1. Which of the following are interfaces? (3 correct answers)
    a. Servlet
    b. HttpServlet
    c. ServletRequest
    d. HttpServletRequest

2. Which of the following are abstract classes? (2 correct answers)
    a. Servlet
    b. HttpServlet
    c. GenericServlet
    d. HttpServletRequest

3. Which of the following statements is true? (1 correct answer)
    a. HttpServlet extends GenericServlet that implements Servlet.
    b. HttpServlet extends GenericServlet that extends Servlet.
    c. HttpServlet implements GenericServlet that extends Servlet.

4. Which of the following statements are true? (2 correct answers)
    a. HttpServlet IS-A GenericServlet.
    b. HttpServlet IS-A Servlet.
    c. HttpServlet IS-A ServletRequest.

5. Here are some actions taken by the Container when a client request arrives. Place them in the correct order starting from what happens *first*.
    a. Calls the void service(HttpServletRequest, HttpServletResponse) method of the servlet.
    b. Creates a pair of request and response objects.
    c. Finds the correct servlet based on the URL.
    d. Creates a new thread or allocates an existing thread to the client's request.

    (b,c,d,a)

6. How can a servlet access it's associated ServletConfig object? (1 correct answer)
    a. getServletConfig();
    b. request.getServletConfig();
    c. response.getServletConfig();

d. getServletContext().getServletConfig();
e. request.getSession().getServletConfig();

7. How can a servlet access the application’s ServletContext object? (3 correct answers)

a. getServletContext();
b. request.getServletContext();
c. response.getServletContext();
d. getServletConfig().getServletContext();
e. request.getSession().getServletContext();

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

a. request.getRequestDispatcher().forward(“hello.jsp”);
b. request.getRequestDispatcher().dispatch(“hello.jsp”);
c. request.getRequestDispatcher(“hello.jsp”).forward(request, response);
d. request.getRequestDispatcher(“hello.jsp”).dispatch(request, response);

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

a. request.redirect(“hello.jsp”);
b. response.redirect(“hello.jsp”);
c. request.sendRedirect(“hello.jsp”);
d. response.sendRedirect(“hello.jsp”);

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

a. true
b. false

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

a. true
b. 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)

    a. true

    b. false

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

    a. true

    b. false

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

    a. true

    b. false

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

    a. In the ServletConfig object of the servlet.

    b. 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)

    a. In the ServletConfig object of the servlet.

    b. 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>
```

a. Yes.
b. No, because servlet-name contains a space.
c. No, because servlet-class has a wrong value.
d. No, because param-name is a reserved Java keyword.
e. No, because param-value contains an explanation mark (!).
f. 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 {

}
```

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

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

```
class Test extends HttpServlet {

}
```

a. Compilation fails because there is no init() method defined.
b. An exception is thrown at runtime because the servlet has no modifier.
c. An exception is thrown at runtime because service() has no method to call!
d. Deployment succeeds but we get a message “GET is not supported by this URL” if we try to access its URL.

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
    }
}
```

a. Compilation fails because doGet() is empty.

b. An exception is thrown at runtime because doGet() is empty.
c. 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;
}
}
```

a. Compilation fails because doGet() must be void.
b. Deployment succeeds but nothing is displayed to the user’s browser.
c. A NullPointerException is thrown at runtime because null is returned.
d. 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();
}
}
```

a. Compilation fails because doGet() is protected.
b. Compilation fails because doGet() does not declare a ServletException.
c. 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();
}
}
```

a. An exception is thrown at runtime because the Container cannot modify the **request** and **response** objects.

b. An exception is thrown at runtime when the **out** object is closed.

c. 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();
}
}
```

a. A ServletException is thrown at runtime because the <html> and <body> tags are missing.

b. Deployment succeeds and **Hello!!** is presented on the browser.

c. 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();
}
}
```

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

26. 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();
}
}
```

a. An exception is thrown at runtime because out is not closed.
b. Deployment succeeds and **Hello!!** is presented on the browser.
c. Deployment succeeds but no output is presented on the browser.

27. Does this servlet compile successfully? (1 correct answer)

```
1 public class Test extends HttpServlet {
2 public void doGet(HttpServletRequest request,
3                   HttpServletResponse response)
4                   throws Exception {
5     response.setContentType("text/html; charset=UTF-8");
6     response.getWriter().println("Hello!!").close();
7 }
8 }
```

a. Compilation succeeds.
b. There is a compilation error at line 4.
c. There is a compilation error at line 6.
d. Both lines 4 and 6 contain a compilation error.

28. Does this servlet compile successfully? (1 correct answer)

