Adapter class

The Adapter class provides the default modification of all methods of an interface; we don't need to modify all the methods of the interface so we can say it reduces coding burden. Sometimes or often we need a few methods of an interface. For that the Adapter class is very helpful since it already modifies all the methods of an interface and by implementing the Adapter class, we only need to modify the required methods.

The following examples contain the following Adapter classes:

- ContainerAdapter class.
- KeyAdapter class.
- FocusAdapter class.
- WindowAdapter class.
- MouseAdapter class.
- ComponentAdapter class.
- MouseMotionAdapter class.

Examples

The frame created by the programmer and displayed on monitor comes with title bar and three icons(buttons) on the title bar

I. Minimize button
II. Maximize button
III. Close button

The first two button works implicitly and the close button does not work and requires extra code to close the Frame.

There are three different styles exist to close the Frame are:

I. WindowListener
II. WindowAdaptor
III. Anonymous inner class

Example of WindowAdapter class

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

class windowAdapter1 extends Frame
{
public windowAdapter1()
{
super("MyFrame");
//addWindowListener(new myClass());  or
myClass m=new myClass();
addWindowListener(m);
setSize(300,400);
setVisible(true);
}
public static void main(String args[])
{
new windowAdapter1();
}
}

class myClass extends WindowAdapter
{
public void windowClosing(WindowEvent evt)
{
System.exit(0);
}


}
```

Anonymous class

If you write a class without name then it's known as Anonymous class.

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

class myFrame extends Frame
{
public myFrame()
{
super("My App");
addWindowListener(new WindowAdapter()
{
		public void windowClosing(WindowEvent evt)
		{
		System.exit(0);
		}
});
setSize(300,400);
setVisible(true);
}
public static void main(String args[])
{
new myFrame();
}


}
```

Q: create a LOGIN form that contains only two fields, i.e., username and password using frame and awt components. We also create a button for performing the action. After submitting the login form, the underlying code of the form checks whether the credentials are authentic or not to allow the user to access the restricted page. If the users provide authentic credentials, print "Login Successful" otherwise print "Enter valid user id and password".

```
import java.sql.*;
import java.awt.event.*;
import java.awt.*;
public class Login extends Frame implements ActionListener,WindowListener{
   Label lbluid,lblpwd;
   TextField txtuid,txtpwd;
   Button btnSignIn;
   Panel p1;
   public Login()
   {
      super("LOGIN FORM");
      lbluid=new Label("Enter the User ID:");
      lblpwd=new Label("Enter the Password:");
      txtuid=new TextField(20);
      txtpwd=new TextField(20);
```

```
     txtpwd.setEchoChar('*');
     btnSignIn=new Button("SignIn");
     p1=new Panel();
     p1.setBackground(Color.red);
     add(p1);
     p1.add(lbluid); p1.add(txtuid);
     p1.add(lblpwd); p1.add(txtpwd);
     p1.add(btnSignIn);
     btnSignIn.addActionListener(this);
     addWindowListener(this);

   }

   public void actionPerformed(ActionEvent evt)
   {
     Object obj=evt.getSource();
     if(obj==btnSignIn)
     {
       String query="select * from login where uid='"+txtuid.getText()+"' and
pwd='"+txtpwd.getText()+"'";
       try{
         Class.forName("org.apache.derby.jdbc.ClientDriver");
         Connection
conn=DriverManager.getConnection("jdbc:derby://localhost:1527/bns","bns","bns");
         Statement stmt=conn.createStatement();
         ResultSet rs=stmt.executeQuery(query);
         boolean status= rs.next();
         if(status)
         {
           DemoMenu dc=new DemoMenu();
           dc.setSize(400,400);
           dc.setVisible(true);
           dc.setLocation(200, 150);
         }
         else
         {
           DemoChoice dc=new DemoChoice();
           dc.setSize(400,400);
           dc.setVisible(true);
           System.out.println("Enter the valid user id or password");
         }

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

     }
   }

   public static void main(String args[])
   {
     Login l=new Login();
```

```
        l.setSize(600,600);
        l.setVisible(true);
        l.setLocation(300, 100);
    }

    @Override
    public void windowOpened(WindowEvent e) {

    }

    @Override
    public void windowClosing(WindowEvent e) {
        System.exit(0);
    }

    @Override
    public void windowClosed(WindowEvent e) {

    }

    @Override
    public void windowIconified(WindowEvent e) {

    }

    @Override
    public void windowDeiconified(WindowEvent e) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of
generated methods, choose Tools | Templates.
    }

    @Override
    public void windowActivated(WindowEvent e) {

    }

    @Override
    public void windowDeactivated(WindowEvent e) {

    }


}
```