WJP CCEE Practice Test

Total points 13/19

0 of 0 points

PRN: *

250240320013

Centre: * Dropdown

Kharghar

Name: *

Amol Gavit

Quizz 13 of 19 points

Explanation:
JDBC (Java Database Connectivity) architecture consists of several key components that facilitate communication between Java applications and databases. The primary components include:

JDBC Driver – A software component that enables Java applications to interact with a database.

Driver Manager – Manages a list of database drivers and establishes connections.

JDBC URL – A connection string that specifies the database location and access details.

✓ Which of the following is NOT a component of JDBC architecture? *

- A) JDBC Driver
- B) Driver Manager
- C) JDBC URL
- (●) D) HTTP Servlet

Breakdown of Each Option:
JDBC Driver – Correct.

The JDBC driver acts as a bridge between Java applications and databases, translating JDBC calls into database-specific operations.

Driver Manager – Correct.

The Driver Manager is responsible for managing database drivers and establishing connections.

JDBC URL – Correct.

The JDBC URL provides the necessary connection details for accessing a database.

HTTP Servlet – Incorrect.

Servlets handle HTTP requests and responses but are not part of JDBC architecture.

✗ Which of the statement is correct? *

- A - The JDBC module provides a JDBC-abstraction layer that removes the need to do tedious JDBC related coding.
- (●) B - The ORM module provides integration layers for popular object-relational mapping APIs, including JPA, JDO, Hibernate, and iBatis.
- C - The Java Messaging Service JMS module contains features for producing and consuming messages.
- D - All of the above.

Correct answer

- (●) D - All of the above.

Explanation:
Spring provides various modules to simplify development, including JDBC, ORM, and JMS. Each of these modules plays a crucial role in handling database operations, object-relational mapping, and messaging.

JDBC Module – Correct.
The JDBC module provides a JDBC-abstraction layer, reducing the need for manual JDBC coding.
It simplifies database interactions by handling connection management and error handling.

ORM Module – Correct.
The ORM module integrates with popular object-relational mapping (ORM) frameworks like JPA, JDO, Hibernate, and iBatis.
It helps developers work with databases using object-oriented principles instead of raw SQL.

JMS Module – Correct.
The JMS module provides features for producing and consuming messages in a messaging system.

It simplifies working with Java Message Service (JMS), enabling asynchronous communication.

All of the above – Correct.

Since all the individual statements are correct, the best answer is "All of the above."

✓ doGet(), doPost(),doHead, doDelete(), doTrace()... *
Choose What type of servlets use these methods ?

- ( ) Generic servlet
- (•) HttpServlets
- ( ) GlobalServlets
- ( ) None of the above

Explanation:
The methods doGet(), doPost(), doHead(), doDelete(), and doTrace() are specific to HTTP-based servlets. These methods handle different types of HTTP requests, making them part of the HttpServlet class.

Generic Servlet – Incorrect.

GenericServlet is a protocol-independent servlet class.

It does not provide built-in support for HTTP methods like doGet() and doPost().

HttpServlets – Correct.

HttpServlet is a subclass of GenericServlet designed specifically for handling HTTP requests.

It provides methods like doGet(), doPost(), doHead(), doDelete(), and doTrace() to process different types of HTTP requests.

GlobalServlets – Incorrect.

GlobalServlets is not a recognized servlet type in Java.

✓ Which method is used to execute an SQL statement that does not return a ResultSet, such as an INSERT, UPDATE, or DELETE statement?

- (•) A) executeUpdate()
- ( ) B) executeQuery()
- ( ) C) execute()
- ( ) D) executeCommand()

Explanation:
In JDBC, when executing SQL statements that modify data but do not return a ResultSet, the correct method to use is executeUpdate(). This method is specifically designed for INSERT, UPDATE, DELETE, and DDL statements (such as CREATE TABLE).

Breakdown of Each Option:
executeUpdate() – Correct.
This method executes SQL statements that modify data and returns an integer representing the number of affected rows.
Example usage:
int rowsAffected = statement.executeUpdate("UPDATE employees SET salary = 50000 WHERE id = 1");

executeQuery() – Incorrect.
This method is used for SELECT statements that return a ResultSet.
Example:
ResultSet rs = statement.executeQuery("SELECT * FROM employees");

execute() – Incorrect.
This method can execute any SQL statement, but it returns a boolean indicating whether the result is a ResultSet.
It is useful when the type of SQL statement is unknown.

executeCommand() – Incorrect.
This method does not exist in JDBC.

