Advanced Java Solution Set October 2016

(2½ hours)

Total Marks: 75

N. B.: (1) **All** questions are **compulsory**.
(2) Make **suitable assumptions** wherever necessary and **state the assumptions** made.
(3) Answers to the **same question** must be **written together**.
(4) Numbers to the **right** indicate **marks**.
(5) Draw **neat labeled diagrams** wherever **necessary**.
(6) Use of **Non-programmable** calculators is **allowed**.

1 Attempt *any two* of the following: **10**

a. What is event? [1 mark]
List various event classes in Java. [1 mark]
Explain any two event class. [3 mark]
In the delegation model, an *event* is an object that describes a state change in a source. It can be generated as a consequence of a person interacting with the elements in a graphical user interface. Some of the activities that cause events to be generated are pressing a button, entering a character via the keyboard, selecting an item in a list, and clicking the mouse.

Event Class	Description
ActionEvent	Generated when a button is pressed, a list item is double-clicked, or a menu item is selected.
AdjustmentEvent	Generated when a scroll bar is manipulated.
ComponentEvent	Generated when a component is hidden, moved, resized, or becomes visible.
ContainerEvent	Generated when a component is added to or removed from a container.
FocusEvent	Generated when a component gains or loses keyboard focus.
InputEvent	Abstract superclass for all component input event classes.
ItemEvent	Generated when a check box or list item is clicked; also occurs when a choice selection is made or a checkable menu item is selected or deselected.
KeyEvent	Generated when input is received from the keyboard.
MouseEvent	Generated when the mouse is dragged, moved, clicked, pressed, or released; also generated when the mouse enters or exits a component.
MouseWheelEvent	Generated when the mouse wheel is moved.
TextEvent	Generated when the value of a text area or text field is changed.
WindowEvent	Generated when a window is activated, closed, deactivated, deiconified, iconified, opened, or quit.

b What is the use of adapter class? [2 mark]
Explain with suitable example. [3 mark]
An *adapter class* can simplify the creation of event handlers in certain situations. An adapter class provides an empty implementation of all methods in an event listener interface. Adapter classes are useful when you want to receive and process only some of the events that are handled by a particular event listener interface. You can define a new class to act as an event listener by extending one of the adapter classes and implementing only those events in which you are interested.
Following is a sample code to illustrate adapter class
import java.awt.*;

```
import java.awt.event.*;
import java.applet.*;
/*<applet code="AdapterDemo" width=300 height=100>
</applet>
*/
public class AdapterDemo extends Applet {
public void init() {
addMouseListener(new MyMouseAdapter(this));
addMouseMotionListener(new MyMouseMotionAdapter(this));
}
}
class MyMouseAdapter extends MouseAdapter {
AdapterDemo adapterDemo;
public MyMouseAdapter(AdapterDemo adapterDemo) {
this.adapterDemo = adapterDemo;
}
// Handle mouse clicked.
public void mouseClicked(MouseEvent me) {
adapterDemo.showStatus("Mouse clicked");
}
}
class MyMouseMotionAdapter extends MouseMotionAdapter {
AdapterDemo adapterDemo;
public MyMouseMotionAdapter(AdapterDemo adapterDemo) {
this.adapterDemo = adapterDemo;
}
// Handle mouse dragged.
public void mouseDragged(MouseEvent me) {
adapterDemo.showStatus("Mouse dragged");
}
}
```

c. List various layouts used in Java. [1 mark]
Explain any two with example. [4 mark]

