Part Multiple-Choices (A)

1. Which of the following is example of JSP tag ?

   a. include b. extends c. import

   d. setSession e. isError

2. How many instances of a filter does the servlet container create ?

   a. One instance of a filter for each filter class.

   b. Depend on implementation of container

   c. One instance of a filter for each filter element defines the deployment descripter

   d. One instance of a filter for each filter-mapping element defined in the deployment descripter.

   e. One instance of a filter for each servlet-context.

3. How many instances of a JSP Servlet does the container create ?

   a. One instance

   b. 2 Dependent on Implementation of Container

   c. 1-3 instances

   d. 4 Dependent on number of request

   e. None of these.

4. A Session cookie is _____ when the browser is closed ?

   a. unavailable b. deleted c. Store to cookie server

   d. store to the client computer e. None of the above

5. There are five checkboxes on a form and all of them have the same name, music. Assume all of them are checked. The code: request.getParameter(“music”) on the web server (JSP Page) returns _____.

   a. a string containing all values of the five checkboxes.

   b. a string containing the value of the first checkboxes.

   c. null

   d. a string array containing alll values of the fives checkboxes

   e. An exception occurs

Part Multiple-Choices (B)

1. Which one is **not** Servlet characteristic?
   a. It runs in server tier
   b. It is pure java language
   c. File extension is .svl
   d. It manage by web container
   e. None of above
2. Creating Servlet which method did you have to override?
   a. doGet
   b. doPost
   c. processRequest
   d. a and b
   e. none of above
3. How did servlet get request message?
   a. from URL query string www.action.com?name=value&name=value
   b. from HTTP header
   c. from submitted http form
   d. all of above
   e. none of above
4. Which object that use for writing output on web browser?
   a. PrintStream
   b. PrintWriter (PrintWriter out = response.getWriter())
   c. HttpWriter
   d. OutWriter
   e. None of above
5. Which directory that have to contain web.xml file?
   a. META-INF
   b. classes
   c. WEB-INF (อยู่ใน Web Pages/WEB-INF โดยเก็บ web.xml และ glassfish-web.xml)
   d. lib
   e. none of above

6. which directory that use for collected the import library (such as derbyclient.jar or another external jar file)?

   a. lib
   b. tlds
   c. classes
   d. WEB-INF
   e. None of above

7. For the servlet lifecycle which step that servlet invoke method doGet, doPost?

   a. Create
   b. Initial
   c. Servicing Request
   d. Unload
   e. None of above

8.

```
<servlet>
        <display-name>TestServlet</display-name>
        <servlet-name>TestServlet</servlet-name>
        <servlet-class>int303.exam.servlet.TestServlet</servlet-
    class>
</servlet>
<servlet-mapping>
        <servlet-name>TestServlet</servlet-name>
        <url-pattern>/Test</url-pattern>
</servlet-mapping>
<servlet-mapping>
        <servlet-name>TestServlet</servlet-name>
        <url-pattern>/TestServ</url-pattern>
</servlet-mapping>
```

KEY การเรียก

```
<servlet>
        <servlet-name>AddServlet</servlet-name>      -------> ชื่อ
        <servlet-class>servlet.AddServlet</servlet-class>       ------> Class
 </servlet>
<servlet-mapping>
        <servlet-name>AddServlet</servlet-name>
          <url-pattern>/AddServlet</url-pattern>                  -------> เรียก
          <url-pattern>/Add</url-pattern>
</servlet-mapping>
```

from this web deployment descriptor how can we invoke the TestServlet?

a. Invoke via URL [http://localhost:xxxx/application or project name]/Test
b. Invoke via form action = “TestServlet” **ชื่อ ไม่ใช่ Url ผิด
c. Invoke via forwarding requestDispatcher(“/TestServ”)
d. a and c
e. none of above

9. If you invoke wrong URL or the URL that did not have file or servlet, what is the status code of that event?

a. 404
b. 403
c. 200
d. 500 **เกิด Error ใน Server เช่น Null Pointer
e. None of above

10.

```
///// Servlet template/////
        String name = getServletConfig.getInitParameter(“num”);
        String num = getServletConfig.getInitParameter(name);
        out.print(num);
```

