

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

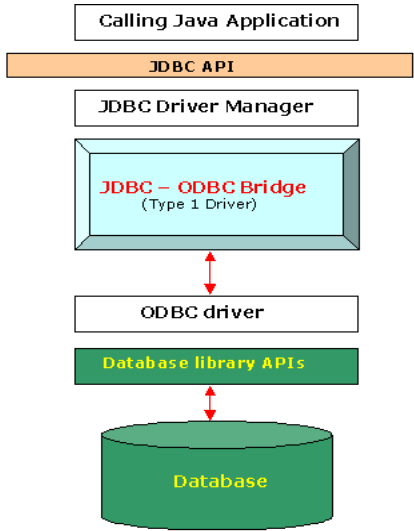
Duration: 2 hrs 30 minutes

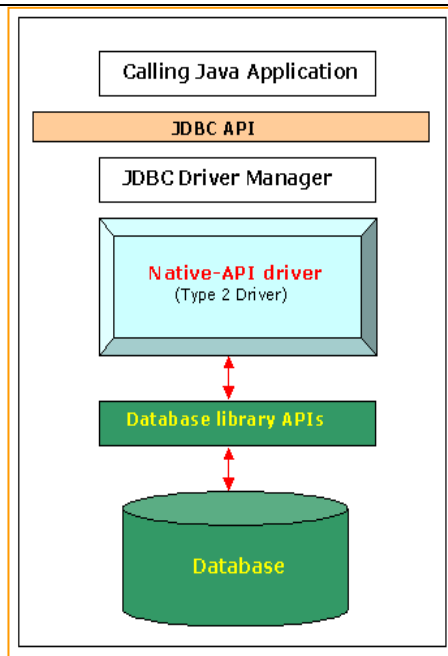
Date:

SOLUTION SET

Q1	ATTEMPT ANY 3 FROM THE FOLLOWING:	[21]
A	<p>Write a swing snippet for generating the following.</p>  <p>Components texts, labels 5m + buttons 2m</p> <pre> public class Registration extends JFrame { JLabel l1, l2, l3, l4, l5, l6, l7, l8; JTextField tf1, tf2, tf5, tf6, tf7; JButton btn1, btn2; JPasswordField p1, p2; Registration() { setVisible(true); setSize(700, 700); setLayout(null); setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE); setTitle("Registration Form in Java"); l1 = new JLabel("Registration Form in Windows Form:"); l1.setForeground(Color.blue); l1.setFont(new Font("Serif", Font.BOLD, 20)); </pre>	7

	<pre> l2 = new JLabel("Name:"); l3 = new JLabel("Email-ID:"); l4 = new JLabel("Create Passowrd:"); l5 = new JLabel("Confirm Password:"); l6 = new JLabel("Country:"); l7 = new JLabel("State:"); l8 = new JLabel("Phone No:"); tf1 = new JTextField(); tf2 = new JTextField(); p1 = new JPasswordField(); p2 = new JPasswordField(); tf5 = new JTextField(); tf6 = new JTextField(); tf7 = new JTextField(); btn1 = new JButton("Submit"); btn2 = new JButton("Clear"); btn1.addActionListener(this); btn2.addActionListener(this); l1.setBounds(100, 30, 400, 30); l2.setBounds(80, 70, 200, 30); l3.setBounds(80, 110, 200, 30); l4.setBounds(80, 150, 200, 30); l5.setBounds(80, 190, 200, 30); l6.setBounds(80, 230, 200, 30); l7.setBounds(80, 270, 200, 30); l8.setBounds(80, 310, 200, 30); tf1.setBounds(300, 70, 200, 30); tf2.setBounds(300, 110, 200, 30); p1.setBounds(300, 150, 200, 30); p2.setBounds(300, 190, 200, 30); tf5.setBounds(300, 230, 200, 30); tf6.setBounds(300, 270, 200, 30); tf7.setBounds(300, 310, 200, 30); btn1.setBounds(50, 350, 100, 30); btn2.setBounds(170, 350, 100, 30); add(l1); add(l2); add(tf1); add(l3); add(tf2); add(l4); add(p1); add(l5); </pre>	
--	---	--