✓ What is the purpose of the web.xml file in a servlet-based application? *

- ( ) A) To define servlet classes
- ( ) B) To configure URL mappings for servlets
- ( ) C) To set initialization parameters
- (•) D) All of the above

Explanation:
The web.xml file, also known as the Deployment Descriptor, is a crucial configuration file in a Java servlet-based application. It defines various settings that the web container

Breakdown of Each Option:
To define servlet classes – Correct.
The web.xml file allows developers to specify servlet classes that should be managed by the web container.

To configure URL mappings for servlets – Correct.
The web.xml file maps servlet classes to specific URL patterns, enabling request routing.

To set initialization parameters – Correct.
The web.xml file allows setting initialization parameters for servlets, which can be accessed using getInitParameter().

✓ What is a servlet in Java? * 1/1

- ( ) A) A Java program that runs on the client side
- (•) B) A Java program that runs on the server side ✓
- ( ) C) A JavaScript function
- ( ) D) A type of database connection

Explanation:
A servlet is a Java program that runs on the server side and is used to handle HTTP requests in web applications. Servlets process client requests, generate dynamic responses, and interact with databases or other backend services.

Breakdown of Each Option:
A Java program that runs on the client side – Incorrect.
Java servlets do not run on the client side; they execute on the server and respond to client requests.

A Java program that runs on the server side – Correct.
Servlets are server-side components that handle web requests and generate responses dynamically.

A JavaScript function – Incorrect.
JavaScript functions run on the client-side within a web browser, whereas servlets operate on the server.

A type of database connection – Incorrect.
Servlets can interact with databases using JDBC, but they are not database connections themselves.

✓ Which JDBC class or interface is used to retrieve the results of a SQL query?

- (●) A) ResultSet
- ( ) B) Statement
- ( ) C) Connection
- ( ) D) SQLException

Explanation:
In JDBC, the ResultSet interface is used to retrieve and manipulate the results of a SQL query. When executing a SELECT statement, the database returns a ResultSet, which acts as a cursor pointing to the retrieved rows.

Breakdown of Each Option:
ResultSet – Correct.
The ResultSet interface represents the result set of a database query.
It provides methods like next(), getString(), and getInt() to access data from the retrieved rows.

Statement – Incorrect.
The Statement interface is used to execute SQL queries but does not store results.
It is responsible for sending SQL commands to the database.

Connection – Incorrect.
The Connection interface is used to establish a connection to the database but does not retrieve query results.

SQLException – Incorrect.
SQLException is an exception class used for handling database errors, not retrieving query results.

✓ What is the way with which we can use constructor for a servlet? *

- (●) Initialization and Constructor function
- ( ) Setup() method
- ( ) Constructor Function
- ( ) Initialization

Explanation:
In Java Servlets, constructors are not commonly used for initialization because servlets rely on the init() method for setup. The correct way to use a constructor in a servlet is through Initialization and Constructor function, as servlets require a ServletConfig object for proper initialization.

Initialization and Constructor function – Correct.
Servlets can have constructors, but they cannot access servlet-specific parameters in the constructor.
The init() method is the preferred way to initialize a servlet because it receives a ServletConfig object created by the container.

Setup() method – Incorrect.
There is no setup() method in the servlet lifecycle.
Initialization is handled by the init() method, not a separate setup function.

Constructor Function – Incorrect.
While servlets can have constructors, they cannot access servlet parameters in them.
The servlet container calls the no-argument constructor, but initialization happens in init().

Initialization – Incorrect.
This option is too vague and does not specify the correct mechanism for servlet initialization.

✓ Which HTTP method is used to request a resource without changing its state on the server?

- (●) A) GET
- ( ) B) POST
- ( ) C) PUT
- ( ) D) DELETE

Explanation:
The GET method is used to request a resource from the server without modifying its state. It is considered a safe and idempotent HTTP method, meaning multiple GET requests will not alter the server's data.

Breakdown of Each Option:
GET – Correct.
The GET method retrieves data from the server without making any changes.
It is commonly used for fetching web pages, API responses, and static resources.

POST – Incorrect.
The POST method is used to send data to the server, often creating or modifying resources.

PUT – Incorrect.
The PUT method is used to update or replace an existing resource on the server.

DELETE – Incorrect.
The DELETE method is used to remove a resource from the server.