Experiment

String name = getServletConfig().getInitParameter("num"); ----> เรียก Num = “Name”

String num = getServletConfig().getInitParameter(name); ----> เรียก Name = “Wellcome”

out.println(num);

out.println(name);

***.getServletContext().getRequestDispatcher

```
<servlet>
            <display-name>TestServlet</display-name>
            <servlet-name>TestServlet</servlet-name>
            <servlet-class>int303.exam.servlet.TestServlet</servlet
        -class>
            <init-param>
                <param-name>num</param-name>
                <param-value>Name</param-value>
            </init-param>
            <init-param>
                <param-name>name</param-name>
                <param-value>Wellcome</param-value>
            </init-param>
</servlet>
```

From this code what is the output on web browser?

a. Name
b. Welcome **ถ้า Syntax ถูกจะให้คำนี้
c. num
d. Page Error
e. null **Syntax ผิดเป็นหน้าว่าง

11. Which method that use to get data for ***text field*** in html form?

String text = request.getParameter("name");

a. getInitParameter(String parameterName)
b. getParameterValues(String name)
c. getParameter(String)
d. getParameterNames()
e. None of above

12. What is the benefit of MVC concept?

a. Perform reuse code
b. Reduce develop time (can develop many parts in parallel)
c. Increase performance (all parts can develop by the suitable people)
d. All of above
e. None of above

13. What is the difference between forward and include?

a. The forwarding is to send all request and response to another page or servlet and the former page or servlet cannot modify response any more, but include can modify response after all.
b. The forwarding is to send request and response to another page or servlet and the former page or servlet can modify response after the forwarded page finish using the response, but include cannot do it at all.
c. The forward and include is the same methodology but it use the difference file path. For the forward use absolute path, but include use relative path
d. Forwarding can forward to servlet and jsp file, but include can use with servlet only

e. None of above

14. Which method that use for sharing variable or object between multiple page or servlet?

a. addValue(String name, Object value)

b. setParameter(String name, Object value)

c. setAttribute(String name, Object value)

d. addParameter(String name, Object value)

e. none of above

15. What is the characteristic of JSP?

a. JSP can mix the static and dynamic content

b. JSP allows the server script

c. File extension is .jsp

d. All of above

e. None of above

16. Which one is **not** the JSP syntax?

a. Declaration **ส่วนประกอบย่อยของ Scripting (Expressions, Scriptlets)

b. Scripting <% int x = 5 %>

c. Directive <%@ taglib prefix="prefixOfTag" uri="uri" %>

d. Action <c:forEach >

e. Comment

17. What is the JSP benefit?

a. Separate the presentation module out of servlet (Controller)

b. Can be reuse by many servlet

c. Support with the web design tools (Bootstrap)

d. Compatible with MVC concept

e. All of above

18. Which one is **not** step of converting JSP into Servlet?

KEY : JSP source is parsed, Java servlet code is generated, JSP servlet is compiled, loaded, and run

a. Convert JSP to Servlet code
b. Compile the Servlet code
c. Load the Object of that JSP into Web Container
d. Generate Servlet-Mapping for the generated Servlet code
e. None of above

19. What is the parameter that use for precompiling the JSP?

a. _jsp_compile
b. jsp_compile
c. jsp_precompile
d. _jsp_precompile
e. _precompile

20. What is the reserve prefix of jsp parameter?

a. _jsp
b. jsp_
c. jsp-
d. servlet_
e. _servlet

21. What is the scope of sharing data?

a. Page
b. Session
c. Application
d. All of above
e. None of above

22. What is the jsp directive syntax? <%@ directive {attribute="value"}* %>

a. <%@.....%>
b. <%.....%> Scripting
c. <!--.....--> Comment
d. <%jsp:.....%>
e. <%=.....%> Implicit <%= session.getAttribute(“name”) %>

23. What is the default value of directive “page” attribute “session”?

    a. Null
    b. False
    c. True
    d. Create
    e. None of above

24. What is the syntax of the declaration scripting?

    a. <%!.....%>
    b. <%!.....!%>
    c. <%=.....%>
    d. <%.....%>
    e. <?.....?>

