Important Instructions to examiners:

1) The answers should be examined by key words and not as word-to-word as given in themodel answer scheme.

2) The model answer and the answer written by candidate may vary but the examiner may tryto assess the understanding level of the candidate.

3) The language errors such as grammatical, spelling errors should not be given moreImportance (Not applicable for subject English and Communication Skills).

4) While assessing figures, examiner may give credit for principal components indicated in thefigure. The figures drawn by candidate and model answer may vary. The examiner may give credit for anyequivalent figure drawn.

5) Credits may be given step wise for numerical problems. In some cases, the assumed constantvalues may vary and there may be some difference in the candidate's answers and model answer.

6) In case of some questions credit may be given by judgment on part of examiner of relevant answer based on candidate's understanding.

7) For programming language papers, credit may be given to any other program based on equivalentconcept.

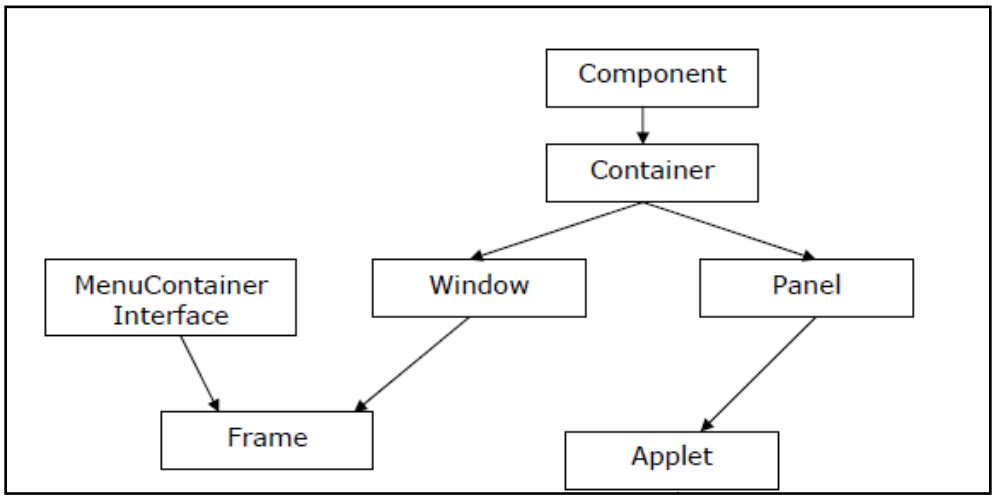
Marks

1. a) Attempt any THREE of the following: **12**

(i) Write AWT class hierarchy and state the used of panel class.

(Diagram - 2 Marks, Panel - 2 Marks)

Ans: The AWT defines windows according to a class hierarchy that adds functionality with each level. The two most common windows are those derived from Panel, which is used by applets, and those derived from Frame, which creates a standard window. Much of the functionality of these windows is derived from their parent classes. Thus, a description of the class hierarchies relating to these two classes is fundamental to their understanding. Figure below shows the class hierarchy for Panel



Applet class hierarchy

Panel Panel is the simplest container class. A panel provides space in which an application can attach any other component, including other panels.

The default layout manager for a panel is the FlowLayout layout manager.

It doesn't add any new methods; it simply implements Container. Panel is the super-class for Applet. When screen output is directed to an applet, it is drawn on the surface of a Panel object. In essence, **a Panel is a window that does not contain a title bar, menu bar, or border**. That is why we don't see these items when an applet is run inside a browser.

(ii) Explain following terms:

1) IP address

2) Port

(Explanation of IP address - 2 Marks and port explanation - 2 Marks)

Ans:

1) **IP address:** IP address is a four byte (32-bit) address, which is usually expressed in dotted decimal format. IP address is often referred to as sequence between 0 to 255 separated by dots (.). Internet address is a number that uniquely identifies a computer on the net.

Example: 192.168.0.1

2) **Port:** Port is addresses within computers that are used to enable communication between two or more computers (client and server). TCP/IP reserves the lower 1024 ports for specific protocol.

Example: port number 21 is for FTP, 23 is for Telnet,http is a protocol that web browsers and servers use to transfer text and images,etc.

(iii) List names of all four JDBC drivers.

(For each type - 1 Mark)

Ans: JDBC technology-based drivers generally fit into one of four categories

1. JDBC-ODBC bridge plus ODBC driver:
2. Native-API partly-Java driver
3. JDBC-Net pure Java driver
4. Native-protocol pure Java driver

(iv) State any four features of swing.

(For each - 1 Mark)

Ans: 1. Swing is a set of classes that provides more flexible and powerful components.

2. Swing components are light-weight components.
3. Swing components are 100% pure Java components.
4. Swing components have pluggable look and feel with rich user interface components
5. Swing components follow Model View Architecture(MVC) design pattern.
6. In Swing,you can add images and pictures as ImageIconin almost all swing components.
7. Swing components do need native support.
8. Swing components are not implemented by platform-specific code.

b) Attempt any ONE of the following: 06

(i) Write a program to read contents of html file from url and display on screen using URL connection class.

