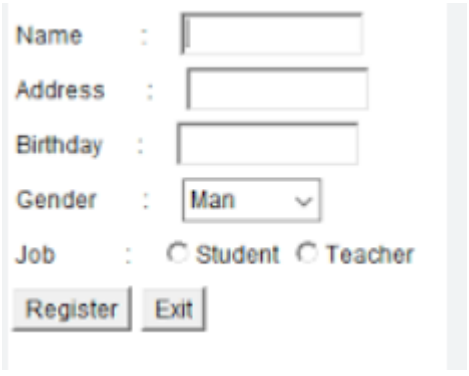
SVKM'S
Mithibai College of Arts, Chauhan Institute of Science &
Amrutben Jivanlal College of Commerce and Economics (Autonomous)
Academic Year (2022-23)
Class: SYBSC Semester: IV
Program: B.Sc Computer Science Max. Marks: 75
Course Name: Advanced Java Time:
Course Code: USMACS402 Duration: 2 hrs 30 minutes
Date:

SOLUTION SET

Q1	ATTEMPT ANY 3 FROM THE FOLLOWING:	[21]
A	What is Swing? Discuss the features of Swing in detail. Swing 2m +5 exploring features of Swing 5m Java Swing provides platform-independent and lightweight components. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc. Java swing components are platform-independent. Swing components are lightweight. Swing supports pluggable look and feel. Swing follows MVC	7
B	Write a Swing Code to design the following:  public void init() {	7

```
        setLayout(new FlowLayout(FlowLayout.LEFT));

        add(new Label("Name         :"));
        add(new TextField(10));

        add(new Label("Address      :"));
        add(new TextField(10));

        add(new Label("Birthday     :"));
        add(new TextField(10));


        add(new Label("Gender       :"));
        Choice gender = new Choice();
        gender.addItem("Man");
        gender.addItem("Woman");
        Component add = add(gender);

        add(new Label("Job          :"));
        CheckboxGroup job = new CheckboxGroup();
        add(new Checkbox("Student", job, false));
        add(new Checkbox("Teacher", job, false));

        add(new Button("Register"));
        add(new Button("Exit"));
        }
```

C | Write a note on BLOB and CLOB. | 7

BLOB 3.5m and CLOB 3.5m.

A BLOB is binary large object that can hold a variable amount of data with a maximum length of 65535 characters.

These are used to store large amounts of binary data, such as images or other types of files. Fields defined as TEXT also hold large amounts of data. The difference between the two is that the sorts and comparisons on the stored data are case sensitive on BLOBs and are not case sensitive in TEXT fields. You do not specify a length with BLOB or TEXT.

Storing blob in to database

To store Blob datatype to database, using JDBC program follow the steps given below

Step 1: Connect to the database

You can connect to a database using the getConnection() method of the DriverManager class.

Connect to the MySQL database by passing the MySQL URL which is jdbc:mysql://localhost/sampleDB (where sampleDB is the database name), username and password as parameters to the getConnection() method.

```
String mysqlUrl = "jdbc:mysql://localhost/sampleDB";
Connection con = DriverManager.getConnection(mysqlUrl, "root", "password");
```

Step 2: Create a Prepared statement

Create a PreparedStatement object using the prepareStatement() method of the Connection interface. To this method pass the insert query (with place holders) as a parameter.

```
PreparedStatement pstmt = con.prepareStatement("INSERT INTO MyTableVALUES(?, ?)");
```

Step 3: Set values to the place holders

Set the values to the place holders using the setter methods of the PreparedStatement interface. Chose the methods according to the datatype of the column. For Example if the column is of VARCHAR type use setString() method and if it is of INT type you can use setInt() method.

And if it is of Blob type you can set value to it using the setBinaryStream() or setBlob() methods. To these methods pass an integer variable representing the parameter index and an object of InputStream class as parameters.

```
pstmt.setString(1, "sample image");
//Inserting Blob type
InputStream in = new FileInputStream("E:\images\cat.jpg");
pstmt.setBlob(2, in);
```

Step 4: Execute the statement

Execute the above created PreparedStatement object using the execute() method of the PreparedStatement interface.