```
public class Test extends HttpServlet {
public void init(ServletConfig config) {
    super.init(config);
}
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();
}
}
```

a. Compilation succeeds.
b. Compilation fails because there is no init(ServletConfig) in GenericServlet.
c. Compilation fails because init(ServletConfig) of GenericServlet throws ServletException.
d. Compilation fails because init(ServletConfig) of GenericServlet throws IOException and ServletException.

29. Does this servlet compile successfully? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest req,
                  HttpServletResponse response)
                  throws IOException {
    req.getRequestDispatcher("index.html").
        forward(req, response);
}
}
```

a. Compilation succeeds.
b. Compilation fails because index.html is not a jsp.
c. Compilation fails because doGet should declare ServletException as well.
d. Compilation fails because the request object’s reference must be **request** and not **req**.

30. Does this servlet compile successfully? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest req,
                  HttpServletResponse response)
                  throws IOException {
```

```
    response.sendRedirect("index.html");
}
}
```

a. Compilation succeeds.
b. Compilation fails because index.html is not a jsp.
c. Compilation fails because doGet should declare ServletException as well.
d. Compilation fails because the request object’s reference must be **request** and not **req**.

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
    }
}
```

a. request.getRequestDispatcher(“hello.jsp”).forward(request, response);
b. request.getRequestDispatcher(“hello.jsp”).forward(response, request);
c. response.getRequestDispatcher(“hello.jsp”).forward(request, response);
d. response.getRequestDispatcher(“hello.jsp”).forward(response, request);
e. None of the above.

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

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

a. An IllegalStateException is thrown at runtime because response.getWriter() is called more than once.
b. An IllegalStateException is thrown at runtime at line 5 because the request is dispatched after writing data.
c. An IllegalStateException is thrown at runtime at line 6 because data is written after the request has been dispatched.

d. An IllegalStateException is thrown at runtime at line 7 because the writer is closed after the request has been dispatched.

e. The browser displays the content of **hello.jsp** without any exception at runtime.

f. The browser displays **aa** without any exception at runtime.

g. The browser displays **a** without any exception at runtime.

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

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    getServletContext().getRequestDispatcher("hello.jsp")
                       .forward(request, response);
}
```

a. An exception is thrown at runtime because the provided URL does not start with “/”.

b. Compilation fails because the provided URL does not start with “/”.

c. The browser displays the content of hello.jsp.

d. The browser displays an empty page.

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

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    getServletContext().getRequestDispatcher("/hello.jsp");
}
```

a. An exception is thrown at runtime because the provided URL is absolute.

b. Compilation fails because the provided URL is absolute.

c. The browser displays the content of hello.jsp.

d. The browser displays an empty page.

35. A servlet init parameter with name “ice” and value “tea” is properly declared. What is the result of this code? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
```

```
    response.getWriter().println(
        getServletConfig().getInitParameter("tea")
    );
}
```

a. A NullPointerException occurs at runtime.
b. The browser displays an empty page.
c. The browser displays “null”.
d. The browser displays “ice”.
e. The browser displays “tea”.
f. Compilation fails.

36. A servlet init parameter with name “ice” and value “tea” is properly declared. What is the result of this code? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(getInitParameter("ice"));
}
```

a. A NullPointerException occurs at runtime.
b. The browser displays an empty page.
c. The browser displays “null”.
d. The browser displays “ice”.
e. The browser displays “tea”.
f. Compilation fails.

37. A context init parameter with name “ice” and value “tea” is properly declared. What is the result of this code? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(
        getServletContext().getInitParameter("tea")
    );
}
```

a. A NullPointerException occurs at runtime.
b. The browser displays an empty page.
c. The browser displays “null”.
d. The browser displays “ice”.
e. The browser displays “tea”.
f. Compilation fails.

38. A context init parameter with name “ice” and value “tea” is properly declared. What is the result of this code? (1 correct answer)

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(getContextParameter("ice"));
}
```

a. A NullPointerException occurs at runtime.
b. The browser displays an empty page.
c. The browser displays “null”.
d. The browser displays “ice”.
e. The browser displays “tea”.
f. Compilation fails.