- FlowLayout
- BorderLayout
- GridLayout
- CardLayout
- GridBagLayout

d Write AWT based Java program that will read a number from user(textbox) and display its factorial(label). [5 mark]

```
import java.awt.*;
import java.awt.event.*;
/*<applet code=FactEvent width=400 height=400></applet>*/

 public class FactEvent extends java.applet.Applet implements ActionListener
{
            TextField t1

            int fact=1,m;
```

```
        Button b1,b2,b3;
        String msg;
        Label l1,lf;
        FactEvent e;
        public void init()
        {       e=this;
                t1=new TextField(3);
                b1=new Button("FIND FACTORIAL");
                l1=new Label("ENTER THE NUMBER");
                lf=new Label("Factorial : ");

                add(l1);
                add(t1);
                add(l2);
                add(lf);
                add(b1);
                b1.addActionListener(this);

        }

        public void actionPerformed(ActionEvent ae)
        {
                String str=t1.getText();
                if(str!="")
                {
                int num=Integer.parseInt(str);
                for(int i=num;i>0;i--)
                {
                        fact=fact*i;
                }

                msg="" +fact;
                lf.setText(msg);
                fact=1;
                }

        }
}
```

2 Attempt *any two* of the following: **10**

a. Write a Java program using swing components to illustrate the use of JSplitPane. [5 mark]

b What is purpose of JFileChooser? [1 mark]
Write down constructors and methods of the same. [4 mark]

JFileChooser provides a simple mechanism for the user to choose a file.
The following code pops up a file chooser for the user's home directory that sees only .jpg and .gif images:

```
JFileChooser chooser = new JFileChooser();
FileNameExtensionFilter filter = new FileNameExtensionFilter(
```

```
    "JPG & GIF Images", "jpg", "gif");
chooser.setFileFilter(filter);
int returnVal = chooser.showOpenDialog(parent);
if(returnVal == JFileChooser.APPROVE_OPTION) {
   System.out.println("You chose to open this file: " +
        chooser.getSelectedFile().getName());
}
```

JFileChooser() Constructs a JFileChooser pointing to the user's default directory.
JFileChooser(**File** currentDirectory) Constructs a JFileChooser using the given File as the path.
JFileChooser(**File** currentDirectory, **FileSystemView** fsv) Constructs a JFileChooser using the given current directory and FileSystemView.
JFileChooser(**FileSystemView** fsv) Constructs a JFileChooser using the given FileSystemView.
JFileChooser(**String** currentDirectoryPath) Constructs a JFileChooser using the given path.
JFileChooser(**String** currentDirectoryPath, **FileSystemView** fsv) Constructs a JFileChooser using the given current directory path and FileSystemView.

c. How messagebx is displayed in Java? [3 mark]
Give suitable example. [2 mark]
Messagebox is displayed with the help of JoptionPane Class.
JOptionPane.showMessageDialog
(null, “Hello, this message is in a message type box.");

d Differentiate between JTextArea, JTextPasswordField and JTextField. [5 mark]

3 Attempt *any two* of the following: **10**

a. Write the purpose of following methods of servlet interface [1 mark per each]

i. init : Used for initializing the Servlet parameters provided by the ServletConfig object.
Is called only once when the Servlet is loaded first time.
None of the Servlets method can be called unless the Servlet is initialized using init().

ii. destroy Is called only once immediately before the Servlet is unloaded.
Is used to clear all retained resources (eg. Database connection, file handlers etc.)

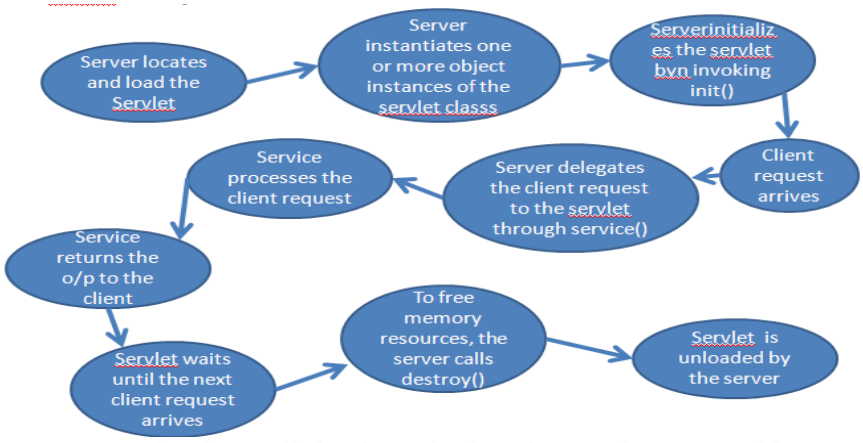
iii. service : Is the heart of the Servlet. (Request-response model.)
Is called to handle a single client request.