Retrieving blob from database

The getBlob() method of the ResultSet interface accepts an integer representing the index of the column (or, a String value representing the name of the column) and retrieves the value at the specified column and returns it in the form of a Blob object.

```
while(rs.next()) {
  rs.getString("Name");
  rs.getString("Type");
  Blob blob = rs.getBlob("Logo");
```

}

The getBytes() method of the Blob Interface retrieves the contents of the current Blob object and returns as a byte array.

Using the getBlob() method you can get the contents of the blob in to a byte array and create an image using the write() method of the FileOutputStream object.

```
byte byteArray[] = blob.getBytes(1,(int)blob.length());
FileOutputStream outPutStream = new FileOutputStream("path");
outPutStream.write(byteArray);
```

CLOB stands for Character Large Object in general, an SQL Clob is a built-in datatype and is used to store large amount of textual data. Using this datatype, you can store data up to 2,147,483,647 characters.

The java.sql.Clob interface of the JDBC API represents the CLOB datatype. Since the Clob object in JDBC is implemented using an SQL locator, it holds a logical pointer to the SQL CLOB (not the data).

MYSQL database provides support for this datatype using four variables.

TINYTEXT: A CLOB type with a maximum of 28-1 (255) characters.

TEXT: A CLOB type with a maximum of 216-1 (65535) characters.

MEDIUMTEXT: A CLOB type with a maximum of 224-1 (16777215) characters.

LONGTEXT: A CLOB type with a maximum of 232-1 (4294967295) characters.

Storing Clob datatype in to table in a database

To store Clob datatype to database, using JDBC program follow the steps given below

Step 1: Connect to the database

You can connect to a database using the getConnection() method of the DriverManager class.

Connect to the MySQL database by passing the MySQL URL which is jdbc:mysql://localhost/sampleDB (where sampleDB is the database name), username and password as parameters to the getConnection() method.

```
String mysqlUrl = "jdbc:mysql://localhost/sampleDB";
Connection con = DriverManager.getConnection(mysqlUrl, "root", "password");
```

Step 2: Create a Prepared statement

Create a PreparedStatement object using the prepareStatement() method of the Connection interface. To this method pass the insert query (with place holders) as a parameter.

```
PreparedStatement pstmt = con.prepareStatement("INSERT INTO Technologies(Name,
Type, Article ) VALUES (?, ?, ?)");
```

Step 3: Set values to the place holders

Set the values to the place holders using the setter methods of the PreparedStatement interface. Chose the methods according to the datatype of the column. For Example if the column is of VARCHAR type use setString() method and if it is of INT type you can use setInt() method.

And if it is of Clob type you can set value to it using the setCharacterStream() or setClob() methods. To these methods pass an integer variable representing the parameter index and an object of the Reader class as parameters.

```
pstmt.setString(1, "JavaFX");
pstmt.setString(2, "Java Library");
FileReader reader = new FileReader("E:\images\javafx.txt");
pstmt.setClob(3, reader);
pstmt.execute();
```

Step 4: Execute the statement

Execute the above created PreparedStatement object using the execute() method of the PreparedStatement interface.

Retrieving blob from database

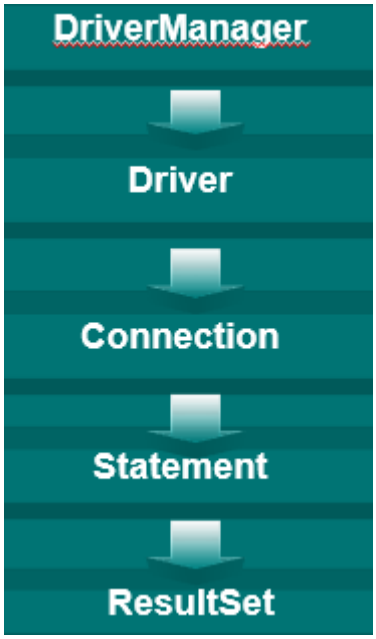
The getClob() method of the ResultSet interface accepts an integer representing the index of the column (or, a String value representing the name of the column) and retrieves the value at the specified column and returns it in the form of a Clob object.

```
while(rs.next()) {
  System.out.println(rs.getString("Name"));
  System.out.println(rs.getString("Type"));
  Clob clob = rs.getClob("Article");
}
```

The getCharacterStream() method of the Clob Interface retrieves the contents of the current Clob object and returns as a Reader object.

