he is requested not to buy this book.
5. In case of any dispute whatsoever the maximum anybody can claim against NEERAJ PUBLICATIONS is just for the price of the Book.
6. If anyone finds any mistake or error in this Book, he is requested to inform the Publisher, so that the same could be rectified and he would be provided the rectified Book free of cost.
7. The number of questions in NEERAJ study materials are indicative of general scope and design of the question paper.
8. Question Paper and their answers given in this Book provide you just the approximate pattern of the actual paper and is prepared based on the memory only. However, the actual Question Paper might somewhat vary in its contents, distribution of marks and their level of difficulty.
9. Any type of ONLINE Sale/Resale of "NEERAJ BOOKS/NEERAJ IGNOU BOOKS" published by "NEERAJ PUBLICATIONS" on Websites, Web Portals, Online Shopping Sites, like Amazon, Flipkart, Ebay, Snapdeal, etc. is strictly not permitted without prior written permission from NEERAJ PUBLICATIONS. Any such online sale activity by an Individual, Company, Dealer, Bookseller, Book Trader or Distributor will be termed as ILLEGAL SALE of NEERAJ IGNOU BOOKS/NEERAJ BOOKS and will invite legal action against the offenders.
10. Subject to Delhi Jurisdiction only.

**© Reserved with the Publishers only.**

**Spl. Note:** *This book or part thereof cannot be translated or reproduced in any form (except for review or criticism) without the written permission of the publishers.*

## **How to get Books by Post (V.P.P.)?**

If you want to Buy NEERAJ IGNOU BOOKS by Post (V.P.P.), then please order your complete requirement at our Website [www.neerajignoubooks.com](http://www.neerajignoubooks.com). You may also avail the 'Special Discount Offers' prevailing at that Particular Time (Time of Your Order).

To have a look at the Details of the Course, Name of the Books, Printed Price & the Cover Pages (Titles) of our NEERAJ IGNOU BOOKS You may Visit/Surf our website [www.neerajignoubooks.com](http://www.neerajignoubooks.com).

No Need To Pay In Advance, the Books Shall be Sent to you Through V.P.P. Post Parcel. All The Payment including the Price of the Books & the Postal Charges etc. are to be Paid to the Postman or to your Post Office at the time when You take the Delivery of the Books & they shall Pass the Value of the Goods to us by Charging some extra M.O. Charges.

We usually dispatch the books nearly within 4-5 days after we receive your order and it takes Nearly 5 days in the postal service to reach your Destination (In total it take atleast 10 days).



# **NEERAJ PUBLICATIONS**

(Publishers of Educational Books)

(An ISO 9001 : 2008 Certified Company)

**1507, 1st Floor, NAI SARAK, DELHI - 110006**

**Ph. 011-23260329, 45704411, 23244362, 23285501**

E-mail: [info@neerajignoubooks.com](mailto:info@neerajignoubooks.com) Website: [www.neerajignoubooks.com](http://www.neerajignoubooks.com)

# CONTENTS

## ADVANCED INTERNET TECHNOLOGIES

### *Question Bank – (Previous Year Solved Question Papers)*

<i>Question Paper—June, 2019 (Solved)</i>	1-7
<i>Question Paper—December, 2018 (Solved)</i>	1-6
<i>Question Paper—June, 2018 (Solved)</i>	1-5
<i>Question Paper—December, 2017 (Solved)</i>	1-5
<i>Question Paper—June, 2017 (Solved)</i>	1-3
<i>Question Paper—December, 2016 (Solved)</i>	1-5
<i>Question Paper—June, 2016 (Solved)</i>	1-5
<i>Question Paper—December, 2015 (Solved)</i>	1-10
<i>Question Paper—June, 2015 (Solved)</i>	1-8
<i>Question Paper—June, 2014 (Solved)</i>	1-5

<i>S.No.</i>	<i>Chapterwise Reference Book</i>	<i>Page</i>
<b>Servlets and JSP Programming</b>		
1.	Servlet Programming	1
2.	Java Database Connectivity	20
3.	Java Server Pages–I	41
4.	Java Server Pages–II	58

<i>S.No.</i>	<i>Chapter</i>	<i>Page</i>
<b>EJB and XML</b>		
5.	Introduction to Java Beans	74
6.	Enterprise Java Beans: Architecture	90
7.	EJB: Deploying Enterprise Java Beans	101
8.	XML: Extensible Markup Language	117
<b>Web Security and Case Study</b>		
9.	Web Security Concepts	127
10.	Security Implementation	133
11.	Case Study	148
		■ ■

**Sample Preview  
of the  
Solved  
Sample Question  
Papers**

*Published by:*



**NEERAJ  
PUBLICATIONS**

[www.neerajbooks.com](http://www.neerajbooks.com)

# QUESTION PAPER

(June – 2019)

(Solved)

## ADVANCED INTERNET TECHNOLOGIES

Time: 3 Hours ]

[ Maximum Marks: 100

Note: Question no. 1 is compulsory. Attempt any three questions from the rest.