iv. getServletConfig : Provides the ServletConfig object for initializing the Servlet’s paramete

v. getServletInfo : Provides the Servlet metadata (eg. Author, Servlet version copyright info etc.).
It need to be overridden in Servlet to return the required information

b What are the merits and demerits of servlet? Exlain.

c. With suitable diagram explain the life cycle of servlet.



d Write a servlet that will display whether the number entered by user is odd or even.

4 Attempt *any two* of the following: **10**

a. List various classes used in JDBC. [1 mark]
Also write the purpose of each. [4 mark]
JDBC Classes, Intefaces and exceptions include
Driver (Interface) gives JDBC a launching point for database connectivity by responding to DriverManager connection requests and providing information about the implementation in question.
DriverManager keeps a list of all classes that implement the Driver Interface. Whaen an application is run, it loads all the drivers found in the memory. When opening a connection to a database it selects most appropriate driver from the previously loaded drivers.
Connection (Interface) represents a connection with a data source. This interface can be used to retrieve information regarding the tables in the database to which connection is opened.
Statement (Interface) represents static SQL statements that can be used to retrieve ResultSet object(s). Objective is to pass to the database the SQL command for execution and to retrieve output results from the database in the form of the ResultSet.
ResultSet is databse result set generated from a currently executed SQL statement. The data from the query is delivered in the form of a table. The rows of the table are returned to the program in sequence.
RowSet Rowset object extends ResultSet interface to add support for disconnected result set and thereby helps in retrieval of dat completely.
PreparedStatement PreparedStatement object is an SQL statement that is pre-compiled and stored. This object can then be executed multiple times much more efficiently than preparing and issuing the same statement each time it is needed.
CallableStatement CallableStatement represents a stored procedure. It can be used to execute stored procedure in a RDBMS that supports them.
DataSource DataSource object abstracts a data source. Can be used in place of DriverManager to eficiently obtain data source connection.

b How result of a query is processed in JDBC? [2 mark]
Explain with suitable example. [3 mark]

c. List [1 mark]
and explain JSP directives. [4 mark]
page
inclued
taglib

d Explain the include and forward action element of JSP.
jsp:include : Includes a file at the time the page is requested
jsp:forward : Forwards the requester to a new page

5 Attempt *any two* of the following: **10**

a. List [1 mark]
and explain different types of Enterprise Beans? [4 mark]
Session
A **session bean** encapsulates business logic that can be invoked programmatically by a client over local, remote, or web service client views. To access an application that is deployed on the server, the client invokes the session bean's methods. The session bean performs work for its client, shielding it from complexity by executing business tasks inside the server.
A session bean is not persistent.
Types of Session Beans
Session beans are of three types: stateful, stateless, and singleton.
Stateful Session Beans
Stateless Session Beans
Singleton Session Beans

Message-driven
A **message-driven bean** is an enterprise bean that allows Java EE applications to process messages asynchronously.
This type of bean normally acts as a JMS message listener, which is similar to an event listener but receives JMS messages
instead of events. The messages can be sent by any Java EE component (an application client, another enterprise bean, or a
web component) or by a JMS application or system that does not use Java EE technology.

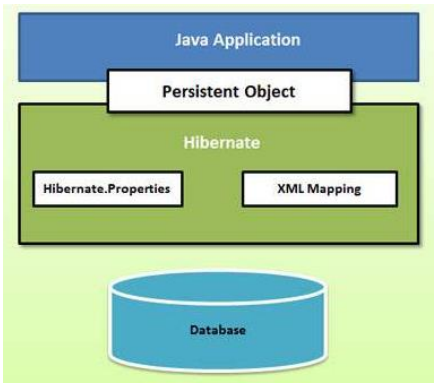
b Write the benefits of EJB.