(import statement - 1 Mark , URL and URLConnection - 2 Marks ,logic to retrieve content form content length - 3 Marks)

Ans:// Demonstrate URLConnection.

```
import java.net.*;
import java.io.*;
importjava.util.Date;
classUCDemo
{
public static void main(String args[]) throws Exception {
int c;
URL hp = new URL("http://www.internic.net");
URLConnectionhpCon = hp.openConnection();
// get date
long d = hpCon.getDate();
if(d==0)
System.out.println("No date information.");
else
System.out.println("Date: " + new Date(d));
// get content type
System.out.println("Content-Type: " + hpCon.getContentType());
// get expiration date
d = hpCon.getExpiration();
if(d==0)
System.out.println("No expiration information.");
else
System.out.println("Expires: " + new Date(d));
// get last-modified date
d = hpCon.getLastModified();
if(d==0)
System.out.println("No last-modified information.");
else
System.out.println("Last-Modified: " + new Date(d));
```

```
// get content length
intlen = hpCon.getContentLength();
if(len == -1)
System.out.println("Content length unavailable.");
else
System.out.println("Content-Length: " + len);
if(len != 0) {
System.out.println("=== Content ===");
InputStream input = hpCon.getInputStream();
int i = len;
while (((c = input.read()) != -1)) { // && (--i > 0)) {
System.out.print((char) c);
}
input.close();
} else {
System.out.println("No content available.");
}
}
}
```

[Note: Content length code to be considered as valid answer.]

(ii) Design a servlet to accept a username and password in two text boxes and display message as 'weak password' if it contain less than five characters.

(HTML form code - 2 Marks with named html parameter, Servlet import - 1 Mark, doGet or doPost with correct logic to check length of password - 3 Marks)

Ans:

```
// Client Program
<html>
<head>
<title> Accept user name & password</title>
</head>
<body>
```

```
<form name=”form1” method=get    action = “http://localhost:8080/ex1/ServletDemo”>
Name : <input type=text  name=”user1”>
Password : <input type = text  name= “password1” >
<input type= submit” value=”Login”>
<input type= submit” value=”Reset”>
</form>
</body>
</html>

//Server Program
importjavax.servlet.*;
importjavax.servlet.http.*;
public class ServletDemo extends HttpServlet
{
public void doGet(ServletRequestreq, ServletResponse res) throws
IOException,ServletException
{
res.setContentType(“text/html”);
        PrintWriter out =res.getWriter();
        out.println(“<html>”);
        out.println(“<head><title> Accept user name & password</title></head>
        out.println(“<body>”);
        String name=(String) req.getParameter(“user1”);
        String password=(String) req.getParameter(“password1”);
intlen=password.length();
if (len<=5)
out.println(“Your password is weak”);
else
{
        out.println(“Name =”+name +”<br>”);
        out.println(“Password=”+password +”<br>”);
```

```
}
        out.println("</body></html>");

}
}
```

2. Attempt any TWO of the following: 16

a) Create an application which contains three scrollbars for changing the background colours of window with three colours as red, green and blue respectively.

(import statement - 1 Mark ,Scrollbar initialization - 2 Marks, Listener registration with event handling - 4 Marks, Applet tag - 1 Mark)

Ans:

```
import java.awt.event.*;
import java.awt.*;
importjava.applet.*;
public class Scroll extends Applet implements AdjustmentListener
{
        Scrollbar r,g,b;
        public void init()
        {
        r = new ScrollBar(ScrollBar.VERTICAL,255,5,0,255);
                g = new ScrollBar(ScrollBar.VERTICAL,255,5,0,255);
                b = new ScrollBar(ScrollBar.VERTICAL,255,5,0,255);
add(r);
add(g);
add(b);
                r.addAdjustmentListener(this);
                g.addAdjustmentListener(this);
                b.addAdjustmentListener(this);
        }
public void adjustmentValueChanged(AdjustmentEventae)
        {
```

```
setBackround(new Color(r.getValue(), g.getValue(), b.getValue()));

        }
}
/*<applet code="Scroll.class" height=300 width=400>
</applet>*/
```

[**Note: Code written with frame implementation to be considered]**

b) Explain with suitable examplehow to create and use dialog box in java.

Ans: Dialog Box Control ***(2 Marks)***

The dialog box to hold a set of related controls. Dialog boxes are primarily used to obtain user input. They are similar to frame windows, except that dialog boxes are always child windows of a top-level window. Also, dialog boxes don't have menu bars. In other respects, dialog boxes function like frame windows. (You can add controls to them, for example, in the same way that you add controls to a frame window.) Dialog boxes may be **modal or modeless**. When a modal dialog box is active, all input is directed to it until it is closed. This means that you cannot access other parts of your program until you have closed the dialog box. When a modeless dialog box is active, input focus canbe directed to another window in your program. Thus, other parts of your program remain active and accessible.

Dialog boxes are of type Dialog. ***(2 Marks)***

Two commonly used constructors are shown here: **Dialog (Frame parentWindow, booleanmode) Dialog (Frame parentWindow, String title, booleanmode)** Here, parentWindowis the owner of the dialog box. If mode is **true**, the dialog box is modal. Otherwise, it is modeless. The title of the dialog box can be passed in title. Generally, you will subclass **Dialog**, adding the functionality required by your application. Following is a modified version of the preceding menu program that displays a modeless dialog box when the New option is chosen. Notice that when the dialog box is closed, **dispose( )** is called. This method is defined by **Window**, and it frees all system resources associated with the dialog box window.