**Q.1. (a) What is JDBC driver? Explain different methods to lead the drivers in JDBC.**

**Ans.** Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access any kind of tabular data, especially relational database. It is part of Java Standard Edition platform, from Oracle Corporation. It acts as a middle layer interface between java applications and database.

The JDBC classes are contained in the Java Package java.sql and javax.sql.

JDBC helps you to write Java applications that manage these three programming activities:

- Connect to a data source, like a database.
- Send queries and update statements to the database
- Retrieve and process the results received from the database in answer to your query

**Also Ref.:** See Chapter-1, Page No. 21, 'Types of JDBC Drivers'.

**(b) Write a program using servlet and JDBC for developing an online submission of request form to get details of a newly launched smartphone from XYZ Company Ltd. You need to create a database comprising of the following fields:**

- Name
- Email\_id
- Mobile\_No.
- Gender
- Address

**Ans. Register.java**

This servlet class receives all the data entered by user and stores it into the database. Here, we are performing the database logic. But you may separate it, which will be better for the web application.

```
import java.io.*;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;
```

```
public class Register extends HttpServlet {
    public void doPost(HttpServletRequest request,
        HttpServletResponse response)
```

```
    throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
```

```
        String name = request. get Parameter
        ("userName");
```

```
        String email = request. get Parameter
        ("userEmail");
```

```
        String mobile = request. get Parameter
        ("userMobile");
```

```
        String gender = request. get Parameter
        ("userGender");
```

```
        String address = request.get Parameter
        ("userAddress");
```

```
        try{
            Class.forName("oracle.jdbc.driver.Oracle
            Driver");
```

```
            Connection con = DriverManager. get
            Connection ("jdbc:oracle:thin: @localhost: 1521:
            xe", "system", "oracle");
```

```
PreparedStatement ps=con.prepareStatement(
“insert into registeruser values(?,?,?,?)”);
ps.setString(1,name);
ps.setString(2,email);
ps.setString(3,mobile);
ps.setString(4,gender);
ps.setString(5,address);
int i=ps.executeUpdate();
if(i>0)
out.print(“You are successfully registered...”);
} catch (Exception e2) {System.out.println(e2);}
out.close();
}
}
```

(c) Briefly explain any three system recovery procedures.

Ref.: See Chapter-10, Page No. 143, Q. No. 3.

(d) Explain any two types of attributes that can be declared in an XML DTD.

Ans. When declaring attributes, you can specify how the processor should handle the data that appears in the value. We can categorize attribute types in three main categories “

- String type
- Tokenized types
- Enumerated types

Following table provides a summary of the different attribute types:

Sr. No.	Type and Description
1.	<b>CDATA</b> CDATA is character data (text and not markup). It is a <i>String Attribute Type</i> .
2.	<b>ID</b> It is a unique identifier of the attribute. It should not appear more than once. It is a <i>Tokenized Attribute Type</i> .
3.	<b>IDREF</b> It is used to reference an ID of another element. It is used to establish connections between elements. It is a <i>Tokenized Attribute Type</i> .
4.	<b>IDREFS</b> It is used to reference multiple ID's. It is a <i>Tokenized Attribute Type</i> .

Sr. No.	Type and Description
5.	<b>ENTITY</b> It represents an external entity in the document. It is a <i>Tokenized Attribute Type</i> .
6.	<b>ENTITIES</b> It represents a list of external entities in the document. It is a <i>Tokenized Attribute Type</i> .
7.	<b>TOKEN</b> It is similar to CDATA and the attribute value consists of a valid XML name. It is a <i>Tokenized Attribute Type</i> .
8.	<b>TOKENS</b> It is similar to CDATA and the attribute value consists a list of valid XML name. It is a <i>Tokenized Attribute Type</i> .
9.	<b>NOTATION</b> An element will be referenced to a notation declared in the DTD document. It is an <i>Enumerated Attribute Type</i> .
10.	<b>ENUMERATION</b> It allows defining a specific list of values where one of the values must match. It is an <i>Enumerated Attribute Type</i> .

