# COP 4610L – Exam #2 – Fall 2007

**Name:**

**KEY**

November 6, 2007

100 points total

(Please print clearly)

## Multiple Choice Section (3 points each)

1. This interface enables any Java program to send SQL queries to any database, and receive back result tables with the desired data.

    a. PreparedStatement Interface

    b. RowSet Interface

    c. JDBC

    d. Servlet Interface

    e. XHTML

2. When a user attempts to connect to a MySQL database the first table which is accessed by MySQL as it attempts to verify the user is the

    a. User table.

    b. Password table.

    c. Databases table.

    d. Connection table.

    e. None of the above.

3. The `form` attribute which specifies the server-side form handler, i.e., the program that handles the request is named:

    a. get.

    b. post.

    c. action.

    d. localhost.

    e. doGet.

4. The `service` method (in the Servlet interface) of a servlet is invoked by
    a. The servlet's `init` method to allow it to respond to a request.
    b. The servlet container to allow it to respond to a request.
    c. Directly by the client to allow it to respond to a request.
    d. Any of the above.
    e. None of the above.

5. The MySQL command to show the privileges that have been granted to a specific user is the SHOW GRANTs command.  The basic syntax of this command is:
    a. SHOW GRANTS ON <database> FOR <username@hostname>;
    b. SHOW GRANTS FOR <username@hostname>;
    c. SHOW GRANTS ON <username@hostname>;
    d. Any of the above are correct.
    e. None of the above are correct.

6. To remove certain privileges granted to a user in MySQL the REVOKE command is utilized. Assuming that the administrator would like to revoke the SELECT privileges on the attribute <score> in the table <finalexam> in the <grades> database from the user <student>, the correct form of this command is:
    a. REVOKE SELECT  <score> FROM <student> ON <grades.finalexam>;
    b. REVOKE SELECT <grades.finalexam.score> FROM <student>;
    c. REVOKE SELECT (<score>) FROM <grades.finalexam> ON <student>;
    d. REVOKE SELECT (<score>) ON <grades.finalexam> FROM <student>;
    e. None of the above are correct.

7. The two most common HTTP requests (also known as methods) are:
    a. GET and PUT.
    b. GET and ACTION.
    c. GET and POST.
    d. GET and RESPONSE.
    e. GET and CONNECT.

8. A servlet's lifecycle begins when
   a. The user invokes it through a URL.
   b. The servlet container loads it into memory.
   c. The `init` method of the servlet is invoked.
   d. The servlet is darn good and ready.
   e. None of the above.

9. HTTP does not support persistent information that could help a web server determine that a request is from a particular client. For this reason HTTP is known as a
   a. Stateless protocol.
   b. Sessionless protocol.
   c. Transparent protocol.
   d. Connectionless protocol.
   e. None of the above.

10. The following SQL statement is syntactically valid:

    SELECT DISTINCT <attribute1>, MAX(<attribute 2>)
    FROM <table>
    GROUP BY <attribute 1>
    HAVING MAX(<attribute 2>) > 10;
    f. TRUE
    g. FALSE

11. Servlets are typically used on the client side of a networking application.
    h. TRUE
    i. FALSE

12. Database tables are manipulated in Java as:
    j. JDBC objects
    k. ODBC objects
    l. ResultSet objects
    m. Connection objects

3

13. Given the database schema shown in Problem #20, the following SQL statement is syntactically valid:

    SELECT *

    FROM suppliers

    WHERE suppliers.s# = (SELECT *

    FROM shipments

    WHERE shipments.p# = "P5");

    a. TRUE
    b. FALSE

14. In the Internet reference model, the transport layer is above the network layer.
    a. TRUE
    b. FALSE

15. This Java object is used to submit a query to a database.
    a. Connection
    b. Statement
    c. QueryObject
    d. ResultSet

16. This Java interface, which extends the Statement interface, is used to execute a precompiled SQL statement with or without IN parameters:
    a. ResultSetTableModel
    b. jdbcRowSetTest
    c. PreparedStatement
    d. RowSet

4

**Short Answer Section (points as shown)**

17.  (10 points) Write the required portion of the web.xml file for a web application which contains a servlet named exam2 that uses the Exam2.class file.  Ignore the web-app tag in the file.

```
<servlet>
     <servlet-name> exam2 </servlet-name>
     <servlet-class> Exam2 </servlet-class>
</servlet>
<servlet-mapping>
     <servlet-name> exam2 </servlet-name>
     <url-pattern> /exam2 </url-pattern>
</servlet-mapping>
```

18. (10 points) Show the required Tomcat directory structure for a Java servlet web application.

