Reg.No | | | | | | | |

SNS COLLEGE OF TECHNOLOGY

A

(Autonomous)

MCA- Internal Assessment –III (August 2023)

Academic Year 2022-2023(Even) / Second Semester

19CAT608 – Java Programming – Answer key

Time: 1 1/2 Hours **Maximum Marks: 50**

Answer All Questions

PART - A (5 x 2 = 10 Marks)

		CO	BL
1	**Can you define multiple drivers used in JDBC.** Yes: The first parameter of getConnection() is a URL that will uniquely identify the driver to use for that connection.	CO4	Und
2	**Is it possible to connect multiple databases? Write a single statement to update or extract data from two or three databases.** Multiple Databases on One Server Instance It is possible to use SQL to write one query that combines the information from many databases. There are two requirements: You must prefix all table references with the database name	CO4	Ana

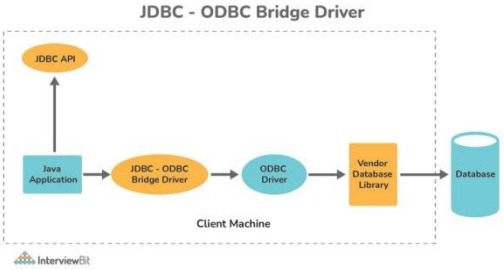
3	**Write the syntax of doGet and doPost method.**
	`doGet(HttpServletRequest request, HttpServletResponse response)`
	`doPost(HttpServletRequest request, HttpServletResponse response)`
		CO4	Rem

4	**Differentiate Marshalling and Unmarshalling.** Marshalling is the process of transforming the memory representation of an object to a data format suitable for the storage and transmission. Unmarshalling refers to the process of transforming a representation of an object that is used for storage or transmission to a representation of the object that is executable.	CO5	Ana
5	Analysis how the Components are differ from AWT and Swing. **AWT provides comparatively less powerful components. Swing provides more powerful components**. AWT is a thin layer of code on top of the operating system. Swing is much larger swing also has very much richer functionality.	CO5	Ana

PART - B (2x13=26, 1x14=14Marks)

6 (a) Examine different Java drivers available in Java and evaluate the best one for connectivity. CO4 Eva

Type I: JDBC - ODBC bridge driver
Type II: Native API – Partially Java Driver
Type III: Network Protocol - Fully Java Driver
Type IV: Thin Driver - Fully Java Driver



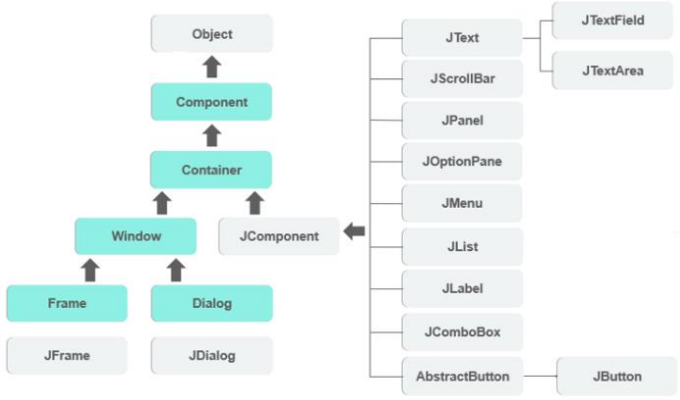
(Or)