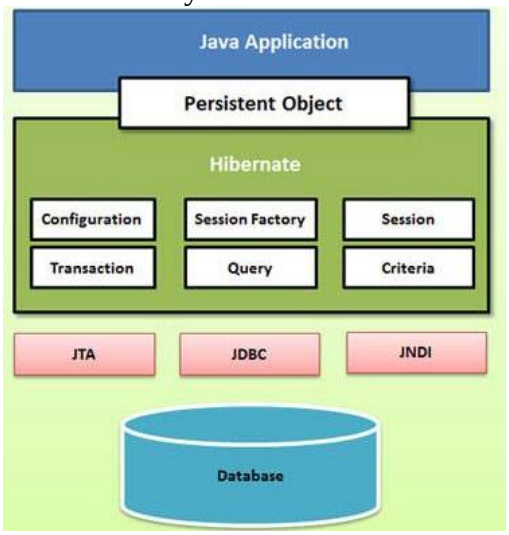
c. List[1 mark]
and explain the JSF lifecycle phases. [4 mark]
- Restore view phase
- Apply request values phase
- Process validations phase
- Update model values phase
- Invoke application phase
- Render response phase

d What is facelet? [1 mark]
Write the features of facelets [4 mark]
Facelets is a powerful but lightweight page declaration language that is used to build JavaServer Faces views using HTML
style templates and to build component trees.
Facelets features include the following:
- Use of XHTML for creating web pages
- Support for Facelets tag libraries in addition to JavaServer Faces and JSTL tag libraries
- Support for the Expression Language (EL)
- Templating for components and pages

6 Attempt *any two* of the following: **10**

a. With suitable diagram explain the architecture of Hybernate.





b Write a short on Interceptors in struts.

c. What is the use of Action component of Strut 2? [2 mark]
Write and explain various roles of Action component. [3 mark]

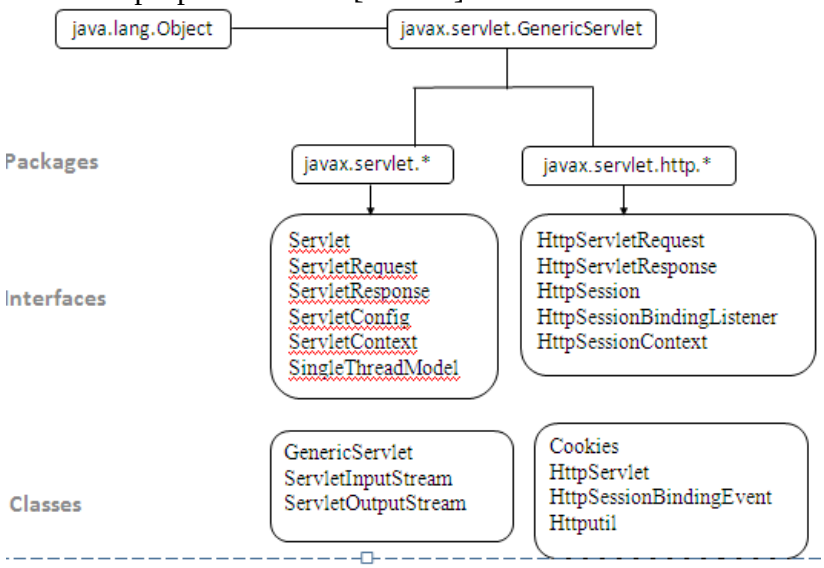
d What is the importance of hibernate mapping file? [3 mark]
Explain with suitable example. [2 mark]

7 Attempt *any three* of the following: **15**

a. Write AWT based Java program that demonstrate the use of textbox, label and checkbox.

b Explain how to assign color to frame background using JColorChooser class.

c. List various classes of Servlet API. [1 mark]
Write the purpose of each. [4 mark]



d What are the differenttypes of statements in JDBC? Explain.

e. Explain with suitable example <navigation-rule> element of faces-config.xml file of JSF.

f. What is the need of Hibernate?