Example:*(4 Marks)*

```
import java.awt.*;
```

```
public class FrameDemo extends Frame
{

publicFrameDemo(String title)
      {

super(title);
setVisible(true);
       Dialog d=new Dialog(this," My Dialog", true);
d.show();
d.setSize(100,100);
d.setVisible(true);

      }

public static void main(String args[])
      {
FrameDemo f=new FrameDemo("Frame Demo");
f.setVisible(true);
f.setSize(200,200);
f.setLocation(10,10);

      }
}
```

c) Explain use of Socket and ServerSocket classes with suitable example.

(ServerSocket – 2 Marks, Client Socket – 2 Marks)

Ans: ServerSocket:

A ServerSocket handles the requests and sends back an appropriate reply. The actual tasks that a server socket must accomplish are implemented by an internal SocketImplinstance.A server socket waits for requests to come in over the network. It performs some operation based on that

request, and then possibly returns a result to the requester.The actual work of the server socket is performed by an instance of the SocketImpl class. An application can change the socket factory that creates the socket implementation to configure itself to create sockets appropriate to the local firewall.

Client socket:

We have used the class named ClientSocketInformation.java that implement the constructor of the Socket class passing two arguments as hostName and TIME_PORT. This program throws an IOException for the exception handling. This exception is thrown to indicate an I/O problem of some sort occurred. Here, we are going to explore a method to retrieve the host name of the local system in a very simple way. In this way we find out the Local address, Local host information, reuseAddress, and address of the local system in a very simple manner.

Example of TCP/IP client server communication using Socket and ServerSocket class.

(Server socket implementation - 2 Marks, Client implementation - 2 Marks)

//Client Program

```
import java.io.*;
import java.net.*;
public class MyClient
{
public static void main(String args[ ])throws Exception
{
        Socket s= new Socket("127.0.0.1",500);
PrintWriterpw=new PintWriter(s.getOutputStream( ));
pw.println("Hello Server");
pw.flush();
pw.close();
}
}
```

```
//ServerProgram
import java.io.*;
import java.net.*;
public class MyServer
{
public static void main(String args[ ])throws Exception
{
ServerSocketss=new ServerSocket(500);
System.out.println("Before Accept");
        Socket s= ss.accept();
System.out.println("After Accept");
BufferedReaderbr=new BufferedReader(new
InputStreamReader(s.getInputStream()));
         String msg=br.readLine();
System.out.println("Message from client":+msg);
s.close();
}
}
```

Output:

```
C:\jdk1.3\bin> java MyServer
Before Accept
After Accept
Message from client : Hello Server
```

3. Attempt any FOUR of the following: 16

a) Explain following methods with syntaxes:

(i) setMenuBar()

(ii) setFont()

(iii)setBackground()

(iv)setVisible()

(Each correct syntax - ½ Marks, Explanation - ½ Marks)

Ans: (i) setMenuBar()

This method is data member function of Frame class. This method used for to the MenuBar of frame. This method accepts one argument as object of MenuBar class. The syntax of method as follows:

publicvoidsetMenuBar (MenuBarmb)

Example:

```
Frame frm=new Frame("MSBTE");
MenuBarmbr=new MenuBar();
frm.setMenuBar(mbr);
```

(ii) setFont()

This method is member method of Component class. To use a font that you have created, you must select it using setFont(). The syntax of method as follows:

public void setFont(Font f)

Here, 'f' is the object that contains the desired font.

Example:

```
Font font =new Font("Times New Roman",Font.BOLD,12)
setFont(font);
```

(iii) setBackground()

This method is data member function of Component class. To set the background color of window, use setBackground(). The syntax of method as follows:

public void setBackground(Color newColor)

Here, newColor specifies the new color.

Example:

setBackground(Color.red);

iv) setVisible()

This method is data member function of Component class. These method is used for to make the component visible. The syntax of method as follows:

voidsetVisible(booleanvisibleFlag)

The component is visible if the argument to this method is true. Otherwise, it is hidden.

Example:

Frame f=new Frame()

f.setVisible(true);

b) List different layouts and explain border layout with suitable example.

(List - 1 Mark, Explanation - 1 Mark, Example - 2 Marks)

Ans: A Layout Manager automatically arranges your controls within a window by using some type of algorithm. List of Layout Manger are as follows

1. FlowLayout
2. BorderLayout
3. GridLayout
4. CardLayout
5. BoxLayout
6. GroupLayout
7. SpringLayout

BorderLayout

The BorderLayout class implements a common layout style for top-level windows. It has four narrow, fixed-width components at the edges and one large area in the center. The four sides are referred to as north, south, east, and west. The middle area is called the center. Here are the constructors defined by BorderLayout:

BorderLayout( )

BorderLayout(inthorz, intvert)

The first form creates a default border layout. The second allows you to specify the horizontal and vertical space left between components in horz and vert, respectively. BorderLayout defines the following constants that specify the regions:

BorderLayout.CENTER

B orderLayout.SOUTH

BorderLayout.EAST

BorderLayout.WEST

BorderLayout.NORTH

When adding components, you will use these constants with the following form of add( )which idefined by Container:

void add(Component compObj, Object region);

Here, compObj is the component to be added, and region specifies where the componentwill be added.