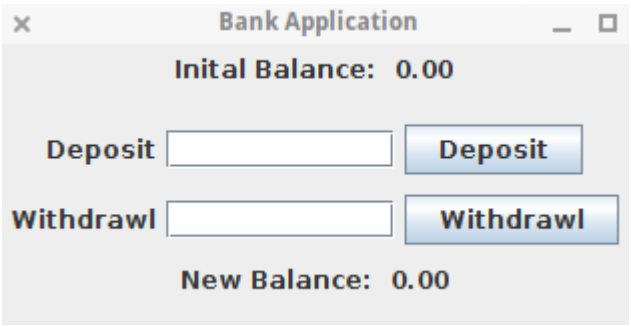
(b) Develop a Java program for implementation of JDBC to connect and manipulate data from MSaccess. CO4 Cre

```
import java.sql.*;
public class MsAcessODBC {
   public static void main(String[] args) {
      try {
```

```
            // loading thejdbc odbc driver
            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
            // creating connection toth data base
            Connection con = DriverManager.getConnection("jdbc:odbc:mydsn", "",
"");
            Statement st = con.createStatement();
            // create an execute sql command on database
            ResultSet rs = st.executeQuery("Select * from student order by rollno
asc");
            ResultSetMetaData rsmd = rs.getMetaData();
            // this getColumnCount reurn the number of column in the selected table
            int numberOfColumns = rsmd.getColumnCount();
            // while loop and with while loop code use for print the data
            while (rs.next()) {
               for (int i = 1; i <= numberOfColumns; i++) {
                  if (i > 1)
                     System.out.print(", ");
                  String columnValue = rs.getString(i);
                  System.out.print(columnValue);
               }
               System.out.println("");
            }
            st.close();
            con.close();
         } catch (Exception ex) {
            System.err.print("Exception: ");
            System.err.println(ex.getMessage());
         }
      }
```

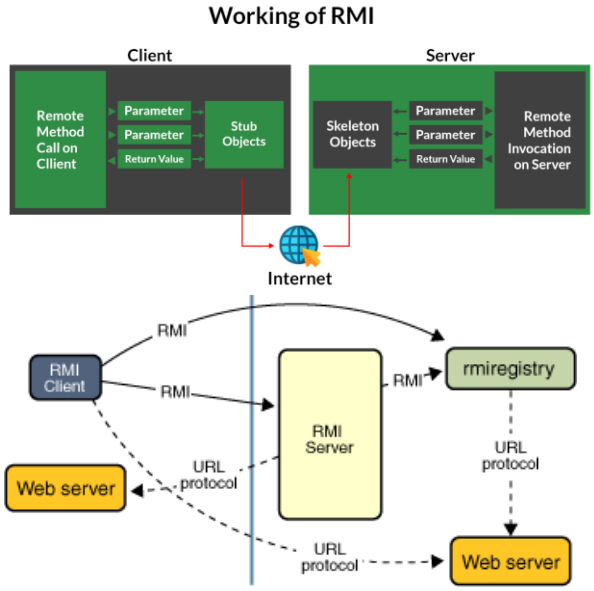
7 (a) Exemplify commonly used Swing components and design the user interface to get customer details to open the Bank account. CO5 Cre





(Or)

(b) Elevate RMI mechanism used in Java Program. CO5 Eva



8 (a) Write a HTTP servlet program to authenticate a user and retrieve all information from a HTML registration Form and store in a given data base.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class GfgRegister extends HttpServlet {

   protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();

    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String pass = request.getParameter("pass");

    try {
```

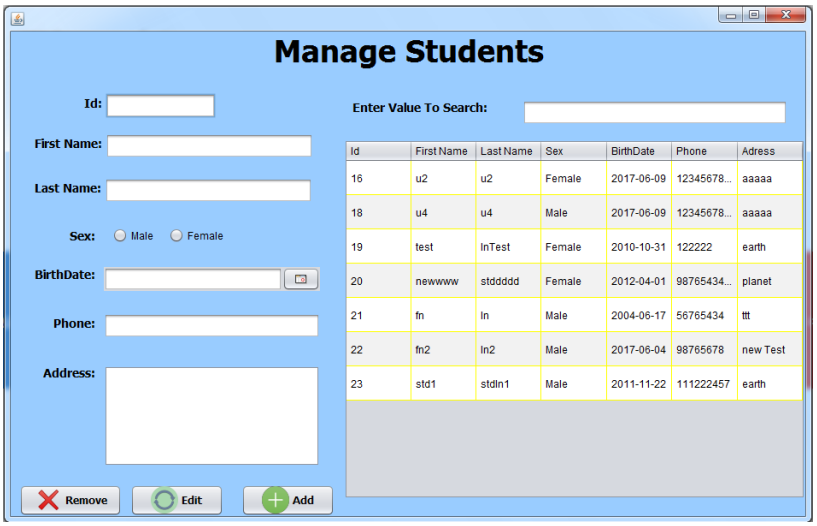
CO4 App

```
      // loading drivers for mysql
      Class.forName("com.mysql.jdbc.Driver");