(e) What is Secure Socket Layer (SSL)? Explain SSL working in detail.

Ans. Ref.: See Chapter-10, Page No. 135, ‘SSL Authentication’ and Page No. 146, Q. No. 10.

Q.2. (a) What is servlet collaboration? Explain how servlet collaboration takes place through a shared object.

Ans. Ref.: See Chapter-1, Page No. 9, ‘Servlet Collaboration’, ‘Collaboration Through System Property’ and Page No. 12, Q. No. 9.

(b) Write a JSP program to display the current date.

Ans. We are fetching the current date and time using date object

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ page import="java.util.*" %>
```

# Sample Preview of The Chapter

*Published by:*



**NEERAJ  
PUBLICATIONS**

[www.neerajbooks.com](http://www.neerajbooks.com)



# ADVANCED INTERNET TECHNOLOGIES

**SERVLETS AND JSP PROGRAMMING**

**Servlet Programming**



## INTRODUCTION

Initially, the web sites were the collection of web pages that linked together by Hyper Text Markup Language (HTML). In addition to that, now-a-days various advance features have been added in web server technologies such as multimedia, e-commerce applications, online banking etc. These features can be created by using technologies like Java Server Pages (JSP) and Active Server Pages (ASP). Servlets are the Java platform technology that can be run on any platform or any server. Servlets enhance the functionality of web applications. They are dynamically loaded at runtime when needed. When a client sends a request the web server/container initiates the required servlet. The servlet processes the client request and sends the response back to the server. The servlets are not bind to a specific client server protocol but they are commonly used with HTTP (Hypertext Transfer Protocol). HTTP is a request-response oriented protocol. It consists of a request method, a URI (Uniform Resource Identifier), header fields and a body. An HTTP response contains a result code and header fields and a body. The service method of Http Servlet dispatches a request to different Java method for different request methods. It recognizes the standard HTTP

methods such as GET, HEAD, PUT, TRACE, OPTIONS, POST etc. Servlets use classes in Java packages `javax.servlet` and `javax.servlet.http`. Servlets which provides an advance server side extension and follow the standard framework.

## CHAPTER AT A GLANCE

### HOW TO INSTALL SERVLET ENGINE/WEB SERVER

Java Servlet is used for writing server side programming language; hence it is required to execute it in a Java Virtual Machine by using a service called Servlet Engine. A space servlet engine is software that increases the scope of web server software such as IIS (Internet Information Server, Apache etc.) and enables the web server software to execute servlets. Generally, this Servlet engine is contained in Servlet Engine or it could be added as a module. Few web servers have a built in Servlet engine like Sun Java web server, Gefion Software's Life web server etc. But other web servers like Netscape Enterprise Server, IIS and Apache group require a Servlet Engine add on module.

Servlets must run on web servers with built in web containers such as the Planet web server or on a standalone

## 2 / NEERAJ : ADVANCED INTERNET TECHNOLOGIES

Servlet container like Tomcat. Here, we will be using Apache Tomcat 4.0. It is an open source software implementation of the Java Servlet and Java Server Pages technologies.

To configure Tomcat as a stand alone server we need to download Tomcat 4.0 and JDK 1.3 (Java Development Kit) standard edition. In order to install and run this container we need to do the following:

- (a) Download and install it from the location <http://jakarta.apache.org/builds/jakarta-tomcat-4.0/release/v4.0.1/bin/>.
- (b) Download jakarta-tomcat-4.0.1.zip file to C:\temp. Now, to download the Sun JDK, go to <http://java.sun.com/j2se/1.3/download.html>, choose the Windows and click download. Scroll down and accept the license agreement. Then click the FTP download and put the download file j2sdk-1\_3\_1\_03-win.exe file to C:\temp\ directory.

Now, we need to create two environment variables, CATALINA\_HOME and JAVA\_HOME.

- (a) Right click on My Computer.
- (b) Select Properties
- (c) Select the Advanced tab.
- (d) Select the Environment Variables button.
- (e) In the System Variable group, select New.
- (f) In the Variable Name field, type JAVA\_HOME.
- (g) In the Variable Value field, type the name of the path to your JVM. (as C:\Program Files\Java)
- (h) Select OK.
- (i) In the System Variable group, select New.
- (j) In the Variable Name field, type CATALINA\_HOME.
- (k) In the Variable value field, type the name of the path to Tomcat (as C:\ProgramFiles\ Apache Tomcat 4.0)
- (l) Select OK.
- (m) Select OK.
- (n) Select OK.

Once the installation is done, the next question arises how to start and stop the server. For which follow these steps:

- To start the server, start a command prompt window and run the command:  
C:\jakarta-tomcat\bin\startup.bat
- To stop the server, run the command:  
C:\jakarta-tomcat\bin\shutdown.bat

### YOUR FIRST JAVA SERVLET

Once the basic container is installed and configured, next step is to write the servlet programming. The servlet programming interface i.e. Java Servlet API has two packages: javax.servlet and javax.servlet.http. These

packages contain many classes which are being used by the servlet. The Java Servlet API is a part of J2EE platform and has following advantages over other scripts called CGI.

- Servlets add dynamic behaviour to servers. It makes very easy to write complex services for web based applications.
- Because servlets are written in java, it has full access to Java's advance features such as database connectivity, network awareness, object orientation etc.
- Servlets can be interfaced with different databases like Oracle and SQL server.

Now, lets' create a new file and save it as firstservlet.java