Example:

Here is an example of a BorderLayout with a component in each layout area:

```
// Demonstrate BorderLayout.
importjava.awt.*;
importjava.applet.*;
importjava.util.*;
/*
<applet code="BorderLayoutDemo" width=400 height=200>
</applet>
*/
public class BorderLayoutDemo extends Applet
 {
        public void init()
         {
        setLayout(new BorderLayout());
        add(new Button("This is across the top."),BorderLayout.NORTH);
        add(new Label("The footer message might go here."),BorderLayout.SOUTH);
        add(new Button("Right"), BorderLayout.EAST);
```

```
        add(new Button("Left"), BorderLayout.WEST);
        String msg = "The reasonable man adapts " +"himself to the world;\\n" +
        "the unreasonable one persists in " +"trying to adapt the world to himself.\\n" +"Therefore
        all progress depends " +"on the Znreasonable man.\\n\\n" +" - George Bernard
        Shaw\\n\\n";
        add(new TextArea(msg), BorderLayout.CENTER);
        }
}
```

c) Explain with example prepared statement object.

(Explanation - 2 Marks, Example - 2 Marks)

Ans: The Prepared statement object:

The PreparedStatement interface extends the Statement interface which gives you added functionality with a couple of advantages over a generic Statement object. This statement gives you the flexibility of supplying arguments dynamically. Creating PreparedStatement Object:

```
PreparedStatementpstmt = null;
try {
  String SQL = "Update Employees SET age = ? WHERE id = ?";
pstmt = conn.prepareStatement(SQL);
  . . .
}
catch (SQLException e) {  . . . }
finally {
  . . .
}
```

All parameters in JDBC are represented by the? symbol, which is known as the parameter marker. You must supply values for every parameter before executing the SQL statement.

The setXXX() methods bind values to the parameters, where XXX represents the Java data type of the value you wish to bind to the input parameter. If you forget to supply the values, you will receive anSQLException.

Each parameter marker is referred to by its ordinal position. The first marker represents position 1, the next position 2, and so forth. This method differs from that of Java array indices, which start at 0.

All of the Statement object's methods for interacting with the database (a) execute(), (b) executeQuery(), and (c) executeUpdate() also work with the PreparedStatement object. However, the methods are modified to use SQL statements that can take input the parameters.

Closing PreparedStatement Object:

Just as you close a Statement object, for the same reason you should also close the PreparedStatement object.

A simple call to the close() method will do the job. If you close the Connection object first it will close the PreparedStatement object as well. However, you should always explicitly close the PreparedStatement object to ensure proper cleanup.

```
PreparedStatementpstmt = null;
try {
  String SQL = "Update Employees SET age = ? WHERE id = ?";
pstmt = conn.prepareStatement(SQL);
  . . .
}
catch (SQLException e) {
  . . .
}
finally {
pstmt.close();
}
```

Example:

```
import java.io.*;
importjava.sql.*;
public class PreparedStatementDemo
  {
        public static void main(String args[])
                {
```

```
                    try
                     {
                            Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                            Connection con=DriverManager.getConnection("jdbc:odbc:dsn");
                            String sql ="update emp set salary =? Where empid=?" ;
                            PreparedStatementps =con.prepareStatement(sql);
                            ps.setInt(1,20000);
                            ps.setInt(2,101);
int  n=ps.executeUpdate();
if(n>0)
System.out.println("Record is updated successfully ");
else
System.out.println("Record is not updated successfully ");
}
catch(Exception e)
{
System.out.println("Exception :"+e);
}
                        }
}
```

d) Write a program to delete a record from student table whose roll no. is 101.

(Assume suitable data for table - Load driver, Create connection - 1 Mark, Execute query-1 Mark, Correct statement -1 Mark, Correct logic - 1 Mark)

Ans:

```
import java.io.*;
importjava.sql.*;
public class Student
 {
        public static void main(String args[])
```

```
        {
            try
                {
                    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                    Connection con=DriverManager.getConnection("jdbc:odbc:dsn");
                    Statement st=con.createStatement();
                    String sql="delete from student where roll_no=101"
                    int r=st.executeUpdate(sql);
if(r>0)
System.out.println("Record is deleted successfully ");
else
System.out.println("Record is not deleted successfully ");
}
catch(Exception e)
{
System.out.println("Exception :"+e);
}
                }
}
```

OR

```
import java.io.*;
importjava.sql.*;
public class Student
 {
      public static void main(String args[])
       {
            try
                {
                    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
                    Connection con=DriverManager.getConnection("jdbc:odbc:dsn");
                    String sql="delete from student where roll_no=?"
```

```
                    PreparedStatementps=con.prepareStatement(sql);
                    ps.setInt(1,101);
                    int r=ps.executeUpdate();
                    if(r>0)
                    System.out.println("Record is deleted successfully ");
                    else
                    System.out.println("Record is not deleted successfully ");
                }
            catch(Exception e)
                {
                    System.out.println("Exception :"+e);
                }
        }
    }
```

e) Design a swing application containing a JCombobox for three flavour as strawberry, vanilla and chocolate.

