USN	1	P	E	1	7	M	C	A		

PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

INTERNAL ASSESSMENT TEST – 1

Date : 18/02/19 **Max Marks**: 40
Subject & Code : Advanced Java(17MCA41) **Time** : 8:30 am – 10:00 am
Name of faculty : Ms.Asha Joseph

Note: ***Note: Answer any FIVE full questions, choosing one full question from each Part***

	PART-A	
1 2	What are Servlets? Explain the basic Servlet Structure. OR Write a short note on the Servlet life cycle.	8
	PART-B	
3 4	Compare the different methods for reading form data from Servlets with example. OR What are cookies? Describe the methods used for cookie handling.	8
	PART –C	
5 6	Elaborate on the various HTTP request headers. OR Explain the process of session tracking using the HttpSession object.	8
	PART-D	
7 8	Explain the different tags used in JSP. Illustrate with an example. OR What are page directives? Explain the various attributes of page directives.	8
	PART- E	
9 10	Discuss in detail about Java Bean and the different action tags associated with it. OR a) Differentiate between jsp:forward and jsp:include action tags with an example. b) Explain the purpose of jsp:plugin tag with an example.	8

USN	1	P	E	1	7	M	C	A		

PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

PART - A

1. What are Servlets? Explain the basic Servlet Structure.

- A Servlet is an Java application programming interface (API) running on the server,
- It intercepts requests made by the client and generates/sends a response.
- When an HTTP request arrives at the web server which should be handled by a servlet, the web server forwards the request to the servlet container.
- The servlet container then forwards the request to the servlet that is to handle the request.

SERVLET STRUCTURE

- A servlet class should extend HttpServlet and override **doGet** or **doPost**, depending on whether the data is being sent by GET or by POST.
- Both doGet and doPost take two arguments: an **HttpServletRequest** and **HttpServletResponse.**
- The HttpServletRequest has methods by which you can find out about incoming information such as form data, HTTP request headers, and the client's hostname.
- The HttpServletResponse lets you specify outgoing information such as HTTP status codes (200, 404, etc.) and response headers (Content-Type, Set-Cookie, etc.).
- It lets you obtain a PrintWriter with the document content is sent back to the client.
- Classes to import - java.io (for PrintWriter, etc.), javax.servlet (forHttpServlet), and javax.servlet.http (for HttpServletRequest and HttpServletResponse).
- doGet and doPost throw two exceptions (ServletException and IOException) , it is required to include them in the method declaration.

2. Write a short note on the Servlet life cycle.

- The web container maintains the life cycle of a servlet instance.
- Servlet class is loaded.
- Servlet instance is created.
- init() method is invoked.
- service() method is invoked.
- destroy() method is invoked.
- 1**) Servlet class is loaded**
- The classloader is responsible to load the servlet class. The servlet class is loaded when the first request for the servlet is received by the web container.
- **2) Servlet instance is created**
- The web container creates the instance of a servlet after loading the servlet class. The servlet instance is created only once in the servlet life cycle.
- 3) **init() method is invoked**
- The web container calls the init method only once after creating the servlet instance. The init method is used to initialize the servlet.
- **public void** init()
- 4) **service() method is invoked**

- The web container calls the service() method each time when request for the servlet is received.
- If servlet is not initialized, it follows the first three steps as described above then calls the service method.
- The service method checks the HTTP request type(GET,POST,DELETE,PUT) and calls doGet, doPost, doDelete and doPut

- **public void** service(ServletRequest request, ServletResponse response) **throws** ServletEx ception, IOException
- 5) **destroy() method is invoked**
- The web container calls the destroy method before removing the servlet instance from the service. It gives to clean up any resource like memory, thread etc.
- **public void** destroy()

PART - B

1. Compare the different methods for reading form data from Servlets with example.

- Reading Single Values: getParameter
- Reading Multiple Values: getParameterValues
- Reading all Parameters: getParameterNames
- **getParameter()**
  To read a request (form) parameter, call the getParameter method of HttpServletRequest, supplying the parameter name as an argument.
  Example:
  request.getParameter("first_name") + "\n" + " Last Name: " + request.getParameter("last_name") + "\n" + "</body>" + "</html>");
  getParameterValues()
- If the same parameter name appears in the form more than once, getParameterValues is called.
  It returns an array of strings instead of getParameter (which returns a single string).
  String[] values=request.getParameterValues("t1");

```
    out.println("Selected Values...");
      for(int i=0;i<values.length;i++)
     {
        out.println("<li>"+values[i]+"</li>");
     }       out.close();
```

  getParameterNames()

- Useful to get a full list of parameter names.
- getParameterNames get the list of parameters in the form of an Enumeration.
- Each entry is cast to a String and used in a getParameter or getParameterValues call.
- hasMoreElements and nextElement methods
- Enumeration en=req.getParameterNames(); while(en.hasMoreElements()) {
- Object objOri=en.nextElement();
- String param=(String)objOri;
- String value=req.getParameter(param);
- pw.println("Parameter Name is '"+param+"' and Parameter Value is '"+value+"'"); } pw.close();

2. What are cookies? Describe the methods used for cookie handling.

- Cookies are text files stored on the client computer and they are kept for the purpose of information.
- There are three steps involved in identifying returning users −
- Server script sends a set of cookies to the browser. For example name, age, or identification number etc.
- Browser stores this information on local machine for future use.
- When next time browser sends any request to web server then it sends those cookies information to the server and server uses that information to identify the user.