25. What did the scripting expression used for?

    a. For writing the complex script (if-else, loop) JSTL
    b. For declared variable
    c. For comment source code
    d. For display the calculated value
    e. None of above

26. Which one **did not** define in jsp implicit object?

    a. HttpResponse
    b. ServletContext
    c. HttpSession
    d. PageContext
    e. PrintStream

27. Which one is not the implicit variable in jsp?

    a. session
    b. exception
    c. application
    d. out
    e. none of above

28. How could you invoke jsp?
    a. Via URL
    b. Via another Servlet forward or include
    c. Via another jsp by link
    d. All of above
    e. None of above

29. What is the disadvantage of inserting scripting in jsp?
    a. It will display slower
    b. It will make the page contain multiple language (HTML,Java)
    c. It difficult for designer to create and understand
    d. b and c
    e. all of above

30. If the scripting of jsp is cause of the exception what status code will be display?

```
<c:forEach items="${sessionScope.allStudent.allData}" var="student" varStatus="vs">
```

    a. 404
    b. 403
    c. 500
    d. 200
    e. 300

31. Which one is not the session strategies solution?
    a. Cookie **Client
    b. Hidden Fields **Client
    c. Content Base Routing **Server
    d. URL Rewriting **Client
    e. None of above

32. What is the characteristic of cookie?
    a. Store in client computer
    b. Save in text file
    c. Cannot be delete by user until it timeout
    d. A and b
    e. None of above

33. Which method use for saving cookie into client?

KEY : HttpServletResponse.addCookie[Cookie aCookie]

a. setCookie(Cookie c)
b. addCookie(Cookie c)
c. writeCookie(Cookie c)
d. saveCookie(Cookie c)
e. none of above

34. Which one of client can set accepting cookie from server?

a. Web browser
b. Windows firewall
c. Administrative tools
d. Network and security
e. None of above

35. How to delete cookie in client via server command?

Transient and expired cookies ***will be deleted*** by the browser according to its own approach.

a. setMaxAge(-1) **session cookie
b. setMaxAge(0) **A MaxAge of 0 is a request for the browser to delete the cookie.
c. deleteCookie(String cookieName)
d. cannot delete cookie via server command
e. none of above

36. How to get HttpSession session object

HttpSession session = request.getSession(true);

a. request.getSession()
b. response.getSession()
c. getServletContext.getSession()
d. getServletConfig.getSession()
e. none of above

37. For the method getSession there is one parameter that boolean type. What is the difference between getSession(true) and getSession(false)

    a. If use getSession(true) its mean is the system already have the HttpSession object it will be retrieve, *but if did not exist it will get null value. If use getSession(false) it will create new HttpSession instate*

    b. For the method getSession() if the system already have HttpSession object it will retrieve that object, if not have HttpSession object the system will check for the paramer. If true it will create new HttpSession object, if false it will return null value

    c. If use getSession(true) that mean it allows this servlet to use HttpSession. If use getSession(false) it will not allows this servlet to use HttpSession if you use it will throw NullPointerException

    d. If use getSession(true) that mean it allows to sharing HttpSession with another page or servlet. If use getSession(false) it will not allows another page or servlet to use sharing variable in this HttpSession

    e. None of above

38. How to clear all value in session?

    a. session.removeAttribute()

    b. session.invalidate()

    c. session.clear()

    d. session.destroy()

    e. session.clearValues()

39. What is the ratio of session?

    a. One session : one browser(IE, Chrome, Safari)

    b. One session : one client

    c. One session : one server

    d. One session : one browse tab

    e. None of above

40. What is the way that can cause session lost?

a. Client close web browse
b. Server destroy session object
c. Session timeout
d. All of above
e. None of above

41. What is the difference between forward and sendRedirect?

a. The forwarding is the way to change page by server invoke the new page, but the sendRedirect is the way that the sever send back to the invoker page and then make the client change page automatically
b. For the forward it will use the relative path, but the sendRedirect use absolute path สลับกัน
c. The forward manage by RequestDispatcher. The sendRedirect manage by response

request.getServletContext().getRequestDispatcher("/xxxx").forward(req,res)

d. All of above
e. None of above

46. How to optimize the session object on server?

a. Destroy session object if it did use anymore
b. Remove the no use attribute
c. Set the suitable timeout of session
d. All of above
e. None of above