(Correct syntax - 2 Marks, Correct logic- 2 Marks)

Ans:

```
import java.awt.*;
importjava.awt.event.*;
importjava.swing.*;
/*
<applet  code="JComboBoxDemo.class" width=300 height=100>
</applet> */
    public class JComboBoxDemo extends JApplet implements ItemListener
{
    JLabeljl;
    ImageIconstrawberry, vanilla, chocolate;
    publicvoidinit()
     {
```

```
                Container con=getContentPane();
                con.setLayout(new FlowLayout());
                JComboBoxjc=new JComboBox();
                jc.addItem("strawberry");
                jc.addItem("vanilla");
                jc.addItem("chocolate");
                con.add(jc);
                j1=new JLabel(new ImageIcon("strawberry.gif"));
                con.add(j1);
          }
        public void itemStateChanged(ItemEventie)
        {
                String s=(String)ie.getItem();
                J1.setIcon(new ImageIcon(s+".gif"));
        }
}
```

OR

```
importjava.awt.*;
importjava.swing.*;
/*
<applet  code="JComboBoxDemo" width=300 height=100>
</applet> */
public class JComboBoxDemo extends JApplet
{
publicvoidinit()
        {
                Container con=getContentPane();
                con.setLayout(new FlowLayout());
                JComboBoxjc=new JComboBox();
                jc.addItem("strawberry");
                jc.addItem("vanilla");
```

```
            jc.addItem("chocolate");
            con.add(jc);
        }
    }
```

f) Explain the interfaces of javax. servlet package.

(Any four - 1 Mark each)

Ans: The javax.servlet package contains a number of interfaces and classes that establish the framework in which servlets operate. The following table summarizes the core interfaces that are provided in this package. The most significant of these is Servlet. All servlets must implement this interface or extend a class that implements the interface.

1. Servlet: All servlets must implement the Servlet interface. It declares the init( ), service( ), and destroy( ) methods that are called by the server during the life cycle of a servlet. A method is also provided that allows a servlet to obtain any initialization parameters. It declares life cycle methods for a servlet.

2. ServletConfig: The ServletContext interface is implemented by the server. It enables servlets to obtain information about their environment. Allows servlets to get initialization parameters.

3. ServletContext: The ServletContext interface is implemented by the server. It enables servlets to obtain information about their environment. Enables servlets to log events and access information about their environment.

4. ServletRequest: The ServletRequest interface is implemented by the server. It enables a servlet to obtain information about a client request. Used to read data from a client request.

5. ServletResponse: The ServletResponse interface is implemented by the server. It enables a servlet to formulate a response for a client. Used to write data to a client response.

6. SingleThreadModel : This interface is used to indicate that only a single thread will execute the service( ) method of a servlet at a given time. It defines no constants and declares no methods. If a servlet implements this interface, the server has two options. First, it can create several

instances of the servlet. When a client request arrives, it is sent to an available instance of the servlet. Second, it can synchronize access to the servlet. Indicates that the servlet is thread safe.

4. a) Attempt any THREE of the following: 12

(i) What are listeners? List the names of any two components with their listeners required.
(Explanation - 2 Marks, List -2 Marks)

Ans: A listener is an object that is notified when an event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events. Second, it must implement methods to receive and process these notifications.

The methods that receive and process events are defined in a set of interfaces found in java.awt.event. For example, the MouseMotionListener interface defines two methods to receive notifications when the mouse is dragged or moved. Any object may receive and process one or both of these events if it provides an implementation of this interface.

List of component and there required listener interface

Component	Event class	Listener Interface
Button	ActionEvent	ActionListener
Checkbox	ItemEvent	ItemListener
Choice	ItemEvent	ItemListener
List	ActionEvent, ItemEvent	ActionListener,ItemListener
MenuItem	ActionEvent, ItemEvent	ActionListener,ItemListener
ScrollBar	AdjustmentEvent	AdjustmentListener

(ii) Explain following methods:

1) getInputStream( )

2) accept( )

3) getProtocol( )

4) getLocalHost( )

(Each - 1 Mark)

Ans: 1) getInputStream()

This method is data member function of Object class. The syntax of method as follows:

publicInputStreamgetInputStream() throws IOException

It returns the InputStream associated with the invoking socket. If this socket has an associated channel then the resulting input stream delegates all of its operations to the channel. If the channel is in non-blocking mode then the input stream's read operations will throw an IOException .

Example:

```
Socket client= new Socket("MsbteServer",1079);
InputStream ins=client.getInputStream();
DataInputStream in =new DataInputStream(ins);
```

2) accept()

Waits for an incoming client. This method blocks until either a client connects to the server on the specified port or the socket times out, assuming that the time-out value has been set using the setSoTimeout() method. Otherwise, this method blocks indefinitely. The syntax of this method as follows

public Socket accept() throws IOException

Example:

```
ServerSocketss=new ServerSocket(1079);
Socket s=ss.accept();
```

3) getProtocol()

This method is data member function of URL class. It returns name of protocol used by url. The syntax of this method as follows.

public String getProtocol();

Example:

URL u=new URL("http://msbte.com");

System.out.println("Protocol Name: "+u.getProtocol());

4) getLocolHost()

This method is static data member function of InetAddress class. It returns the InetAddress object that represents the local host.

public static InetAddressgetLocalHost() throws UnknownHostException.

Example:

InetAddress address= InetAddress.getLocalHost();

System.out.println("Address:"+address);

(iii) Describe any four navigational methods for moving record pointer in a Resultset.