```
1. import java.io.*;
2. import java.servlet.*;
3. import java.servlet.http.*;
4. public class firstservlet extends
   HttpServlet
5. {
6.     public void doGet(Http Servlet
   Request req, HttpServletResponse
   res)
7.         throws ServletException,
   IOException
8.     {
9.         res.setContentType("text/
   html") ;
10.    PrintWriter out = res.get Writer()
   ;
11.    out.println("<HTML>
   <HEAD>");
12.    out.println("<<TITLE>First   Java
   Servlet<</TITLE></ HEAD>");
13.    out.println("<BODY>Hello,
   Client</BODY></HTML>");
14.    out.close();
15. }
16. }
```

**Lines from 1-3:** are called packages which contain the classes.

**Line 4:** firstservlet is a class. This is a standard base class for HttpServlets.

**Lines from 6-7:** The HttpServletResponse object used to set the content type of the response that we have to send. All the response headers need to be set before ServletOutputStream.

**Line 10:** writes text to the response message.

The PrintWriter gets closed when out.close() is encountered.

**Compiling and Running the Servlet:** To compile the above written servlet, type the following command at the Dos prompt: javac firstservlet.java

If the program gets compiled successfully, a `firstservlet.class` file will be created. Now, to run the servlet, type the following address in the browser:

```
http://localhost:8080/star/servlet/  
com.stardeveloper.servlets.firstservlet
```

As we have seen certain methods are invoked by server in order to handling the requests. Each time server sends a request to a servlet. The service method() accepts two parameters - request object and a response object. The request object used to send the request whereas the response object used to return the response.

**SERVLET LIFE CYCLE:** A Java Servlet has a life cycle, which defines servlet processing i.e. how the servlet receives and responds to requests. The servlet life cycle is defined by the `javax.servlet.Servlet` interface. It consists of following steps:

1. Initially, the servlet class is loaded by the container.
2. The Container implements the `init()` method only once during the life of the servlet.
3. After initialization, the service method() is invoked. This method invokes two parameters `HttpServletRequest` and `HttpServletResponse`. Once the request is determined, the same is send to the appropriate method.

4. At last, the `destroy()` method is called. This method takes the servlet out of service and invoked only once during the lifecycle of a servlet.

Three main stages in the life of Java Servlet are:

1. **Servlet Initialization:** At the outset, the servlet's constructor is called with the `init()` method. The purpose of this method is to allow a servlet to perform the initialization before the `HttpRequest` being invoked. This method is called only once on any given instance.
2. **Servlet Execution:** Once the `init()` method is called, the next step is to start executing application logic in a servlet i.e. all the requests received by the servlet container are forwarded to servlet's appropriate service() method. Further, `HttpServlet` class breaks this method into various methods as `doGet()`, `doPost()`, `doDelete()`, `doPut()`, `doOptions()` and `doTrace()` depending on the type of the request it receives.
3. **Servlet Destruction:** Once the servlet finishes execution of service and there is no request pending, servlet's `destroy()` method is called. This method is used to destroy a servlet instance, which is out of service. Few activities which can be implemented in this method are performing de-allocation of those resources which were

being allocated during initialization, closing database connection etc.

Lets' understand these stages with the help of the following program:

