

2과목	JSP 프로그래밍	(36~60)
출제위원 : 방송대 김희천		
출제범위 : 교재전체 (해당 멀티미디어강의 포함)		

36. 다음이 설명하는 HTTP 요청 방식은 무엇인가? (2점)

원하는 방식으로 인코딩된 데이터를 요청 메시지의 몸체에 포함하여 전송하면서 자원을 요청하는 경우 사용됨

- ① GET
- ② HEAD
- ③ POST
- ④ FTP

37. “Eclipse를 이용한 웹 프로젝트의 배포”가 의미하는 것은? (2점)

- ① 웹 프로젝트 폴더를 압축 파일로 만드는 것이다.
- ② 개발한 웹 프로젝트를 웹 컨테이너에 내보내는 것이다.
- ③ 웹 컨테이너를 서버로 등록시키는 것이다.
- ④ 버퍼의 내용을 일차적으로 클라이언트에 보내는 것이다.

38. JSP 페이지에서 사용할 메소드나 변수를 선언하기 위한 태그는 다음 중 어느 것인가? (4점)

- ① <% ... %>
- ② <%! ... %>
- ③ <%@ ... %>
- ④ <%# ... %>

39. 서블릿 클래스에서 doGet( ) 메서드의 인자는 무엇인가? (3점)

- ① request 1개
- ② response 1개
- ③ request와 response
- ④ 인자가 없음

40. 다음은 버퍼가 차면 내용을 클라이언트로 보내고 버퍼를 비우는 의미를 가진다. 밑줄 부분에 들어갈 단어는 무엇인가? (4점)

<%@ page \_\_\_\_\_ = "true" %>

- ① flush
- ② autoFlush
- ③ buffer
- ④ autoBuffer

41. 다음 중 JSP의 내장 객체가 아닌 것은? (3점)

- ① request
- ② response
- ③ out
- ④ cookie

※ 다음 코드를 보고 물음에 답하여라. (42~43)

```
<%@ page contentType="text/html; charset=euc-kr" %>
<HTML>
<BODY>
<%   int data = 20;   %>
including.jsp : <%= data %><br>
<jsp:include page="included.jsp" />
including.jsp : <%= data %>
</BODY>
</HTML>
```

including.jsp

```
<%@ page contentType="text/html; charset=euc-kr" %>
<HTML>
<BODY>
<%   int data = 1000;   %>
included.jsp : <%= data %><br>
</BODY>
</HTML>
```

included.jsp

42. including.jsp 페이지를 요청했을 때, 클라이언트의 브라우저에서 표시되는 결과는 무엇인가? (3점)

- ① included.jsp : 1000
- ② including.jsp : 20
- including.jsp : 20
- ③ including.jsp : 20
- ④ including.jsp : 20
- included.jsp : 1000
- included.jsp : 1000
- including.jsp : 20
- including.jsp : 1000

43. including.jsp와 included.jsp 페이지 사이에서 공유될 수 없는 영역은 무엇인가? (3점)

- ① page
- ② request
- ③ session
- ④ application

44. JSP 페이지의 모듈화를 위해 사용하는 방법이라고 할 수 없는 것은? (3점)

- ① page 지시어
- ② include 지시어
- ③ include 액션 태그
- ④ prelude와 coda

45. 다음 중 <jsp:forward ... /> 액션의 처리를 바르게 설명한 것은? (3점)

- ① 실행 결과의 맨 앞이나 맨 뒤에 자동으로 코드를 포함시킨다.
- ② 속성으로 지정된 페이지의 실행 결과를 태그가 위치한 부분에 삽입시킨다.
- ③ <jsp:include> 액션의 서브 요소로만 사용된다.
- ④ 페이지의 실행이 종료되고 새로운 페이지로의 이동이 일어난다.

46. 쿠키의 유효 시간에 관한 설명으로 올바른 것은? (2점)

- ① 클라이언트 컴퓨터에서 쿠키가 유지되는 기간이다.
- ② 쿠키의 유효 시간이 음수라면 영구히 보관된다.
- ③ 클라이언트의 추가 요청이 없어도 쿠키가 보존되는 기간이다.
- ④ 출력 버퍼에서 쿠키 값을 변경할 수 있는 최대 시간이다.

47. 다음 코드의 의미를 바르게 설명한 것은? (3점)

```
<% Cookie cookie=new Cookie("name","value");
response.addCookie(cookie); %>
```

- ① 쿠키를 만들어 요청 메시지의 헤더에 추가한다.
- ② 쿠키를 만들어 응답 메시지의 헤더에 추가한다.
- ③ 쿠키를 만들어 응답 메시지의 몸체에 추가한다.
- ④ 클라이언트가 보낸 요청 메시지에서 쿠키 값을 읽는다.

48. 세션을 사용하여 로그인/로그아웃을 처리하는 JSP 페이지를 구현하려고 한다. 다음 중 로그인 처리에 필요한 작업은 무엇인가? (2점)

- ① request 영역에 session 객체를 저장한다.
- ② session 객체의 타임아웃 시간을 수정한다.
- ③ session 객체에 사용자 확인을 위한 속성을 추가한다.
- ④ session 객체를 삭제한다.

49. 다음 중 세션이 종료되는 상황이 아닌 것은? (4점)

- ① 클라이언트가 웹 브라우저를 종료한다.
- ② 세션의 유효시간 동안 클라이언트의 추가 요청이 없었다.
- ③ JSP 페이지에서 session.invalidate( )를 실행하였다.
- ④ JSP 페이지에서 session 객체에 저장된 모든 속성을 삭제하였다.