(Any four - 1 Mark each)

Ans: Navigation method of resultset

1. public void beforeFirst() throws SQLException

Moves record pointer or cursor to the beginning of the ResultSet that is before first row.

Syntax:

ResultSetObject.beforeFirst();

Example:

rs.beforeFirst();

2. public void afterLast() throws SQLException

Moves record pointer or cursor to the end of the resultset that is after last row.

Syntax:

ResultSetobject.afterLast();

Example:

rs.afterLast();

3. publicboolean first() throws SQLException

Moves record pointer or cursor to the first row of resultset.

Syntax:

ResultSetObject.first();

Example:

rs.first();

4. public void last() throws SQLException

Moves record pointer or cursor to the last row of result set.

Syntax:

RecordSet.last();

Example:

rs.last();

5. public void previous() throws SQLException

Moves record pointer or cursor to the previous row of result set.

Syntax:

int r= RecordSet. previous ();

Example:

rs. previous ();

6. public void next() throws SQLException

Moves record pointer or cursor to the next row of result set.

Syntax:

int r= RecordSet. next();

Example:

rs.next();

(iv) Explain JScrollPane with suitable example.

(Explanation - 2 Marks, Example - 2 Marks)

Ans: A scroll pane is a component that presents a rectangular area in which a component may be viewed. Horizontal and/or vertical scroll bars may be provided if necessary. Scroll panes are implemented in Swing by the JScrollPane class, which extends JComponent. Some of its constructors are shown here:

JScrollPane (Component comp)

JScrollPane (intvsb, inthsb)

JScrollPane(Component comp, intvsb, inthsb)

Here, comp is the component to be added to the scroll pane. vsb and hsb are int constants that define when vertical and horizontal scroll bars for this scroll pane are shown. These constants are defined by the ScrollPaneConstants interface. Some examples of these constants are described as follows:

Constant	Description
HORIZONTAL_SCROLLBAR_ALWAYS	Always provide horizontal scroll bar
HORIZONTAL_SCROLLBAR_AS_NEEDED	Provide horizontal scroll bar, if needed
VERTICAL_SCROLLBAR_ALWAYS	Always provide vertical scroll bar
VERTICAL_SCROLLBAR_AS_NEEDED	Provide vertical scroll bar, if needed

Here are the steps that you should follow to use a scroll pane in an applet:

1. Create a JComponent object.

2. Create a JScrollPane object. (The arguments to the constructor specify thecomponent and the policies for vertical and horizontal scroll bars.)

3. Add the scroll pane to the content pane of the applet.

importjava.awt.*;

```
importjavax.swing.*;
/*
<applet code="JScrollPaneDemo" width=300 height=250>
</applet>
*/
public class JScrollPaneDemo extends JApplet
 {
public void init()
 {
// Get content pane
Container contentPane = getContentPane();
contentPane.setLayout(new BorderLayout());
// Add 400 buttons to a panel
JPaneljp = new JPanel();
jp.setLayout(new GridLayout(20, 20));
int b = 0;
for(int i = 0; i < 20; i++) {
for(int j = 0; j < 20; j++) {
jp.add(new JButton("Button " + b));
++b;
}
}
// Add panel to a scroll pane
int v = ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED;
int h = ScrollPaneConstants.HORIZONTAL_SCROLLBAR_AS_NEEDED;
JScrollPanejsp = new JScrollPane(jp, v, h);
// Add scroll pane to the content pane
contentPane.add(jsp, BorderLayout.CENTER);
  }
}
```

b) Attempt any ONE of the following: 06

(i) Design a menu as 'File' and 'Edit'. Add option under 'File' menu as 'New' and 'Close' and under 'Edit' as 'Copy'. Display appropriate message when any of the menu item is selected. ***(Event handling - 2 Marks, Correct Syntax - 2 Marks, Correct Logic - 2 Marks)***

Ans:

```
import java.awt.*;
importjava.awt.event.*;
public class MenuDemo extends Frame implements ActionListener
{
TextField t1;
publicMenuDemo()
        {       super();
                MenuBarmbr=new MenuBar();
                setMenuBar(mbr);
                Menu file=new Menu("File");
                Menu edit=new Menu("Edit");
                MenuItem new1=new MenuItem("New");
                MenuItem close1=new MenuItem("Close");
                MenuItem copy1=new MenuItem("Copy");
                file.add(new1);
                file.add(close1);
                edit.add(copy1);
                mbr.add(file);
                mbr.add(edit);
                t1=new TextField();
                add(t1);
                file.addActionListener(this);
                edit.addActionListener(this);
                new1.addActionListener(this);
                close1.addActionListener(this);
                copy1.addActionListener(this);
```

```
        }
        public void actionPerformed(ActionEventae)
        {
                String label=ae.getActionCommand();
                t1.setText(label+" "+" Menu is selected");
        }
        public static void main(String args[])
        {
                MenuDemo m=new MenuDemo();
                m.setVisible(true);
                m.setSize(200,200);
                m.setLocation(20,20);
        }
}
```

(ii) Explain use of following methods with correct syntaxes:

1) **doGet( )**
2) **getInitParameter( )**
3) **getWriter( )**

(Syntax - 1 Mark, Use - 1 Mark)

Ans: 1)doGet()

Syntax:

voiddoGet(HttpServletRequestreq, HttpServletResponse res) throws IOException,ServletException

Use:

This method use to handle HTTP GET request. The servlet is invoked when a form on web pages is submitted.

