Servlet Code-Based Questions (10)

1. What will be the output of this servlet?

```
@WebServlet("/TestServlet")
public class TestServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.getWriter().println("Response A");
        response.sendRedirect("newPage.jsp");
        response.getWriter().println("Response B");
    }
}
```

What will happen when accessing `/TestServlet`?
a) "Response A" and "Response B" will be displayed
b) Only "Response A" will be displayed
c) Compilation error
d) Runtime Exception

---

2. What does this Servlet code output?

```
HttpSession session = request.getSession();
session.setAttribute("user", "John");
session.invalidate();
System.out.println(session.getAttribute("user"));
```

a) Prints `"John"`
b) Prints `null`
c) Throws `NullPointerException`
d) Compilation error

---

3. What will be the session timeout for this servlet?

```
session.setMaxInactiveInterval(600);
```

a) 600 seconds
b) 10 minutes
c) Both (a) and (b)
d) Depends on web.xml configuration

---

4. What happens if `doPost()` is called without overriding it?

```
@WebServlet("/test")
public class MyServlet extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res)
            throws ServletException, IOException {
        res.getWriter().write("Hello GET");
    }
}
```

a) The request will be forwarded to `doGet()`
b) A 405 HTTP error occurs
c) Compilation error
d) Servlet container handles it automatically

---

5. What is the output of this servlet code?

```
public void doGet(HttpServletRequest req, HttpServletResponse res) throws
IOException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    out.println("First Line");
    out.close();
    out.println("Second Line");
}
```

a) "First Line" and "Second Line"
b) "First Line" only
c) Runtime error (IllegalStateException)
d) Compilation error

---

6. What happens if we call `request.getRequestDispatcher("new.jsp").forward(request, response);` after `response.getWriter().println("Hello");`?

a) "Hello" is displayed and then forwarded
b) Only "Hello" is displayed
c) Throws `IllegalStateException`
d) Compilation error

---

7. What does this servlet code do?

```
Cookie cookie = new Cookie("user", "John");
cookie.setMaxAge(0);
response.addCookie(cookie);
```

a) Deletes the cookie
b) Stores the cookie for 0 seconds
c) Throws `IllegalArgumentException`
d) Cookie is ignored

---

8. Which statement is correct for this filter code?

```
@WebFilter("/*")
public class MyFilter implements Filter {
    public void doFilter(ServletRequest request, ServletResponse response,
FilterChain chain)
            throws IOException, ServletException {
        System.out.println("Before Servlet");
        chain.doFilter(request, response);
        System.out.println("After Servlet");
    }
}
```

a) "Before Servlet" is printed, then the request is processed, and then "After Servlet" is printed
b) "Before Servlet" is printed, and the request stops
c) The response is modified before passing to the next filter
d) The filter runs only for GET requests

---

9. What is the issue with this code?

```
@WebServlet("/test")
public class TestServlet extends HttpServlet {
    protected void doGet(HttpServletRequest req, HttpServletResponse res)
            throws IOException {
        req.getRequestDispatcher("/next.jsp").forward(req, res);
        res.getWriter().println("After forwarding");
    }
}
```

a) "After forwarding" is displayed after the forwarded page
b) Compilation error
c) Runtime error (`IllegalStateException`)
d) Code works fine

---

10. What happens when `HttpServletResponse`'s `setStatus(301)` is used?

a) Sends an HTTP 301 response (Moved Permanently)
b) Redirects the browser automatically
c) Throws an exception
d) Behaves like `sendRedirect()`

JSP Code-Based Questions (10)

11. What does this JSP code output?

```
<%! int x = 10; %>
<%= x %>
<% x++; %>
<%= x %>
```

a) `10 11`
b) `10 10`
c) Compilation error
d) Runtime error

---

12. What is the output of this JSP code?

```
<%@ page isThreadSafe="false" %>
<%! int counter = 0; %>
<%= counter++ %>
```

a) Ensures thread safety
b) Displays counter incrementally per request
c) Compilation error
d) Throws an exception

---

13. What does this JSP scriptlet do?

```
<%
session.invalidate();
out.println(session.getId());
%>
```

a) Prints the session ID
b) Throws `IllegalStateException`

c) Prints `null`
d) Compilation error

---

14. What is the scope of a variable declared inside `<%! %>`?

a) Page scope
b) Application scope
c) Instance variable scope
d) Request scope

---

15. What is the output of this JSP snippet?

```
<jsp:useBean id="user" class="java.lang.String"/>
<jsp:setProperty name="user" property="*" />
```

a) Compilation error
b) Runtime exception
c) Stores an empty string
d) Works fine

---

16. What does this JSP implicit object do?

```
<%= application.getAttribute("globalMessage") %>
```

a) Retrieves a session attribute
b) Retrieves an application-wide attribute
c) Throws an error if not set
d) Retrieves a request attribute

---

17. What happens if we include a file with `<jsp:include>`?

```
<jsp:include page="header.jsp" />
```

a) Dynamically includes the content
b) Statically includes content at compile time
c) Throws an error
d) Works only inside `scriptlets`

---

18. What happens if we use `pageContext.forward("next.jsp");` after response output?

a) Works normally
b) Throws `IllegalStateException`
c) Redirects the page
d) Compilation error

---

19. What does this JSP directive do?

```
<%@ page buffer="none" %>
```

a) Disables response buffering
b) Throws an error
c) Buffers with default value
d) Ignores buffering

---

20. What does this JSP directive do?

```
<%@ page autoFlush="false" %>
```

a) Prevents output buffer flushing
b) Flushes response automatically
c) Throws exception if buffer overflows
d) Both (a) and (c)

Servlet Code-Based Questions Answers

1. **(c)** Runtime Exception (`IllegalStateException` occurs because `sendRedirect()` is called after writing to the response).
2. **(b)** `null` (The session is invalidated, so all attributes are lost).
3. **(c)** Both (a) and (b) (600 seconds = 10 minutes).
4. **(b)** A 405 HTTP error occurs (`doPost()` is not overridden).
5. **(c)** Runtime error (`IllegalStateException` occurs when trying to write after closing `PrintWriter`).

6. **(c)** Throws `IllegalStateException` (Output is already written, so forward() cannot be used).
7. **(a)** Deletes the cookie (`setMaxAge(0)` instructs the browser to remove it).
8. **(a)** `"Before Servlet"` is printed, request is processed, then `"After Servlet"` is printed.
9. **(c)** Runtime error (`IllegalStateException` occurs because response output is already committed).
10. **(a)** Sends an HTTP 301 response (browser may cache it and require `Location` header for redirection).

JSP Code-Based Questions Answers

11. **(a)** `10 11` (`<%! %>` declares instance variables, so `x` persists across requests).
12. **(b)** Displays counter incrementally per request (`isThreadSafe=false` synchronizes requests but does not reset the counter).
13. **(b)** Throws `IllegalStateException` (Session is invalidated before accessing `session.getId()`).
14. **(c)** Instance variable scope (`<%! %>` defines a class-level instance variable).
15. **(a)** Compilation error (A `String` cannot be used as a JavaBean).
16. **(b)** Retrieves an application-wide attribute (`application` refers to `ServletContext`).
17. **(a)** Dynamically includes the content (`<jsp:include>` processes at request time).
18. **(b)** Throws `IllegalStateException` (Response is already committed).
19. **(a)** Disables response buffering (`buffer="none"` means output is immediately written to response).
20. **(d)** Both (a) and (c) (`autoFlush=false` prevents automatic flushing, but throws an error if buffer overflows).