```
import java.io.*;  
import java.servlet.*;  
import java.servlet.http.*;  
import java.io.PrintWriter ;  
import java.io.IOException ;  
public class stagesdemo extends HttpServlet  
{  
    public void init() throws  
ServletException  
    {  
        System.out.println("Invoking  
init() method");  
    }  
    public void destroy()  
    {  
        System.out.println("Invoking  
destroy() method");  
    }  
    /*Process HTTP doGet() request*/  
    public void doGet(HttpServletRequest  
Request request, HttpServletResponse  
Response response)  
        throws IOException,  
        ServletException  
    {  
        response.setContentType("text/  
html") ;  
        PrintWriter out = response.  
getWriter() ;  
        out.println("<HTML><HEAD>");  
        out.println("<TITLE>Servlet  
Life Cycle Demo</TITLE></  
HEAD>");  
        out.println("<BODY><p>Calling  
get() method </p></BODY></  
HTML>");  
        out.close();  
    }  
    /*Process HTTP doPost() request*/  
    public void doPost(HttpServletRequest  
Request request, HttpServletResponse  
Response response)  
        throws IOException,  
        ServletException  
    {  
        response.setContentType("text/  
html") ;  
        PrintWriter out = response.  
getWriter() ;  
        out.println("<HTML><HEAD>");  
    }  
}
```

4 / NEERAJ : ADVANCED INTERNET TECHNOLOGIES

```
        out.println("<TITLE>Servlet  
        Life Cycle Demo</TITLE></  
HEAD>");  
        out.println("<BODY><p>Calling  
        Post() method </p></BODY></  
HTML>");  
        out.close();  
    }  
}
```

**HTTPSERVLET REQUEST INTERFACE**

HttpServletRequest and HttpServletResponse are two main interfaces in the Servlet. The Servlet container creates an HttpServletRequest object and passes it as an argument to the servlet's service methods such as doGet() and doPost(). This interface provides access to an input stream and enables it to read data from the client. Various methods are provided which enables the servlet to process the clients' request. Few of them are listed below:

- 1. get\_cookies  
public Cookie[] get\_cookies

It returns an array that contains all the cookie objects present in this request. If no cookies are present in the request, this method returns Null.

- 2. get\_header  
public java.lang.String  
get\_header(java.lang.String name)

It returns the value of the specified request header as a string. A parameter name is passed as a string specify the header name. It returns Null if the request does not have a header of that name.

- 3. get\_method  
public java.lang.String  
get\_method()

It returns the name of the HTTP method with which this request was made e.g. Get, Post or Put.

- 4. get\_query\_string  
public java.lang.String  
get\_query\_string()

It returns a string contained in the request URL. If no query string is present it returns NULL.

- 5. get\_servlet\_path  
public java.lang.string.  
get\_servlet\_path()

It returns a string that contains the name or path of the servlet as specified in the request URL.

- 6. get\_session  
public HttpSession  
get\_session(boolean create)

It returns the current HttpSession associated with this request. If no argument is passed in this method, a new session will be created. In case it is called with a Boolean

argument then, the new session will be created only if the argument is true.

- 7. get\_parameter(String name)  
public String  
get\_parameter(String name)

It returns the value associated with a parameter, which is sent to the servlet as a part of GET and POST request.

**HTTPSERVLETRESPONSE INTERFACE**

HttpServletResponse interface provides functionality in sending a response. It provides access to an output stream and allows the servlet to send data to client. The getWriter() method obtains a reference to PrintWriter object, which is used to send the text to the client. Following methods are used to formulate response to client:

- 1. add\_cookie  
public void  
add\_cookie(Cookie cookie);

It adds the specified cookie to the response and called be multiple times to set more than one cookie.

- 2. send\_error  
public void  
send\_error(int sc)  
throws java.io.  
IOException;  
public void  
send\_error(int sc,  
java.lang.String msg)  
throws java.io.  
IOException;

It is used to send an error message to the client using the specified status code and clearing the buffer. If the response has been committed, it throws an IllegalStateException. Once this method is used, the response should be considered to be committed. The parameter sc represents the error status code. If an input or output exception occurs then throws java.io.IOException called and if the response was committed then IllegalStateException called.

- 3. send\_date\_header  
public void  
send\_date\_header(java.  
lang.String name, long date)

It is used to set a response header with the specified given name and date value. If the header is already set, it overwrites previous value with new value.

- 4. set\_header  
public void  
set\_header(java.  
lang.String name, java.lang.  
String value)

It sets a response header with the specified name and string value. It is used to check the presence of a header.

- 5. add\_header  
public void  
add\_header(java.lang.  
String name, java.lang.String  
value)