2) getInitParameter()

Syntax:

String getInitPrameter(String name)

Use:

This method return the parameter value for given parameter name.

3)getWriter()

Syntax:

PrintWritergetWriter() throw IOException

Use:

It returns a PrintWriter that can be used to write character data to the response. An IllegalStateException is thrown if getOutputStream() has already been invoked for this request.

5. Attempt any TWO of the following: 16

a) Write a program to accept an employee id from the user and modify salary of that employee by 10% raise. Assume suitable structure of emp table.

(Eestablishing connection - 2 Marks, accept input from user - 2 Marks, Correct logic – 2 Marks, correct syntaxes - 2 Marks)

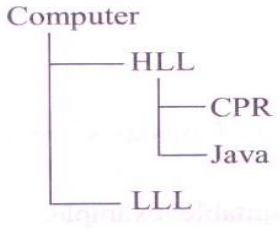
Ans:

```
import java.sql.*;
import java.io.*;
public class prepare
{
public static void main(String args[])
{
int id=0,sal=0;
String name=" ";
Connection con;
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
BufferedReader b=new BufferedReader(new InputStreamReader(System.in));
System.out.println("enter emp id");
id=Integer.parseInt(b.readLine());
}
catch(Exception e)
```

```
{}
try
{
  //dn is DSN
String s="jdbc:odbc:dn";
con=DriverManager.getConnection(s," "," ");
PreparedStatementst=con.prepareStatement("update emp set sal = sal+(sal*0.1) where empno
=?");
st.setInt(1,id);
st.executeUpdate();
System.out.println("Record is upadated successfully");
con.close();
}
catch (Exception ee)
{
        System.out.println(ee);
}
}
}
```

Note: Answer up to update to be considered as valid complete answer.

b) Design swing for following tree structure and show selected name in JLabel.



(use of correct packages - 1 Mark, Applet tag- 1 Mark, Correct Logic - 3 Marks, correct syntaxes - 3 Marks)

Ans:

```
import javax.swing.*;
importjavax.swing.event.*;
importjavax.swing.tree.*;
import java.awt.*;
/*<applet code="tree" height=150 width=150>
</applet>*/
public class tree extends JApplet
{       JTree tree;
        JLabeljl;
        public void init()
        {
                DefaultMutableTreeNode t= new DefaultMutableTreeNode("Computer");
                DefaultMutableTreeNode x=new DefaultMutableTreeNode("HLL");
                t.add(x);
                DefaultMutableTreeNode a=new  DefaultMutableTreeNode("CPR");
                x.add(a);
                DefaultMutableTreeNode b= new DefaultMutableTreeNode ("Java");
                x.add(b);
                DefaultMutableTreeNode  y =new DefaultMutableTreeNode ("LLL");
                t.add(y);
                Container cnt = getContentPane();
                cnt.setLayout(new FlowLayout());
                tree =new JTree(t);
                JScrollPanejs = new JScrollPane(tree);
                jl=new JLabel();
                cnt.add(js);
                cnt.add(tree);
                cnt.add(jl);
                tree.addTreeSelectionListener(new TreeSelectionListener()
                {       public void valueChanged(TreeSelectionEvent e)
```

```
                {
                        String st="status:"+e.getPath();
                        jl.setText(st);
                }
        });
} }
```

c) What are cookies? Design a servlet to store a user name in a cookie and retrieve it.

(Cookies explanation - 2 Marks, Use of appropriate packages - 1 Mark,Servlet code - 3 Marks, Html code - 2 Marks)

Ans: A cookie is stored on client and contains state information. A cookie can store user'sname, address and other information. Servlet can add information in the form of cookie with addCookie() method of HTTPServletResponse interface. This data gets included in header of HttpResponse that is sent to the browser.Information of each cookie includes name, value, expiration date, domain, path etc. Expiration date determines when the cookie is deleted from user's machines.If the domain and path are matching with the url then cookie is supplied to the web server otherwise not.

Program:

```
import java.io.*;
importjavax.servlet.*;
importjavax.servlet.http.*;
public class cookietest extends HttpServlet
{
public void doGet(HttpServletRequestreq,HttpServletResponse res) throws
IOException,ServletException
        {
                PrintWriterpr;
                pr=res.getWriter();
                res.setContentType("text/html");
                String uid= getParameter("username");
                Cookie c1= new Cookie("username",uid);
        res.addCookie(c1);
```

```
        Cookie[] c=req.getCookies();
                if ((c != null) && (c.length> 0))
                {
                        for(int i=0;i<c.length;i++)
                        {
                                Cookie c3=c[i];
                                String nm=c3.getName();
                                String vl=c3.getValue();
                                pr.println("<big>"+nm+"="+vl);
                        }
                }
                else
                pr.print("<big>No cookies available");

        }
// html code  to display form
<html>
<head></head>
<body>
<form method="post" action="http://localhost:8080/examples/servlet/cookietest">
Enter Username : <br>
<input type="text" name="username"><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

6. Attempt any FOUR of the following: 16

a) Write a program to display ip addresses of the domain name www.google.com.

(Correct Logic - 2 Marks, Correct Syntaxes - 2 Marks)

Ans:

```
import java.net.*;
public class NetDemo
{
public static void main(String args[])
{
InetAddressip = InetAddress.getByName("www.google.com");
System.out.println("Address of www.google.com : " + ip);
}
}
```

b) Explain with correct syntax:

(i) executeUpdate( )

(ii) getConnection( )

(For each method, correct syntax – 1 Mark, Explanation – 1 Mark)

Ans: (i) executeUpdate()

It runs the given SQL statement, which can be an INSERT, UPDATE, or DELETE statement.

Syntax:

Public int execute Update(String sql)

Where sql is a String that contains the SQL statement.

Return type 'int' indicates the number of rows affected

Example:

```
Statement st=con.createStatement("update emp set sal = sal+(sal*0.1) where empno =101");
st.executeUpdate();
```

(ii) getConnection()

This method is used to establish a connection with the data source.**DriverManager.getConnection()** method is used to create a connection object. The most

commonly used form of getConnection() requires you to pass a database URL, a username, and a password:

Syntax:

Connection getConnection(String url)

whereurl is String showing the JDBC driver selected.

Example:

String s="jdbc:odbc:dn";

Connection con=DriverManager.getConnection(s," "," ");

c) Write any two construction of:

(i) JButton

(ii) JTree

(For each constructor - 1 Mark)

Ans: (i) JButton

Constructors are:

1) **JButton**() - Creates a button with no set text or icon.

2)**JButton**(Icon icon) - Creates a button with an icon.

3) **JButton**(String text) - Creates a button with text.

4) **JButton**(String text, Icon icon) - Creates a button with initial text and an icon.

[Note: Any two constructors from above list can be considered.]

(ii) JTree

Constructors are :

1) JTree()

Returns a JTree with a sample model.

2) JTree(Hashtable<?,?> value)

Returns a JTree created from a Hashtable which does not display with root.

3) JTree(Object[] value)

Returns a JTree with each element of the specified array as the child of a new root node which is not displayed.

4) JTree(TreeModel newModel)

Returns an instance of JTree which displays the root node -- the tree is created using the specified data model.

5) JTree(TreeNode root)

Returns a JTree with the specified TreeNode as its root, which displays the root node.

6) JTree(TreeNode root, boolean asksAllowsChildren)

Returns a JTree with the specified TreeNode as its root, which displays the root node and which decides whether a node is a leaf node in the specified manner.

7) JTree(Vector<?> value)

Returns a JTree with each element of the specified Vector as the child of a new root node which is not displayed.

[Note:Any two constructors from above list can be considered.]

d) Write life cycle methods of servlet provided by Http servlet class.

(For each method of servlet life cycle- 3 Marks, doPost() - 1 Mark, doGet() – 1 Mark of HTTP Servlet class)

Ans: Servlet life cycle methods:

1) init()

2) service()

3) destroy()

HttpServlet class also has these three life cycle methods, but after service() method call, it brances to doGet() of doPost() according to method used in <Form> tag of html page.

doPost() is used when <form> tag uses post method for redirecting data to servlet.

doGet() is used when <form> tag uses get method for redirecting data to servlet in the form of Querystring.

1. Creation and initialization

The container first creates the servlet instance and then executes the init() method. init() can be called only once in its life cycle by the following ways:
a) For the first time only in its life cycle, just before the service() is invoked.
b) Server administrator can request for the initialization of a servlet directly.

2. Execution of service

Whenever a client requests for the servlet, everytime the service() method is invoked during its

life cycle. From service() then it is branched to the doGet() or doPost() methods for a HttpServlet. The service() method should contain the code that serves the Servlet purpose.

3. Destroy the servlet

destroy() method is invoked first, then Servlet is removed from the container and then eventually garbage collected. destroy() method generally contains code to free any resources like jdbc connection that will not be garbage collected.

e) What is session tracking? Explain.

(Session tracking explanation - 2 Marks, Example - 2 Marks)

Ans: Session provides mechanism for saving state info so that the info can be collected from several interactions between a browser and a server. A session can be created by getSession() of HttpServletRequest. A session state is shared among all the servlets that are associatedwith a particular client.

Session Tracking is a way to maintain state (data) of an user. It is also known as session management in servlet. Every user of a site is associated with javax.servlet.http package. HttpSession object can be used to store all retieved information about the user visiting the site. You can save any set of arbitary java objects in a session object. A servlet uses its request object to retrieve current HttpSessio object. To add data to HttpSession object setAtrribute() method can be used. To get value of session attribute, getAttribute() method is used.

//program to count number of times the page is visited in a session

```
import java.io.*;
importjavax.servlet.*;
importjavax.servlet.http.*;
importjava.util.*;
public class sessiontracker extends HttpServlet
{
public void doGet(HttpServletRequestreq, HttpServletResponse res)

throwsServletException, IOException
        {
                res.setContentType("text/html");
```

```
            PrintWriter out = res.getWriter();
            HttpSession session = req.getSession();
            Integer count = (Integer)session.getAttribute("tc");
            if (count == null)
            count = new Integer(1);
            else
            count = new Integer(count.intValue() + 1);
            session.setAttribute("tc", count);
            out.println("<big>You have visited :" + count + " times");
            out.close();
        }
}
```

[Note: Anyother example can be considered.]