Using the getClob() method you can get the contents of the Clob as a Reader object and create text file with the retrieved contents, using the write() method of the FileOutputStream object.

	Reader r = clob.getCharacterStream();
	char cbuf[] = new char[r.read()];
	r.read(cbuf);
	FileOutputStream outPutStream = new
	FileOutputStream("E:\images\clob_output"+i+".txt");
	outPutStream.write(cbuf.toString().getBytes());	

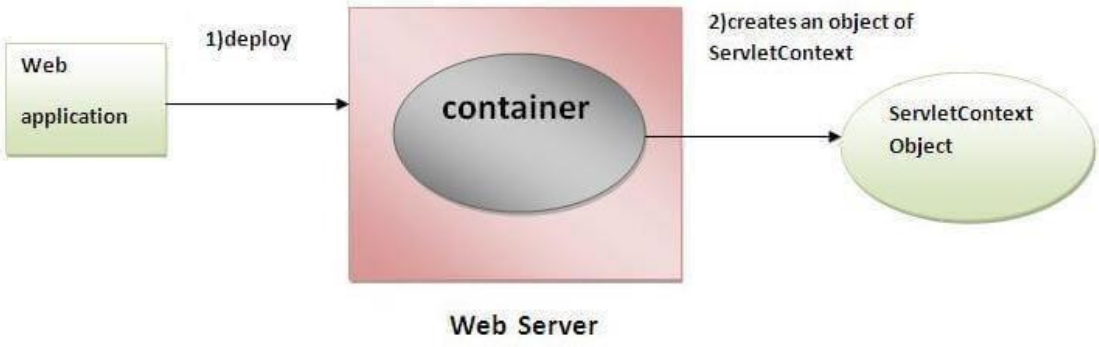
D	State and explain JDBC steps for connecting a java program to a database. JDBC steps with all classes and interface 7M  	7
Q2	ATTEMPT ANY 3 FROM THE FOLLOWING:	[21]

A	Develop servlet application of basic calculator (+,-, *, /) using HttpServletRequest and HttpServletResponse.
	public void doGet(HttpServletRequest request, HttpServletResponse response)
	throws ServletException, IOException {
	response.setContentType("text/html;charset=UTF-8");
	PrintWriter out = response.getWriter();
	out.println("<html><head><title>Servlet CalculatorServlet</title></head><body>"); double n1 = Double.parseDouble(request.getParameter("txtN1"));
		7

	double n2 = Double.parseDouble(request.getParameter("txtN2"));
	double result =0;
	String opr=request.getParameter("opr");
	if(opr.equals("+")) result=n1+n2;
	if(opr.equals("-")) result=n1-n2;
	if(opr.equals("*")) result=n1*n2;
	if(opr.equals("/")) result=n1/n2;
	
	out.println("<h1> Result = "+result);
	out.println("</body></html>");
	}
	
	}	

B	Describe Servletcontext interface with an example. Servletcontext 7M (explanation,methids,syntax and example) An object of ServletContext is created by the web container at time of deploying the project. This object can be used to get configuration information from web.xml file. There is only one ServletContext object per web application. If any information is shared to many servlet, it is better to provide it from the web.xml file using the <context-param> element. Advantage of ServletContext Easy to maintain if any information is shared to all the servlet, it is better to make it available for all the servlet. We provide this information from the web.xml file, so if the information is changed, we don't need to modify the servlet. Thus it removes maintenance problem. Usage of ServletContext Interface There can be a lot of usage of ServletContext object. Some of them are as follows: 1. The object of ServletContext provides an interface between the container and servlet.	7