      //creating connection with the database
      Connection con = DriverManager.getConnection
            ("jdbc:mysql://localhost:3306/geeksforgeeks","root","root");

      PreparedStatement ps = con.prepareStatement
            ("insert into gfglogin values(?,?,?)");

      ps.setString(1, name);
      ps.setString(2, email);
      ps.setString(3, pass);
      int i = ps.executeUpdate();

      if(i > 0) {
        out.println("You are successfully registered at geeksforgeeks");
      }
```

```
        }
        catch(Exception se) {
           se.printStackTrace();
        }

    }
}
```

(Or)

(b) Develop a Java program to design the Students Management system using swing. CO5 Cre



Reg.No | | | | | | | |

SNS COLLEGE OF TECHNOLOGY

(Autonomous)

B

MCA- Internal Assessment –III (August 2023)

Academic Year 2022-2023(Even) / Second Semester

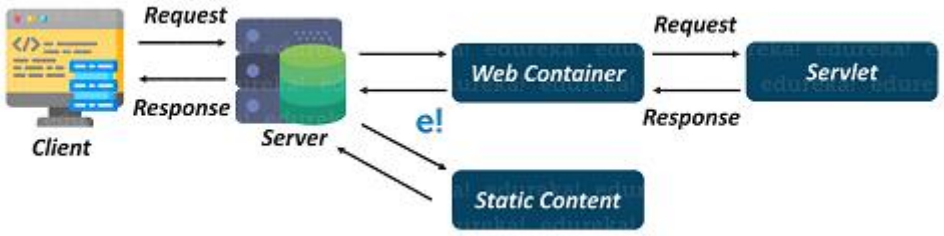
19CAT608 – Java Programming – Answer Key

Time: 1 1/2 Hours **Maximum Marks: 50**

Answer All Questions

PART - A (5 x 2 = 10 Marks)

1	Evaluate the following syntax and write the correct one. String SQL = "Update item limit = ? WHERE itemType = ?"; PreparedStatement ps =conn.prepareStatement(SQL); ResultSet rs = ps.execute(); **Syntax:** String SQL = "Update item SET limit = ? WHERE itemType = ?"; PreparedStatement ps = conn.prepareStatement(SQL); ResultSet rs = ps.executeQuery();	CO4	Eva
2	Define Generic and HttpServlet. 1. GenericServiet is not dependent on any particular protocol. It can be used with any protocol such as HTTP, SMTP, FTP, and so on. 2. HttpServlet is a dependent protocol and is only used with HTTP protocol. All methods are concrete except the service() method. service() method is an abstract method.	CO4	Rem
3	Differentiate executeQuery() and executeUpdate(). **executeQuery() command used for getting the data from database whereas executeUpdate() command used for insert,update,delete or execute() command used forany kind of operations**.	CO4	App
4	Trace the main task of stubs and skeletons in RMI. Stub − **A stub is a representation (proxy) of the remote object at client**. It resides in the client system; it acts as a gateway for the client program. Skeleton − This is the object which resides on the server side. stub communicates with this skeleton to pass request to the remote object.	CO5	Ana
5	Why doesn't a Servlet include main()? How does it work? Servlets don't have a **main() method.** Because servlets are executed using **web containers.** When a client places request for a servlet, then the server hands the requests to the web container where the servlet is deployed.	CO5	Eva



PART - B (2x13=26, 1x14=14 Marks)

6	(a)	Narrate the steps used to establish JDBC Connective. • Import JDBC Packages: Add import statements to your Java program to import required classes in your Java code. • Register JDBC Driver: In this step, JVM to loads the desired driver implementation into memory so that it can fulfill the JDBC requests. There are 2 approaches to register a driver. • The most suitable approach to register a driver is to use	CO4	App

Java's forName() method to dynamically load the driver's class file into memory, *which automatically registers it.* This method is suitable as it allows you to make the driver registration configurable and portable.