Methods used for cookie handling:

- public void setComment(String purpose): This is basically used for describing the purpose of the cookie.
- public String getComment(): Returns the comment describing the purpose of this cookie, or null if the cookie has no comment.
- public void setMaxAge(int expiry): Sets the maximum age of the cookie in seconds.
- public int getMaxAge(): Gets the maximum age in seconds of this Cookie. By default, -1 , which indicates that the cookie will persist until browser shutdown.
- public String getName(): Returns the name of the cookie. The name cannot be changed after creation.
- public void setValue(String newValue): Assigns a new value to this Cookie.
- public String getValue(): Gets the current value of this Cookie.

PART - C

1. Elaborate on the various HTTP request headers.

- Accept The MIME types the browser prefers.

- Accept-Charset The character set the browser expects.
- Accept-Encoding The types of data encodings (such as gzip) the browser knows how to decode. Servlets can explicitly check for gzip support and return gzipped HTML pages to browsers that support them, setting the Content-Encoding response header to indicate that they are gzipped.
- Accept-Language The language the browser is expecting, in case the server has versions in more than one language.
- Authorization Authorization info, usually in response to a WWW-Authenticate header from the server.
- Connection - If a servlet gets a Keep-Alive value here, or gets a request line indicating HTTP 1.1 (where persistent connections are the default), it may be able to take advantage of persistent connections, saving significant time for Web pages that include several small pieces (images or applet classes).

2. Explain the process of session tracking using the HttpSession object.

- Servlet provides an HttpSession Interface which provides a way to identify a user across more than one page request.

- The session persists for a specified time period, across more than one connection or page request from the user.

- The HttpServletRequest interface provides methods to get the object of HttpSession:

- **public HttpSession getSession():**Returns the current session associated with this request, or if the request does not have a session, creates one.
- **public Object getAttribute(String name)**
- This method returns the object bound with the specified name in this session, or null if no object is bound under the name.
- **public Enumeration getAttributeNames()**
- This method returns an Enumeration of String objects containing the names of all the objects bound to this session.
- **public long getCreationTime()**
- This method returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.
- **public String getId()**
- This method returns a string containing the unique identifier assigned to this session.
- **public long getLastAccessedTime()**
- This method returns the last accessed time of the session, in the format of milliseconds.

PESIT Bangalore South Campus

Hosur road, 1km before Electronic City, Bengaluru -100

Department of MCA

PART - D

1. Explain the different tags used in JSP. Illustrate with an example.

The Scriptlet

- A scriptlet can contain any number of JAVA language statements, variable or method declarations, or expressions that are valid in the page scripting language.
- <% code fragment %>
- <% out.println("Your IP address is " + request.getRemoteAddr()); %>

JSP expression tag

- The code placed within is written to the output stream of the response.
- out.print() is not required to write data.
- It is mainly used to print the values of variable or method.
- **<%= statement %>**
- <%= "welcome to jsp" %>
  <%= "Welcome "+request.getParameter("uname") %>

JSP declaration tag

- It is used to declare fields and methods.
- The code written inside the jsp declaration tag is placed outside the service() method of auto generated servlet.
- So it doesn't get memory at each request.
- <%! field or method declaration %>
- <%! int data=50; %>
- <%= "Value of the variable is:"+data %>

2. What are page directives? Explain the various attributes of page directives.

- The **page** directive is used to provide instructions to the container that pertain to the current JSP page.
- <%@page attribute = "value" %>
- Eg
- <%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
- <%@ page import="java.util.*"%>

The attributes used with page directives are:

- **Buffer -** Specifies a buffering model for the output stream.
- **autoFlush -** Controls the behavior of the servlet output buffer.
- **contentType -** Defines the character encoding scheme.
- **errorPage -** Defines the URL of another JSP that reports on Java unchecked runtime exceptions.

- **isErrorPage -** Indicates if this JSP page is a URL specified by another JSP page's errorPage attribute.
- **Import -** Specifies a list of packages or classes for use in the JSP as the Java import statement does for Java classes.
- **Info -** Defines a string that can be accessed with the servlet's **getServletInfo()** method.
- **Language -** Defines the programming language used in the JSP page.
- **Session -** Specifies whether or not the JSP page participates in HTTP sessions.
- **isELIgnored -** Specifies whether or not the EL expression within the JSP page will be ignored.
- **isScriptingEnabled -** Determines if the scripting elements are allowed for use.

PART - E

1.Discuss in detail about Java Bean and the different action tags associated with it.

- JavaBean class in Java
- JavaBeans are classes that encapsulate many objects into a single object (the bean). It is a java class that should follow following conventions:
- Must implement Serializable.
- It should have a public no-arg constructor.
- All properties in java bean must be private.
- It should provide methods to set and get the values of the properties called getter and setter methods.

The action tags used with Java Bean are:

- **jsp:useBean** To get the java bean object from given scope or to create a new object of java bean.
- **jsp:getProperty** To get the property of a java bean, used with jsp:useBean action.
- **jsp:setPropert**y To set the property of a java bean object, used with jsp:useBean action.

2 a) Differentiate between jsp:forward and jsp:include action tags with an example.

- Actions use constructs in XML syntax to control the behavior of the servlet engine.
- Actions can be used to dynamically insert a file, reuse JavaBeans components, forward the user to another page, or generate HTML for the Java plugin.
- **jsp:include**
  To include a resource at runtime, can be HTML, JSP or any other file
- **jsp:forward** - To forward the request to another resource.
  Example
  date.jsp
  Today's date: <%= (new java.util.Date()).toLocaleString()%>
  main.jsp

```
<center> <h2>The include action Example</h2>
<jsp:include page = "date.jsp" flush = "true" /> </center>
<jsp:forward page=”date.jsp”/>
```

b) Explain the purpose of jsp:plugin tag with an example.

jsp:plugin - To generate the browser-specific code that makes an OBJECT or EMBED tag for the Java plugin.

<jsp:plugin align="middle" height="500" width="500" type="applet" code="MouseDrag.class" name="clock" codebase="."/>