39. The Container creates a single instance for every servlet. The client requests are served with various threads. (1 correct answer)

a. true
b. false

40. Assume that the container is running on port 9999 of localhost, our application is called test1 and that the following servlet is declared with url pattern **/processor**. What is the output? (1 correct answer)

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getContextPath());
}
}
```

a. http://localhost:9999/test1/processor
b. /test1/processor
c. /processor
d. /test1

41. Assume that the container is running on port 9999 of localhost, our application is called test1 and that the following servlet is declared with url pattern **/processor**. What is the output? (1 correct answer)

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getServletPath());
}
}
```

a. http://localhost:9999/test1/processor
b. /test1/processor
c. /processor
d. /test1

42. Assume that the container is running on port 9999 of localhost, our application is called test1 and that the following servlet is declared with url pattern **/processor**. What is the output? (1 correct answer)

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getRequestURL());
}
}
```

a. http://localhost:9999/test1/processor
b. /test1/processor
c. /processor
d. /test1

43. Assume that the container is running on port 9999 of localhost, our application is called test1 and that the following servlet is declared with url pattern **/processor**. What is the output? (1 correct answer)

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getRequestURI());
}
}
```

a. http://localhost:9999/test1/processor
b. /test1/processor
c. /processor
d. /test1

44. Consider this form.

```
<form method="post" action="processor">
   <input type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getMethod());
}
}
```

What is the output when the form is submitted? (1 correct answer)

a. GET
b. POST
c. A HTTP 405 message informs that GET is not supported.
d. A HTTP 405 message informs that POST is not supported.

45. Consider this form.

```
<form method="post" action="processor">
   <input type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(request.getMethod());
}
public void doPost(HttpServletResponse response,
                  HttpServletRequest request)
                  throws ServletException, IOException {
    doGet(request, response);
}
}
```

What is the output when the form is submitted? (1 correct answer)

a. GET
b. POST
c. A HTTP 405 message informs that GET is not supported.
d. A HTTP 405 message informs that POST is not supported.

46. Consider this form.

```
<form name="form" method="post" action="processor">
   <input name="submit" type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doPost(HttpServletRequest request,
                   HttpServletResponse response)
                   throws ServletException, IOException {
    response.getWriter().format("%s %s",
    request.getParameter("form"),
    request.getParameter("name"));
}
}
```

What is the output when the form is submitted? (1 correct answer)

a. form OK
b. null OK
c. form null
d. null null

47. Consider this form.

```
<form name="form" method="post" action="processor">
   <input name="submit" type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doPost(HttpServletRequest request,
                   HttpServletResponse response)
                   throws ServletException, IOException {
```

```
    response.getWriter().format("%s %s",
    request.getParameter("form"),
    request.getParameter("submit"));
}
}
```

What is the output when the form is submitted? (1 correct answer)

a. form OK
b. null OK
c. form null
d. null null

48. Consider this form.

```
<form name="form" method="post" action="processor">
   <input type="text" name="name" value="OK"/>
   <input name="submit" type="submit" value="OK"/>
</form>
```

And this servlet.

```
public class Processor extends HttpServlet {
public void doPost(HttpServletRequest request,
                   HttpServletResponse response)
                   throws ServletException, IOException {
    response.getWriter().format("%s %s",
    request.getParameter("name"),
    request.getParameter("OK"));
}
}
```

What is the output when the form is submitted? (1 correct answer)

a. OK OK
b. null OK
c. OK null
d. null null

49. Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target")
       .forward(request, response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getRequestURL());
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. http://localhost:9999/exam/source
b. http://localhost:9999/exam/target
c. null

50. Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target")
       .forward(request, response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getQueryString());
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. user=nikos&pass=12345
b. null

51. Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target")
       .forward(request, response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getAttribute("javax.servlet.
forward.query_string"));
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. user=nikos&pass=12345
b. null

52. Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target")
       .forward(request, response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getAttribute("javax.servlet.
forward.servlet_path"));
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. /source
b. /target
c. null

53. Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target")
       .forward(request, response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getServletPath());
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. /source
b. /target
c. null

54. Consider doGet() of a valid servlet with url pattern /source,

```
request.getRequestDispatcher("target")
       .forward(request, response);
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getParameter("pass"));
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. 12345

b. null

55. Consider doGet() of a valid servlet with url pattern /source,

```
response.sendRedirect("target");
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getParameter("pass"));
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. 12345

b. null

56. Consider doGet() of a valid servlet with url pattern /source,

```
response.sendRedirect("target");
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(request.getServletPath());
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. /source

b. /target

c. null

57. Consider doGet() of a valid servlet with url pattern /source,

```
response.sendRedirect("target");
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(
    request.getAttribute("javax.servlet.forward.query_string")
);
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. user=nikos&pass=12345
b. null

58. Consider doGet() of a valid servlet with url pattern /source,

```
response.sendRedirect("target");
```

and doGet() of another valid servlet with url pattern /target.

```
response.getWriter().println(
    request.getQueryString()
);
```