	<pre> add(p2); add(l6); add(tf5); add(l7); add(tf6); add(l8); add(tf7); add(btn1); add(btn2); } </pre>	
B	<p>What is a Driver? Explain any two types of drivers in JDBC.</p> <p>Driver 1m+ 6m for 2 drivers</p> <p>A JDBC driver (Java Database Connectivity driver) is a small piece of software that allows JDBC to connect to different databases. Essentially, a JDBC driver makes it possible to do three things: Establish a connection with a data source. Send queries and update statements to the data source. Process the results.</p> <ul style="list-style-type: none"> • Type I: “Bridge” - • Type II: “Native” - • Type III: “Middleware” - • Type IV: “Pure” <p>Type I Drivers</p>  <pre> graph TD A[Calling Java Application] --> B[JDBC API] B --> C[JDBC Driver Manager] C --> D["JDBC – ODBC Bridge (Type 1 Driver)"] D <--> E[ODBC driver] E --> F[Database library APIs] F <--> G[(Database)] </pre> <p>Type II Drivers</p>	7



C Differentiate between AWT and Swing.
7 points 7m

7

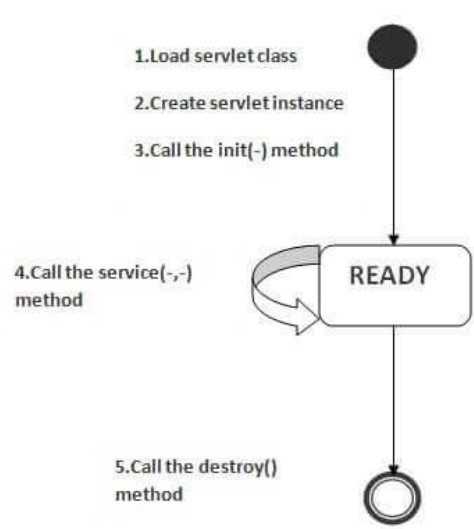
S.NO	AWT	Swing
1.	Java AWT is an API to develop GUI applications in Java	Swing is a part of Java Foundation Classes and is used to create various applications.
2.	The components of Java AWT are heavy weighted.	The components of Java Swing are light weighted.
3.	Java AWT has comparatively less functionality as compared to Swing.	Java Swing has more functionality as compared to AWT.
4.	The execution time of AWT is more than Swing.	The execution time of Swing is less than AWT.
5.	The components of Java AWT are platform dependent.	The components of Java Swing are platform independent.
6.	MVC pattern is not supported by AWT.	MVC pattern is supported by Swing.
7.	AWT provides comparatively less powerful components.	Swing provides more powerful components.

D Describe any 4 methods and any 3 fields used for navigation through database records using a ResultSet object.

4 methods 4m+ 3 fields 3m

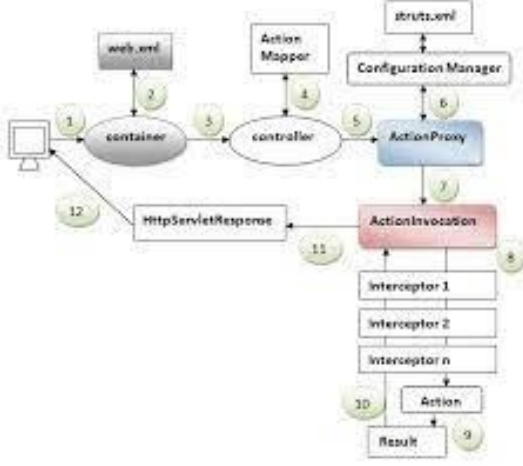
1. boolean first()
2. boolean isFirst()
3. boolean beforeFirst()
4. boolean isbeforeFirst()
1. TYPE_FORWARD_ONLY