※ 다음 코드를 보고 물음에 답하여라. (50~51)

```
<%-- 중간 생략 --%>
<jsp:useBean id="meminfo" class="member.MemberInfo" scope="request" />
<%
    meminfo.setAge(10);
    meminfo.setName("Jimmy");
%>
MemberInfo name : _____ <br>
<%-- 중간 생략 --%>
```

50. 위에서 사용된 자바빈 객체에 관한 설명으로 **잘못된** 것은? (2점)

- ① member.MemberInfo는 클래스의 이름에 해당한다.
- ② meminfo라는 이름의 객체가 존재하지 않으면 자바빈 객체를 생성한다.
- ③ 객체가 가진 속성 name과 age에 값을 지정하고 있다.
- ④ 객체의 사용 범위가 page이므로 해당 페이지에서만 사용할 수 있다.

51. 자바빈 객체의 이름을 출력하기 위해 밑줄 친 부분에 들어갈 **적당한** 내용은? (2점)

- ① <jsp:getProperty name="meminfo" property="name" />
- ② <jsp:getParameter id="meminfo" property="name" />
- ③ <jsp:setProperty name="meminfo" property="name" />
- ④ < out.print(meminfo.getName( )); >

52. 표현 언어(EL)의 사용에 관한 설명이다. **잘못된** 것은? (2점)

- ① 표현식을 대신하여 자주 사용된다.
- ② 태그의 속성 값을 지정할 때 표현 언어 수식을 사용한다.
- ③ 표현 언어의 내장 객체와 JSP의 내장 객체는 다른 것이다.
- ④ 스크립트 요소의 내부에서 사용해야 한다.

53. <%= request.getParameter("name") %>과 같은 의미를 가지는 **표현 언어** 수식을 내장 객체 param을 사용하여 표현하면 무엇인가? (2점)

- ① \${ param.name }                      ② \${ param[name] }
- ③ \$( param.name )                      ④ \$( param[name] )

※ 다음은 JSTL 코어 라이브러리를 사용하는 예이다. 물음에 답하여라. (54~55)

```
_____ prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%-- 중간 생략 --%>
<c:set var="map" value="<%= map %>" />
<_____ var="m" items="{m.key} = {m.value}">
    {m.key} = {m.value} <br>
</_____>
<%-- 중간 생략 --%>
```

54. 코어 라이브러리를 사용하려면 지시어를 추가해야 한다. 밑줄 친 \_\_\_\_\_에 들어가야 할 내용은? (4점)

- ① <%@ import                              ② <%= page
- ③ <%@ taglib                              ④ <%# taglib

55. 밑줄 친 \_\_\_\_\_에는 흐름 제어를 위한 태그가 들어가야 한다. **적당한** 것은 무엇인가? (3점)

- ① c:choose                                  ② c:forEach
- ③ x:import                                  ④ x:forTokens

56. 다음 중 MySQL과 연동하는 JDBC 프로그래밍 작업을 위한 것은 무엇인가? (2점)

- ① Oracle 사이트에서 php-mysql 패키지를 다운받아 설치함
- ② Eclipse 사이트에서 Tomcat을 다운받아 설치함
- ③ MySQL 사이트에서 Connector/J를 다운받아 설치함
- ④ MySQL 사이트에서 MySQL Shell을 다운받아 설치함

57. MVC 패턴을 적용하여 컨트롤러를 구현할 때 커맨드 패턴을 적용할 수 있다. 이때 커맨드와 관련된 정보를 별도의 설정 파일로 만든다면 어떤 장점이 있는가? (2점)

- ① 새로운 커맨드를 추가하면 설정 파일을 다시 읽어야 한다.
- ② 새로운 커맨드를 추가할 경우 소스 코드를 수정해야 한다.
- ③ 새로운 커맨드가 추가되도 설정 파일을 수정할 필요가 없다.
- ④ 새로운 커맨드가 추가되도 소스 코드를 수정할 필요가 없다.

58. 다음은 회원정보 저장을 위한 자바빈 클래스이다. 속성 id의 **getter** 메서드와 **setter** 메서드를 올바르게 작성한 것은? (3점)

```
public class MemberInfo {
    private String id;
    // 중간 생략
}
```

- ① public String getId( ) { return id; }  
public void setId(String id) { id = id; }
- ② public void getId(String id) { this.id = id; }  
public String setId( ) { return id; }
- ③ public String getId( ) { return this; }  
public void setId(String id) { this.id = id; }
- ④ public String getId( ) { return id; }  
public void setId(String id) { this.id = id; }

※ 다음은 요청 파라미터로 전달된 아이디와 암호를 읽고, 회원이 맞는 경우 로그인 처리를 수행하는 login.jsp 페이지의 일부이다. 물음에 답하여라. (59~60)

```
<%
    String id = request.getParameter("id");
    String pass = request.getParameter("pass");
    if (data.isMember(id, pass)) {
        _____("ID", id);
        _____("member_info.jsp");
    }
    else
        // 중간 생략
%>
```

59. 세션 영역에 회원 아이디를 "ID"라는 이름의 속성으로 저장하려 한다. 밑줄 친 \_\_\_\_\_에 들어가야 할 내용은? (3점)

- ① session.getAttribute
- ② session.setAttribute
- ③ cookie.getAttribute
- ④ cookie.setAttribute

60. 로그인 처리가 끝난 후 다음 페이지로 이동하기 위해 밑줄 친 \_\_\_\_\_에 들어가야 할 내용은? (4점)

- ① request.forward
- ② request.sendRedirect
- ③ response.forward
- ④ response.sendRedirect