✗ What is the purpose of the PreparedStatement in JDBC? *

- (●) A) To execute dynamic SQL queries
- ( ) B) To execute precompiled SQL queries
- ( ) C) To manage database transactions
- ( ) D) To retrieve metadata

Correct answer

- (●) B) To execute precompiled SQL queries

Explanation:
The PreparedStatement in JDBC is used to execute precompiled SQL queries, making database interactions more efficient and secure. Unlike a regular Statement, a PreparedStatement is compiled once and can be executed multiple times with different parameters.

Breakdown of Each Option:
To execute dynamic SQL queries – Incorrect.
While PreparedStatement allows parameterized queries, it is not primarily used for dynamic SQL execution.
Dynamic SQL is typically handled using Statement or CallableStatement.

To execute precompiled SQL queries – Correct.
PreparedStatement precompiles SQL queries, reducing execution time and improving performance.
It also helps prevent SQL injection attacks by treating user input as parameters rather than raw SQL.

To manage database transactions – Incorrect.
Transaction management is handled by Connection methods like setAutoCommit(false), commit(), and rollback().
PreparedStatement itself does not manage transactions.

To retrieve metadata – Incorrect.
Metadata retrieval is done using DatabaseMetaData or ResultSetMetaData, not PreparedStatement.

Which class is the superclass of all servlets? *

- A) javax.servlet.HttpServlet
- B) javax.servlet.GenericServlet
- C) java.lang.Object
- D) javax.servlet.Servlet

Explanation:
The GenericServlet class is the superclass of all servlets in Java. It provides a generic, protocol-independent implementation of the Servlet interface, allowing developers to create servlets that handle requests without being tied to a specific protocol like HTTP.

Breakdown of Each Option:
javax.servlet.HttpServlet – Incorrect.
HttpServlet is a subclass of GenericServlet that specifically handles HTTP requests.
It provides methods like doGet(), doPost(), and doDelete() for handling web-based interactions.

javax.servlet.GenericServlet – Correct.
GenericServlet is the direct superclass of HttpServlet and other servlet implementations.
It provides basic lifecycle methods like init(), service(), and destroy().

java.lang.Object – Incorrect.
While all Java classes ultimately extend Object, it is not the direct superclass of servlets.
Servlets extend GenericServlet, which in turn extends Object.

javax.servlet.Servlet – Incorrect.
Servlet is an interface, not a class.
GenericServlet implements Servlet, making it the actual superclass of all servlets.

Which method is called when a servlet is first loaded into memory? *

- A) service()
- B) init()
- C) doGet()
- D) doPost()

Explanation:
When a servlet is first loaded into memory, the init() method is called. This method is responsible for initializing the servlet before it starts handling client requests. It is executed only once during the servlet's lifecycle.

Breakdown of Each Option:
service() – Incorrect.
The service() method is called each time a request is received, but it is not responsible for initialization.
It determines the request type and delegates it to methods like doGet() or doPost().

init() – Correct.
The init() method is called once when the servlet is first loaded into memory.
It is used for resource allocation, such as setting up database connections or loading configuration settings.

doGet() – Incorrect.
The doGet() method is called only for HTTP GET requests, not during servlet initialization.

doPost() – Incorrect.
The doPost() method is called only for HTTP POST requests, not during servlet initialization.

Breakdown of Each Option:
To store Java class files – Incorrect.
The web.xml file does not store Java class files; instead, it configures servlet behavior.
Java class files are stored separately in the WEB-INF/classes directory.

To configure servlet mappings and other web application settings – Correct.
The web.xml file is primarily used to map servlets to specific URLs, define filters, and set initialization parameters.

To manage database connections – Incorrect.
Database connections are typically managed using JDBC, Spring, or Hibernate, not web.xml.
However, web.xml can store database configuration parameters, but it does not directly manage connections.

To define the structure of HTML pages – Incorrect.
The web.xml file does not define HTML page structures; it configures servlet behavior and request handling.

What is the primary purpose of the web.xml file in a Java web application?

- ○ A) To store Java class files
- ◉ B) To configure servlet mappings and other web application settings
- ○ C) To manage database connections
- ○ D) To define the structure of HTML pages

Explanation:
The web.xml file, also known as the Deployment Descriptor, is a crucial configuration file in a Java web application. It defines various settings that the web container needs to manage servlets and other components.

Breakdown of Each Option:
It is an object that provides information about the servlet configuration – Incorrect.
This describes ServletConfig, which provides configuration details specific to a single servlet.
ServletContext, on the other hand, is shared across all servlets in the application.