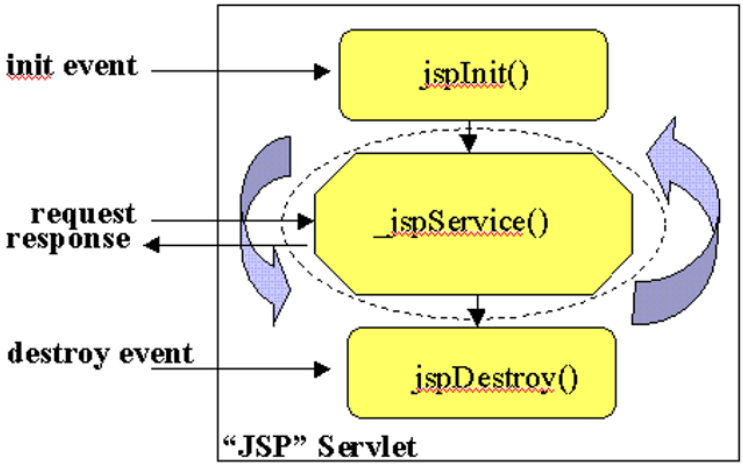
2. The ServletContext object can be used to get configuration information from the web.xml file.
3. The ServletContext object can be used to set, get or remove attribute from the web.xml file.
4. The ServletContext object can be used to provide inter-application communication.



Commonly used methods of ServletContext interface

There is given some commonly used methods of ServletContext interface.

1. public String getInitParameter(String name):Returns the parameter value parameter name.
2. public Enumeration getInitParameterNames():Returns the names of the conte parameters.
3. public void setAttribute(String name,Object object):sets the given object in the
4. public Object getAttribute(String name):Returns the attribute for the specified
5. public Enumeration getInitParameterNames():Returns the names of the conte parameters as an Enumeration of String objects.
6. public void removeAttribute(String name):Removes the attribute with the give servlet context.
7. public class DemoServlet extends HttpServlet{
8. public void doGet(HttpServletRequest req,HttpServletResponse res)
9. throws ServletException,IOException
10. {
11. res.setContentType("text/html");

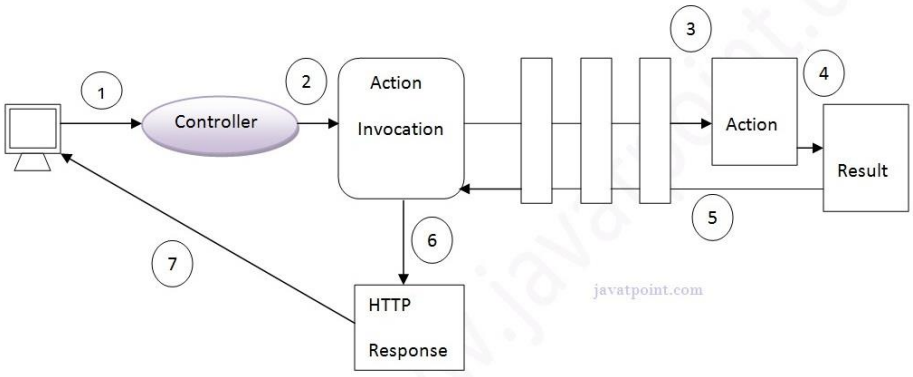
	12. PrintWriter pw=res.getWriter(); 13. 14. //creating ServletContext object 15. ServletContext context=getServletContext(); 16. 17. //Getting the value of the initialization parameter and printing it 18. String driverName=context.getInitParameter("dname"); 19. pw.println("driver name is="+driverName); 20. 21. pw.close(); 22. 23. }}	
C	Describe JSP Life cycle. 3 methods+diagram 7m  	7
D	What are JSP Actions? How they are different from directives? Action explanation + enlisting actions 3m +difference 4 point 4m The action tags are used to control the flow between pages and to use Java Bean.	7

JSP Action Tags	Description
jsp:forward	forwards the request and response to another resource.
jsp:include	includes another resource.
jsp:useBean	creates or locates bean object.
jsp:setProperty	sets the value of property in bean object.
jsp:getProperty	prints the value of property of the bean.
jsp:plugin	embeds another components such as applet.
jsp:param	sets the parameter value. It is used in forward and include mostly.
jsp:fallback	can be used to print the message if plugin is working. It is used in jsp:plugin.

