ITMD 510 Final Study Sample (**solutions**)

1. What SQL operator can be used to perform a search for a substring?

a. STR b. SUB c. WHERE **d. LIKE**

2. What must you have installed on a system before you can use JDBC to access a database on the system?

a. Java b. A DBMS **c. Both A and B** d. Neither A nor B

3. What term refers to data that describes other data?

**a. meta data** b. abstract data c. micro data d. pseudo-data

4. Which of the following is NOT a part of JDBC URL?

a. <protocol> b. <subprotocol> c. <subname> d. **<tablename>**

5. \_\_\_\_\_\_\_\_\_package contains classes that help in connecting to a database

a. sql.java b. mysql.java **c. java.sql** d. java.mysql

6. Two main interfaces that directly use the Collection interface are \_\_ and \_\_.

a. AbstractList, LinkedList **b. Set,List** c. List,Vector d. HashSet,LinkedHashSet

7. addButton.setOnAction((ActionEvent e) {

data.add(new Person("Z","X"));

}

The function above can be rewritten how?

a. addButton.setOnAction((ActionEvent e()) -> { data.add(new Person("Z","X")); });

b. addButton.setOnAction((ActionEvent e() -> { data.add(new Person("Z","X"))); });

**c. addButton.setOnAction((ActionEvent e) -> { data.add(new Person("Z","X")); });**

d. addButton.setOnAction((ActionEvent e -> { data.add(new Person("Z","X")); });

8. Which is not a solid example of encapsulation?

a. A Car class having a has-a relation with class Parts **b. Taking for granted a wikipedia definition**

c. A washing machine and its use of a power Button d. A Touring machine

9. Given the generic following method, what can be passed in as a parameter value?

public static <E extends Number>

void displayArray(E[] array) {

for (E element : array)

System.out.println(element);

}

a. an array whose element type is E b. an array whose element type is Object

**c. an array whose element type is Integer** d.an array whose type is any superclass of Number

10. Variable **result** below has what action performed on it?

newMain(){

Integer i = new Integer(-8);

int **result** = newValue(i);

}

public static int newValue(int i) {

return (i < 0) ? -i : i;

}

a. boxing b. autoboxing c. casting **d. unboxing**

11. Write a create table statement below to create the tickets table (my\_Ticket) for your project given the information listed below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Field Type | Field Size | Primary Key? | Nulll? |
| id | Auto\_increment |  | Y | N |
| ticketNum | Integer |  | N | Y |
| ticketDesc | varchar | 100 | N | Y |

**String sql = "CREATE TABLE my\_Ticket " +**

**"(id INTEGER not NULL AUTO\_INCREMENT, " +**

**" ticketNum INTEGER, " +**

**" ticketDesc VARCHAR(100), " +**

**" PRIMARY KEY ( id ))";**

12. Create an insert statement below to insert a record into the above table that contains the data below.

|  |  |
| --- | --- |
| Variable (Field) name | Data |
| id |  |
| ticketNum | 1001 |
| ticketDesc | PC Virus |

**String sql = "INSERT INTO my\_Ticket(ticketNum, ticketDesc) " +**

**"VALUES ('"+ticket\_num+"','"+ticket\_desc+"')";**

13. Write a statement that will update all ticketNum fields > 200 and whose

ticketDesc field has the containment ‘Bootstrapped virus found’ to a ticketDesc value of ‘Bootsector virus found’ from the table above.

**String sql = "update my\_Ticket set ticketDesc = ‘Bootsector virus found’**

**where ticketDesc like ‘Bootstrapped virus found’ and**

**ticketNum > 200”;**

14. Assume a file called data.dat is opened for reading and contains the following record data. Assume records begin with a number:

**1 Puppy 2 Catnap 3 plaintiff 4 CoolCat 5 Cat and mouse 6 Catburgler.**

Fill in the underline to complete the statement for variable REGEX below to allow for the displaying any data containing the letters cat.

while((line = bufferedReader.readLine()) != null) {

String REGEX = "**cat+**";

Pattern p =

Pattern.*compile*(REGEX,Pattern.*CASE\_INSENSITIVE*);

String phrase = line.substring(2);

Matcher m = p.matcher(phrase);

while(m.find()) {

System.*out*.println(phrase);

}

}

15. Given a button named **btnCancel**, in FX an alert can verify if the cancel button has been pressed how?

**a. Optional<ButtonType> result = alert.showAndWait();**

**if (result.get() == btnCancel){ … }**

b. Optional<Button> result = alert.showAndWait();

if (result.get() == btnCancel){ … }

c. Optional result = alert.showAndWait();

if (result.get() == btnCancel){ … }

d. Optional<ButtonType> result = alert.showAndWait();

if (result == btnCancel){ … }