```
try {Class.forName("oracle.jdbc.driver.OracleDriver");
}
catch(ClassNotFoundException ex)
   System.out.println("Error: unable to load driver class!");
       System.exit(1);
```

The second approach you can use to register a driver is to use the static **registerDriver()**method.

```
try {Driver myDriver = new oracle.jdbc.driver.OracleDriver();
      DriverManager.registerDriver( myDriver );
}catch(ClassNotFoundException ex){
      System.out.println("Error: unable to load driver class!");
      System.exit(1);
}
```

You should use the *registerDriver()* method if you are using a non-JDK compliant JVM, such as the one provided by Microsoft. Here each form requires a database **URL**.

Database URL Formulation: URL Formulation is necessary to create a properly formatted address that points to the database to which you want to connect. Once you loaded the driver, you can establish a connection using the **DriverManager.getConnection()** method.

DriverManager.getConnection() methods are−

getConnection(String url)

getConnection(String url, Properties prop)

getConnection(String url, String user, String password)

Create a connection object

You can create a connection using the database URL, username, and password and also using properties object.

Close

Finally, to end the database session, you need to close all the database connections. However, if you forget, Java's garbage collector will close the connection when it cleans up stale objects.

```
1conn.close();
```

(Or)

(b) Illustrate the interaction between HttpRequest and HttpResponse with simple Java Program. CO4 App

```
mport java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;

public class GfgRegister extends HttpServlet {

  protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
    response.setContentType("text/html;charset=UTF-8");
    PrintWriter out = response.getWriter();

    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String pass = request.getParameter("pass");

    try {
```

```
        // loading drivers for mysql
        Class.forName("com.mysql.jdbc.Driver");

        //creating connection with the database
        Connection con = DriverManager.getConnection
                ("jdbc:mysql://localhost:3306/geeksforgeeks","root","root");

        PreparedStatement ps = con.prepareStatement
                ("insert into gfglogin values(?,?,?)");

        ps.setString(1, name);
        ps.setString(2, email);
        ps.setString(3, pass);
        int i = ps.executeUpdate();

        if(i > 0) {
           out.println("You are successfully registered at geeksforgeeks");
        }

      }
      catch(Exception se) {
         se.printStackTrace();
      }

   }
}
```

7 (a) Explain the steps involved in creating a Java Bean with an example. CO5 App

- Create a class in a package as the bean class.
- Provide the required variables as the properties.
- Provide setter and getter methods for each of the variables.
- Store the package folder inside a classes folder.
- Compile the file as ordinary java file.