Include Directive	Include Action
Translation time	Run time
Copies the included file	References to the included file
For static content	For dynamic content
Cannot pass parameters	Can pass parameters

Q3	ATTEMPT ANY 3 FROM THE FOLLOWING:	[21]
A	Explain why to use JSON and its applications. Also write a program in JSON to encode a class. use JSON and its applications 3M + program 4M JSON or JavaScript Object Notation is a lightweight, text-based, data-interchange format that follows JavaScript object syntax. JSON is used for data transportation and restoration in places of XML structures. The JSON format is often used when data is sent from a server to a web page or a browser for rendering. The most desired and popular applications of JSON are listed below:	7

	1. It can be used with various modern programming languages such as Python, Ruby, Java, etc.
	2. It is used for data transmission between a server and web applications
	3. JSON is used in JS-based applications, for example- web browser extensions and websites.
	4. APIs and web services use the JSON format for providing public data.
	5. JSON data format simplifies complex data by converting the data extraction process into a meaningful and predictable JSON file format.
	6. import org.json.simple.JSONObject;
	7. public class JsonExample1{
	8. public static void main(String args[]){
	9. JSONObject obj=new JSONObject();
	10. obj.put("name","aron");
	11. obj.put("age",new Integer(27));
	12. obj.put("salary",new Double(600000));
	13. System.out.print(obj);
	14. }}	

B	Explain Struts 2 Architecture in detail. Diagram 3M+ explanation 4M 	7
C	Discuss the different data types in JSON with an example. 5 data types 5m+example 2m o JSONValue o JSONObject o JSONArray o JsonString o JsonNumber	7
D	What is Value Stack in Struts2? What is the execution flow with respect to Value Stack?	7

A valueStack is simply a stack that contains application specific objects such as action objects and other model object.

At the execution time, action is placed on the top of the stack.

We can put objects in the valuestack, query it and delete it.

he value stack can be accessed via the tags provided for JSP, Velocity or Freemarker. There are various tags which we will study in separate chapters, are used to get and set struts 2.0 value stack. You can get valueStack object inside your action as follows −

ActionContext.getContext().getValueStack()

Once you have a ValueStack object, you can use the following methods to manipulate that object −

Sr.No	ValueStack Methods & Description
1	Object findValue(String expr) Find a value by evaluating the given expression against the stack in the default search order.
2	CompoundRoot getRoot() Get the CompoundRoot which holds the objects pushed onto the stack.
3	Object peek() Get the object on the top of the stack without changing the stack.
4	Object pop()

	Get the object on the top of the stack and remove it from the stack. 5 void push(Object o) Put this object onto the top of the stack.	
Q4	ATTEMPT ANY 3 FROM THE FOLLOWING:	[12]
A	MVC is the important feature of SWING over AWT. Justify. Swing uses fewer system resources, adds lots more sophisticated components, helps to tailor the look & feel of the programs. swing development has its root in MVC(Model – View- Controller) architecture. MVC allows swing components to be replaced with different data models & views. Plug gable look & feel is result of MVC architecture. Since java is platform independent & runs on client machine , the look & feel of any platform has to be known Lightweight components:- in swing most of the components have their own view supported by java look & feel classes (it can't rely on native system classes) Pluggable look & feel:- supports cross platform look & feel also called java look & feel that remains same across all platforms wherever the program runs.	4
B	State the 4 session management schemes available in servlets. 1. Cookies 2. HttpSession 3. URL Rewriting 4. Hidden form field	4
C	What is Scrollable and non-Scrollable ResultSet? A scrollable updatable result set maintains a cursor which can both scroll and update rows. Derby only supports scrollable insensitive result sets. To create a scrollable insensitive result set which is updatable, the statement has to be created with concurrency mode ResultSet.	4

	Non-scrollable resultset can only move in forward direction from first to last element and also they cannot move directly to any row in the database	
D	Write a note on OGNL in Struts2. The Object Graph Navigation Language (OGNL) is an expression language. It simplifies the accessibility of data stored in the ActionContext. OGNL is based on a context and Struts builds an ActionContext map for use with OGNL. The ActionContext map consists of the following − • Application − Application scoped variables • Session − Session scoped variables • Root / value stack − All your action variables are stored here • Request − Request scoped variables • Parameters − Request parameters • Atributes − The attributes stored in page, request, session and application scope	4

****