7

	2. TYPE_SCROLL_SENSITIVE 3. TYPE_SCROLL_INSENSITIVE	
Q2	ATTEMPT ANY 3 FROM THE FOLLOWING:	[21]
A	<p>Define a servlet? Explain its life cycle methods. Servlet 2m +5m for life cycle</p> <p>Servlet technology is used to create a web application (resides at server side and generates a dynamic web page).</p> <p>Servlet technology is robust and scalable because of java language. Before Servlet, CGI (Common Gateway Interface) scripting language was common as a server-side programming language.</p>  <pre> graph TD Start(()) --> 1[1. Load servlet class] 1 --> 2[2. Create servlet instance] 2 --> 3[3. Call the init(-) method] 3 --> Ready([READY]) Ready --> 4[4. Call the service(-, -) method] 4 --> Ready Ready --> 5[5. Call the destroy() method] 5 --> End((())) </pre>	7
B	<p>Explain forward and include actions in JSP with an example. Forward 3.5m + Include 3.5m</p> <p>jsp:forward action tag</p> <p>The jsp:forward action tag is used to forward the request to another resource it may be jsp, html or another resource.</p> <p>Syntax of jsp:forward action tag without parameter</p> <pre><jsp:forward page="relativeURL <%= expression %>" /></pre> <p>Syntax of jsp:forward action tag with parameter</p> <pre><jsp:forward page="relativeURL <%= expression %>"> <jsp:param name="parametername" value="parametervalue" /> <%=expression%>" /> </jsp:forward></pre> <p>The jsp:include action tag is used to include the content of another resource it may be jsp, html or servlet.</p>	7

	<p>The jsp include action tag includes the resource at request time so it is better for dynamic pages because there might be changes in future.</p> <p>The jsp:include tag can be used to include static as well as dynamic pages.</p> <p>Advantage of jsp:include action tag</p> <p>Code reusability : We can use a page many times such as including header and footer pages in all pages. So it saves a lot of time.</p> <p>Syntax of jsp:include action tag without parameter</p> <pre><jsp:include page="relativeURL <%= expression %>" /></pre> <p>Syntax of jsp:include action tag with parameter</p> <pre><jsp:include page="relativeURL <%= expression %>"> <jsp:param name="parametername" value="parametervalue <%=expression%>" /> </jsp:include></pre>	
C	<p>Describe servletconfig interface in detail.</p> <p>If the configuration information is modified from the web.xml file, we don't need to change the servlet. So it is easier to manage the web application if any specific content is modified from time to time.</p> <p>Advantage of ServletConfig</p> <p>The core advantage of ServletConfig is that you don't need to edit the servlet file if information is modified from the web.xml file.</p> <p>If the configuration information is modified from the web.xml file, we don't need to change the servlet. So it is easier to manage the web application if any specific content is modified from time to time.</p> <p>Advantage of ServletConfig</p> <p>The core advantage of ServletConfig is that you don't need to edit the servlet file if information is modified from the web.xml file.</p> <p>Methods of ServletConfig interface</p> <p>public String getInitParameter(String name):Returns the parameter value for the specified parameter name.</p> <p>public Enumeration getInitParameterNames():Returns an enumeration of all the initialization parameter names.</p> <p>public String getServletName():Returns the name of the servlet.</p> <p>public ServletContext getServletContext():Returns an object of ServletContext.</p>	7
D	<p>Write a servlet code to show the use of request dispatcher, also write web.xml.</p> <pre>String n=request.getParameter("userName"); String p=request.getParameter("userPass"); if(p.equals("servlet"){</pre>	7