What is the output when invoking http://localhost:9999/exam/source?user=nikos&pass=12345? (1 correct answer)

a. user=nikos&pass=12345
b. null

59. There is a top-level folder *help* in the war file with *index.html*inside. What is the output of this code? (1 correct answer)

```
request.getRequestDispatcher("help/index.html")
       .forward(request, response);
```

a. The content of index.html
b. HTTP Status 404

60. There is a top-level folder *help* in the war file with *index.html*inside. What is the output of this code? (1 correct answer)

```
request.getRequestDispatcher("/help/index.html")
       .forward(request, response);
```

a. The content of index.html
b. HTTP Status 404

61. What is the output of this code? (1 correct answer)

```
request.getRequestDispatcher("http://fcom.gr")
       .forward(request, response);
```

a. The content of fcom.gr
b. HTTP Status 404

62. There is a top-level folder *help* in the war file with *index.html*inside. What is the output of this code? (1 correct answer)

```
response.sendRedirect("help/index.html");
```

a. The content of index.html
b. HTTP Status 404

63. There is a top-level folder *help* in the war file with *index.html*inside. What is the output of this code? (1 correct answer)

```
response.sendRedirect("/help/index.html");
```

a. The content of index.html
b. HTTP Status 404

64. What is the output of this code? (1 correct answer)

```
response.sendRedirect("http://fcom.gr");
```

a. The content of fcom.gr
b. HTTP Status 404

65. What happens when this servlet is deployed and a user hits repeatedly the refresh button of his browser? (1 correct answer)

```
public class Test extends HttpServlet {
private Integer number = 0;
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    number = number + 1;
    response.getWriter().println(number);
}
}
```

a. The same value is always displayed.
b. The displayed value increases with every refresh.

66. What happens when this servlet is deployed and a user hits repeatedly the refresh button of his browser? (1 correct answer)

```
public class Test extends HttpServlet {
private Integer number = new Random().nextInt();
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    response.getWriter().println(number);
}
}
```

a. The same value is always displayed.
b. The displayed value increases with every refresh.

67. The following types are ALL interfaces and are used as listeners in a web application. (1 correct answer)
→ ServletContextListener
→ ServletContextAttributeListener
→ ServletRequestListener
→ ServletRequestAttributeListener
→ HttpSessionListener
→ HttpSessionBindingListener
→ HttpSessionAttributeListener
→ HttpSessionActivationListener

a. true
b. false

68. Consider the interface ServletContextListener. What is the signature of the method that is invoked when the servlet context is about to be shut down? (1 correct answer)

a. void contextDeleted(ServletContextEvent)
b. void contextDestroyed(ServletContextEvent)
c. void servletContextDeleted(ServletContextEvent)
d. void servletContextDestroyed(ServletContextEvent)

69. There is a properly declared HttpSessionAttributeListener. How many times its **attributeRemoved**method is invoked when the following servlet is accessed once? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                   HttpServletResponse response)
                   throws ServletException, IOException {
    request.getSession().setAttribute("planet", "earth");
    request.getSession().removeAttribute("planet");
    request.getSession().removeAttribute("planet");
}
}
```

a. 0
b. 1
c. 2

70. There is a properly declared HttpSessionAttributeListener. How many times its **attributeRemoved**method is invoked when the following servlet is accessed once? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                   HttpServletResponse response)
                   throws ServletException, IOException {
    request.getSession().setAttribute("planet", "earth");
    request.getSession().invalidate();
    request.getSession().removeAttribute("planet");
    request.getSession().setAttribute("planet", "earth");
}
}
```

a. 0
b. 1
c. 2

71. There is a properly declared HttpSessionAttributeListener. How many times its **attributeReplaced**method is invoked when the following servlet is accessed once? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                   HttpServletResponse response)
```

```
                  throws ServletException, IOException {
    request.getSession().setAttribute("planet", "earth");
    request.getSession().removeAttribute("earth");
    request.getSession().setAttribute("planet", "earth");
}
}
```

a. 0
b. 1
c. 2

72. There is a properly declared HttpSessionAttributeListener. How many times its **attributeReplaced**method is invoked when the following servlet is accessed once? (1 correct answer)

```
public class Test extends HttpServlet {
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
                  throws ServletException, IOException {
    request.getSession().setAttribute("planet", "earth");
    request.getSession().removeAttribute("planet");
    request.getSession().setAttribute("planet", "venus");
    request.setAttribute("planet", "earth");
    request.setAttribute("planet", "venus");
}
}
```

a. 0
b. 1
c. 2