47. How to set session timeout

KEY: The default timeout is 30 minutes

session.setMaxInactiveInterval(int) can provide session-specific timeout value

a. setSessionTimeout(int second)
b. setSessionTimeLimit(int second)
c. setMaxInactiveInterval(int second)
d. setMaxTimeout(int second)
e. setLimitTimeout(int second)

48. What is the characteristic of HttpSession

a. Provide by HttpServletRequest
b. Can collect multiple object via name and value (similar to Map)
c. Can sharing object on multiple page and servlet
d. Can be destroy by server
e. All of above

49. What will happen if setting the session timeout very fast

a. Increase server performance because it will clear session very fast
b. Improve the performance of using server memory because it can contain many session ,therefore it clear the former session very fast
c. User cannot create transaction well, because of the transaction data are collected in session and the session is clear too fast
d. All of above
e. None of above

50. What will happened if setting session timeout very long

a. Decrease server performance because it cannot clear session as well
b. Lack of server memory because server have to collect many of session in long time
c. Client can absolutely complete transaction because session did not lost
d. a and b
e. None of above

Part Multiple-Choices (C)

1. How is a request **dispatched** to hello.jsp from a doGet() method? (1 correct answer)

    1. request.getRequestDispatcher().forward("hello.jsp");
    2. request.getRequestDispatcher().dispatch("hello.jsp");
    3. request.getRequestDispatcher("hello.jsp").forward(request, response);
    4. request.getRequestDispatcher("hello.jsp").dispatch(request, response);

2. How is a request **redirected** to hello.jsp from a doGet() method? (1 correct answer)

    1. request.redirect("hello.jsp");
    2. response.redirect("hello.jsp");
    3. request.sendRedirect("hello.jsp");
    4. response.sendRedirect("hello.jsp");

10. Dispatching a request occurs on the server-side and redirection on the client-side. (1 correct answer)

    1. True 2. false

11. Both context init parameters and servlet init parameters are declared in the web.xml. (1 correct answer)

    1. True 2. false

12. The value of a servlet init parameter can be changed programmatically, but the value of a context init parameter cannot. (1 correct answer)

    1. True 2.false

13. A context init parameter cannot have the same name with a servlet init parameter. (1 correct answer)

    1. True 2.false

14. A servlet init parameter cannot have the same name with the servlet it refers to. (1 correct answer)

    1. True 2.false

15. Where is a servlet init parameter stored after the servlet is initialized and available for use? (1 correct answer)

    1. In the ServletConfig object of the servlet.
    2. In the ServletContext object of the web application.

16. Where is a context init parameter stored after the servlet is initialized and available for use? (1 correct answer)

    1. In the ServletConfig object of the servlet.
    2. In the ServletContext object of the web application.

17. Assume the servlet **HelloServlet** that belongs to package **com**. The file HelloServlet.class is placed in the directory WEB-INF/classes/com. Is this a correct declaration of an init parameter for this servlet? (1 correct answer)

```
<servlet>
    <servlet-name>Hello Servlet</servlet-name>
    <servlet-class>classes.com.HelloServlet</servlet-class>
    <init-param>
        <param-name>this</param-name>
        <param-value>Hello!</param-value>
    </init-param>
</servlet>
```

    1. Yes.
    2. No, because servlet-name contains a space.
    3. No, because servlet-class has a wrong value.
    4. No, because param-name is a reserved Java keyword.
    5. No, because param-value contains an explanation mark (!).
    6. No, because init-param should be inside a servlet-mapping element.

18. What happens when we compile and deploy this servlet? (1 correct answer)

```
public class Test extends HttpServlet {


}
```

    1. Compilation fails because there is no init() method defined.
    2. An exception is thrown at runtime because service() has no method to call!
    3. Deployment succeeds but we get a message “GET is not supported by this URL” if we access it.