	<pre>RequestDispatcher rd=request.getRequestDispatcher("servlet2"); rd.forward(request, response); } else{ out.print("Sorry UserName or Password Error!"); RequestDispatcher rd=request.getRequestDispatcher("/index.html"); rd.include(request, response); <servlet-mapping> <servlet-name>Login</servlet-name> <url-pattern>/servlet1</url-pattern> </servlet-mapping> <servlet-mapping> <servlet-name>WelcomeServlet</servlet-name> <url-pattern>/servlet2</url-pattern> </servlet-mapping></pre>															
Q3	ATTEMPT ANY 3 FROM THE FOLLOWING:	[21]														
A	<p>What is JSON? Differentiate between JSON and XML.</p> <p>JSON 2m+ json vs xml 5m</p> <p>JSON stands for JavaScript Object Notation. JSON is a lightweight data-interchange format. JSON is plain text written in JavaScript object notation. JSON is used to send data between computers. JSON is language independent</p> <table> <tr> <td>It is JavaScript Object Notation</td> <td>It is Extensible markup language</td> </tr> <tr> <td>It is based on JavaScript language.</td> <td>It is derived from SGML.</td> </tr> <tr> <td>It is a way of representing objects.</td> <td>It is a markup language and uses tag structure to represent data items.</td> </tr> <tr> <td>It does not provides any support for namespaces.</td> <td>It supports namespaces.</td> </tr> <tr> <td>It supports array.</td> <td>It doesn't supports array.</td> </tr> <tr> <td>Its files are very easy to read as compared to XML.</td> <td>Its documents are comparatively difficult to read and interpret.</td> </tr> <tr> <td>It doesn't use end tag.</td> <td>It has start and end tags.</td> </tr> </table>	It is JavaScript Object Notation	It is Extensible markup language	It is based on JavaScript language.	It is derived from SGML.	It is a way of representing objects.	It is a markup language and uses tag structure to represent data items.	It does not provides any support for namespaces.	It supports namespaces.	It supports array.	It doesn't supports array.	Its files are very easy to read as compared to XML.	Its documents are comparatively difficult to read and interpret.	It doesn't use end tag.	It has start and end tags.	7
It is JavaScript Object Notation	It is Extensible markup language															
It is based on JavaScript language.	It is derived from SGML.															
It is a way of representing objects.	It is a markup language and uses tag structure to represent data items.															
It does not provides any support for namespaces.	It supports namespaces.															
It supports array.	It doesn't supports array.															
Its files are very easy to read as compared to XML.	Its documents are comparatively difficult to read and interpret.															
It doesn't use end tag.	It has start and end tags.															
B	<p>What are interceptors in Struts2? What is the execution flow with respect to interceptor?</p> <p>Interceptors 2m+ the execution flow 5m</p> <p>Interceptor is an object that is invoked at the preprocessing and postprocessing of a request. In Struts 2, interceptor is used to perform operations such as validation, exception handling, internationalization, displaying intermediate result etc.</p>	7														

	<p>Advantage of interceptors</p> <p>Pluggable If we need to remove any concern such as validation, exception handling, logging etc. from the application, we don't need to redeploy the application. We only need to remove the entry from the struts.xml file.</p> 	
C	<p>Explain the components of struts framework.</p> <p>Any 5 for 7m</p> <ul style="list-style-type: none"> • Filter Dispatcher. • Action. • Result. • Configuration file. • Interceptors. • Deployment descriptor. • Tag Library. 	7
D	<p>Write a program to depict MVC using struts2.</p> <p>Main snippet with other assumptions 7m</p> <ol style="list-style-type: none"> 1. <code><?xml version="1.0" encoding="UTF-8" ?></code> 2. <code><!DOCTYPE struts PUBLIC "-//Apache Software Foundation//DTD Struts Configuration 2.1//EN" "http://struts.apache.org/dtds/struts-2.1.dtd"></code> 3. <code><struts></code> 4. <code><package name="default" extends="struts-default"></code> 5. <code><action name="product" class="com.Product"></code> 6. <code><result name="success">welcome.jsp</result></code> 7. <code></action></code> 8. <code></package></code> 9. <code></struts></code> 	7

Q4	ATTEMPT ANY 3 FROM THE FOLLOWING:	[12]
A	<p>What is the use of BLOB and CLOB?</p> <p>2m each one</p> <p>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.</p> <p>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.</p> <p>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).</p> <p>MYSQL database provides support for this datatype using four variables.</p> <ul style="list-style-type: none"> • 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. 	4
B	<p>State the three directive elements available in JSP.</p> <p>3 elements for 4m</p> <p>Page</p> <p>Include</p> <p>Taglib</p>	4
C	<p>What is action entity in Struts2 framework?</p> <p>Struts 2 Action</p> <p>Struts 2 Action</p> <p>Action Interface</p> <p>ActionSupport class</p> <p>In struts 2, action class is POJO (Plain Old Java Object).</p> <p>POJO means you are not forced to implement any interface or extend any class.</p> <p>Generally, execute method should be specified that represents the business logic. The simple action class may look like:</p> <p>Welcome.java</p> <pre>package com.mithi; public class Welcome { public String execute(){</pre>	4

	<pre> return "success"; } }</pre>	
D	<p>Write a code snippet to implement JSON. 4m for program to encode or decode</p> <pre> import org.json.simple.JSONObject; public class JsonExample1 { public static void main(String args[]){ JSONObject obj=new JSONObject(); obj.put("name","sonuj"); obj.put("age",new Integer(27)); obj.put("salary",new Double(600000)); System.out.print(obj); } }</pre>	4