```
1  package javaskool;
2  import java.io.*;
3  import java.util.*;
4  import java.sql.*;
5
6  public class Customer implements Serializable{
7
8          private String custID;
9          private String custName;
10         private int qty;
11         private float price;
12         private float total;
13         private int storeCust;
14
15         public String getCustID() {
16             return custID;
17         }
18
19         public void setCustID(String custID) {
20             this.custID = custID;
21         }
22
23         public String getCustName() {
24             return custName;
25         }
26
27         public void setCustName(String custName) {
28             this.custName = custName;
29         }
30
31         public int getQty() {
32             return qty;
33         }
```

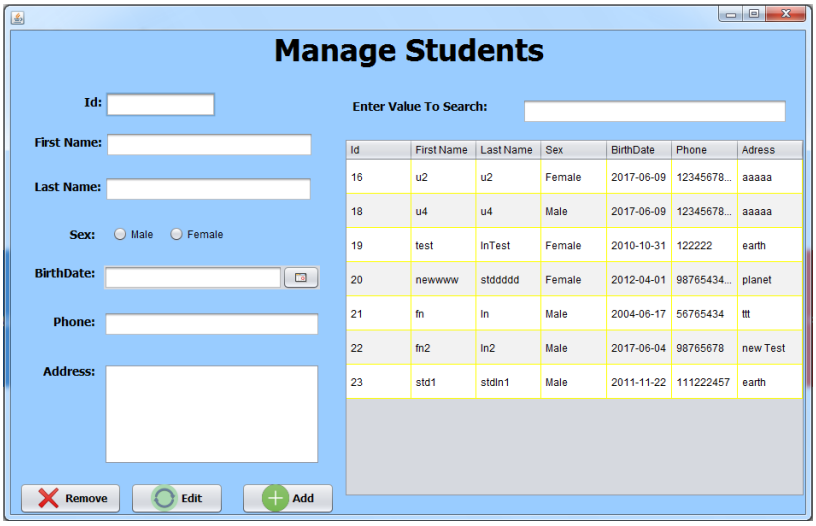
Click Here to download the Complete Code

Getter or Accessor

Setter or Mutator

(Or)

(b) Demonstrate the typical GUI program for any application using Swing concept in Java. CO5 App



8 (a) Develop a Java program for Employee information system using JDBC with mysql.

```
import java.sql.*;
public class MsAcessODBC {
   public static void main(String[] args) {
      try {
         // loading thejdbc odbc driver
         Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
         // creating connection toth data base
         Connection con = DriverManager.getConnection("jdbc:odbc:mydsn", "", "");
         Statement st = con.createStatement();
         // create an execute sql command on database
         ResultSet rs = st.executeQuery("Select * from student order by rollno asc");
         ResultSetMetaData rsmd = rs.getMetaData();
         // this getColumnCount reurn the number of column in the selected table
         int numberOfColumns = rsmd.getColumnCount();
         // while loop and with while loop code use for print the data
         while (rs.next()) {
            for (int i = 1; i <= numberOfColumns; i++) {
               if (i > 1)
                  System.out.print(", ");
               String columnValue = rs.getString(i);
               System.out.print(columnValue);
            }
            System.out.println("");
         }
         st.close();
         con.close();
      } catch (Exception ex) {
         System.err.print("Exception: ");
         System.err.println(ex.getMessage());
      }
   }
```

CO 4 Cre

(Or)

(b) Create a simple RMI Application to access the remote methods using client object. CO 5 Cre

```
1. import java.rmi.*;
2. public interface Adder extends Remote{
3. public int add(int x,int y)throws RemoteException;
4. }
```

```
1. import java.rmi.*;
2. import java.rmi.server.*;
```

```
3. public class AdderRemote extends UnicastRemoteObject implements Adder{
4. AdderRemote()throws RemoteException{
5. super();
6. }
7. public int add(int x,int y){return x+y;}
8. }
```

```
1. import java.rmi.*;
2. import java.rmi.registry.*;
3. public class MyServer{
4. public static void main(String args[]){
5. try{
6. Adder stub=new AdderRemote();
7. Naming.rebind("rmi://localhost:5000/sonoo",stub);
8. }catch(Exception e){System.out.println(e);}
9. }
10. }
```

```
1. import java.rmi.*;
2. public class MyClient{
3. public static void main(String args[]){
4. try{
5. Adder stub=(Adder)Naming.lookup("rmi://localhost:5000/sonoo");
6. System.out.println(stub.add(34,4));
7. }catch(Exception e){}
8. }
9. }
```

For running this rmi example,

1) compile all the java files

javac *.java

2)create stub and skeleton object by rmic tool

rmic AdderRemote

3)start rmi registry in one command prompt

rmiregistry 5000

4)start the server in another command prompt

java MyServer

5)start the client application in another command prompt

java MyClient