20. What happens when this servlet is compiled, deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
    public void doGet(HttpServletRequest request,HttpServletResponse response)
                    throws ServletException, IOException {
        // TODO
    }
}
```

1. Compilation fails because doGet() is empty.
2. An exception is thrown at runtime because doGet() is empty.
3. Deployment succeeds but nothing is displayed to the user's browser.

21. What happens when this servlet is compiled, deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
 public String doGet(HttpServletRequest request,HttpServletResponse response)
                    throws ServletException, IOException {
        // TODO
        return null;
        }
}
```

1. Compilation fails because doGet() must be void.
2. Deployment succeeds but nothing is displayed to the user's browser.
3. A NullPointerException is thrown at runtime because null is returned.
4. A ServletException is thrown at runtime because service() cannot find the proper doGet() method.

22. What happens when this servlet is compiled and deployed? (1 correct answer)

```
public class Test extends HttpServlet {
   protected void doGet(HttpServletRequest request,HttpServletResponse response)
                    throws IOException {
      response.setContentType("text/html; charset=UTF-8");
      PrintWriter out = response.getWriter();
      out.println("<html><body>");
      out.println("<p>Hello!!</p>");
      out.println("</body></html>");
      out.close();
   }
}
```

1. Compilation fails because doGet() is protected.
2. Compilation fails because doGet() does not declare a ServletException.
3. Deployment succeeds and clients are served just fine.

23. What happens when this servlet is deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
      public void doGet(final HttpServletRequest request,final HttpServletResponse response)
                throws IOException, ServletException {
      response.setContentType("text/html; charset=UTF-8");
      final PrintWriter out = response.getWriter();
      out.println("<html><body>");
      out.println("<p>Hello!!</p>");
      out.println("</body></html>");
      out.close();
       }
}
```

1. An exception is thrown at runtime because the Container cannot modify the **request** and **response**objects.
2. An exception is thrown at runtime when the **out**object is closed.
3. Deployment succeeds and clients are served just fine.

24. What happens when this servlet is deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
        public void doGet(HttpServletRequest request,HttpServletResponse response)
                throws ServletException, IOException {
        response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();
        // out.println("<html><body>");
        out.println("<p>Hello!!</p>");
        // out.println("</body></html>");
        out.close();
        }
}
```

1. A ServletException is thrown at runtime because the <html> and <body> tags are missing.
2. Deployment succeeds and **Hello!!** is presented on the browser.
3. The server responds with a HTTP status code 404: “Not Found”.

25. What happens when this servlet is compiled, deployed and called? (1 correct answer)

```
public class Test extends HttpServlet {
        public void doGet(HttpServletRequest request,HttpServletResponse response)
                throws ServletException, IOException {
        // response.setContentType("text/html; charset=UTF-8");
        PrintWriter out = response.getWriter();
        out.println("<html><body>Hello!!</body></html>");
        out.close();
        }
}
```

1. Deployment succeeds and **Hello!!** is presented on the browser.
2. Compilation fails because the content type should be specified before any output is written.
3. An exception is thrown at runtime because the response has not an explicitly set content type.

31. Which of the following statements shoud be inserted for a successful compilation? (1 correct answer)

```
public class Test extends HttpServlet {
    protected void doGet(HttpServletRequest request,HttpServletResponse response){
        // insert statement
    }
}
```

1. request.getRequestDispatcher(“hello.jsp”).forward(request, response);
2. request.getRequestDispatcher(“hello.jsp”).forward(response, request);
3. response.getRequestDispatcher(“hello.jsp”).forward(request, response);
4. response.getRequestDispatcher(“hello.jsp”).forward(response, request);
5. None of the above.

32. What happens when the servlet with the following method is deployed and called? (1 correct answer)

```
1       public void doGet(HttpServletRequest request,HttpServletResponse response)
2                       throws ServletException, IOException {
3           response.getWriter().print('a');
4           request.getRequestDispatcher("hello.jsp").forward(request, response);
5           response.getWriter().print('a');
6           response.getWriter().close();
7       }
8
```

1. An IllegalStateException is thrown at runtime because response.getWriter() is called more than once.
2. An IllegalStateException is thrown at runtime at line 5 because the request is dispatched after writing data.
3. An IllegalStateException is thrown at runtime at line 6 because data is written after the request has been dispatched.
4. An IllegalStateException is thrown at runtime at line 7 because the writer is closed after the request has been dispatched.
5. The browser displays the content of **hello.jsp** without any exception at runtime.
6. The browser displays **aa** without any exception at runtime.