Which of the following is a correct statement about the servlet context? *

- ○ A) It is an object that provides information about the servlet configuration.
- ◉ B) It is an object that allows servlets to communicate with each other.
- ○ C) It is an object that provides methods for handling client requests.
- ○ D) It is an object that allows servlets to interact with the web server.

Explanation:
The ServletContext is an object that allows servlets to interact with the web server. It provides a way for servlets to access global application settings, share resources, and communicate with the servlet container.

It is an object that allows servlets to communicate with each other – Incorrect.
While ServletContext allows servlets to share global attributes, direct servlet-to-servlet communication is typically done using RequestDispatcher or forward/include mechanisms.

It is an object that provides methods for handling client requests – Incorrect.
Handling client requests is the responsibility of HttpServletRequest and HttpServletResponse, not ServletContext.

- ◉ D) It is an object that allows servlets to interact with the web server.

It is an object that allows servlets to interact with the web server – Correct.
ServletContext provides methods to retrieve server information, access resources, and log messages.

✗ Which tag in the web.xml file is used to map a servlet to a specific URL pattern? *0/1

- (●) A) <url-mapping> ✗
- ( ) B) <servlet-mapping>
- ( ) C) <servlet-url>
- ( ) D) <mapping-url>

Correct answer

- (●) B) <servlet-mapping>

Explanation:
In a Java web application, the <servlet-mapping> tag in the web.xml file is used to map a servlet to a specific URL pattern. This ensures that incoming requests are correctly routed to the appropriate servlet.

Breakdown of Each Option:
<url-mapping> – Incorrect.
This tag does not exist in standard servlet configuration.
The correct tag for mapping servlets is <servlet-mapping>.

<servlet-mapping> – Correct.
This tag is used to associate a servlet with a URL pattern, allowing the servlet to handle specific requests.

<servlet-url> – Incorrect.
This tag does not exist in servlet configuration.
URL patterns are defined within <servlet-mapping>.

<mapping-url> – Incorrect.
This tag does not exist in standard servlet configuration.
The correct approach is to use <servlet-mapping>.

✓ How can you set an HTTP response status code in a servlet? * 1/1

- A) response.sendError()
- B) response.setStatus() ✓
- C) response.sendRedirect()
- D) response.setError()

Explanation:
In a servlet, the HTTP response status code can be set using response.setStatus() or response.sendError(), depending on the scenario. The correct answer is response.setStatus(), which allows setting a custom status code without sending an error page.

Breakdown of Each Option:
response.sendError() – Incorrect.
This method sets the status code and sends an error page to the client.

response.setStatus() – Correct.
This method sets the HTTP status code without sending an error page.

response.sendRedirect() – Incorrect.
This method redirects the client to another URL but does not explicitly set a status code.

response.setError() – Incorrect.
This method does not exist in the HttpServletResponse class.

✗ What is the correct sequence of method calls in the lifecycle of a servlet? *0/1

- A) init(), doGet()/doPost(), service(), destroy()
- B) init(), service(), doGet()/doPost(), destroy()
- C) init(), doGet()/doPost(), destroy(), service() ✗
- D) init(), service(), destroy()

Correct answer

- D) init(), service(), destroy()

Explanation:
The correct sequence of method calls in the lifecycle of a servlet is init(), service(), destroy(). This sequence ensures that the servlet is properly initialized, processes client requests, and is eventually removed from memory.

Explanation:
The init() method in a servlet is called only once during its lifecycle. It is executed when the servlet is first loaded into memory by the servlet container, ensuring that initialization tasks are performed before handling client requests.

✓ How many times is the init() method called during the lifecycle of a servlet? *1/1

- (●) A) Once ✓
- ( ) B) Once for each client request
- ( ) C) Once for each HTTP method (GET, POST, etc.)
- ( ) D) It depends on the servlet configuration

Breakdown of Each Option:
Once – Correct.
The init() method is called only once when the servlet is first instantiated.
It is used for one-time setup tasks, such as establishing database connections or loading configuration settings.
Example:
java
```
public void init() throws ServletException {
   System.out.println("Servlet initialized!");
}
```

Once for each client request – Incorrect.
The init() method is not called for every request; instead, the service() method handles each request.

Once for each HTTP method (GET, POST, etc.) – Incorrect.
The init() method is not tied to HTTP methods; it is executed once when the servlet is loaded.

It depends on the servlet configuration – Incorrect.
While servlet loading behavior can be configured (e.g., eager loading using <load-on-startup> in web.xml), the init() method itself is always called only once2