   List each required sub-directory and indicate the required contents of each.

   ---**webapps**[folder]

   ---------**webapp_name** [folder] (e.g., cop4610)

   ---------------- html "driver" files

   ---------------- **WEB-INF**[folder]

   ----------------  **web.xml** (servlet deployment file)

   ---------------------- lib [folder] (contains a copy of mysql-connector.jar)

   -----------------------**classes** [folder]

   ----------------------------- java servlet class files

19. (10 points) Using SQL DDL commands create a table with the following specifications:

Table name: accounts
Attribute 1: integer field, labeled: account_number, do not allow null values
Attribute 2: text field, max 30 characters, labeled, account_name, do not allow null values
Attribute 3: integer field, labeled: amount, set default value of 0
Attribute 4: date field, labeled: opened, do not allow null values
Attribute 5: number field: 8 max, 2 decimals, labeled: base_amount, default 0
The primary key is account_number
Account_name is a foreign key from a table called customers.

CREATE TABLE **ACCOUNTS** (

    ACCOUNT_NUMBER         INTEGER         NOT NULL   UNIQUE,

    ACCOUNT_NAME           VARCHAR(30)    NOT NULL,

    AMOUNT                 INTEGER         DEFAULT 0   NOT NULL,

    OPENED                 DATE            NOT NULL,

    BASE_AMOUNT           NUMBER(8,2)    DEFAULT 0.00 NOT NULL,

    PRIMARY KEY ( ACCOUNT_NUMBER),

    FOREIGN KEY (ACCOUNT_NAME) REFERENCES CUSTOMERS

);

20. (10 points) Write a correct SQL expression that will produce the correct results to the following query assuming the existence of the tables shown below.

The tables in the database

**suppliers (s#, name, status, city)**

**parts (p#, name, weight, color, city)**

**jobs (j#, name, numworkers, city)**

**shipments (s#, p#, j#, qty)**

The query

List the names of those suppliers who ship any black part to any job.

```
SELECT name
FROM suppliers
WHERE s# IN (SELECT s#
               FROM shipments
               WHERE p# IN (SELECT p#
                               FROM parts
                               WHERE color = "black"
                               )
            );
```

or –
```
SELECT name
FROM suppliers NATURAL JOIN shipments
WHERE p# IN (SELECT p#
               FROM parts
               WHERE color = "black"
               );
```

21. (10 points) Explain the basic servlet architecture that implements the request-response model between clients and servers using the HTTP protocol.

- A client application sends an HTTP request to the server.
- The servlet container receives the request and directs it to be processed by the appropriate servlet.
- The servlet does its processing, which may include interacting with a database or other server-side components, such as other servlets or JSPs.
- The servlet returns its results to the client – normally in the form of an HTML, XHTML, or XML document to display in a browser.

```
                                                        ┌───────────────────────────────────────┐
                                                        │              Servlet                   │
                                                        │             Container                  │
         HTTP request              HTTP request         │                                        │
 ┌──────────┐    ───────>  ┌──────────┐  ───────>       │  ┌─────────┐          ┌─────────┐       │
 │   Web    │              │   Web    │                 │  │ Servlet │  • • •    │ Servlet │       │
 │ Browser  │   <───────   │  Server  │   <───────      │  └─────────┘          └─────────┘       │
 └──────────┘              └──────────┘    HTTP         │       │                                │
       HTTP                               response       │       │                                │
     response                                           └───────┼────────────────────────────────┘
                                                                │
                                                         ┌──────┴──────┐
                                                         │  Database   │
                                                         └─────────────┘
```