Appendix A

Listing 5.1: Using data-driven test data

```
File f1=new java.io.File("c://log.txt");
Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
c = DriverManager.getConnection( "jdbc:odbc:DataDriven", "", "" );
stmnt = c.createStatement();
String query = "select * from [TestCases$] ;";
ResultSet rs = stmnt.executeQuery( query );
    while( rs.next())
        {
          String  testcase= rs.getString( "Test Case" );
           int        N1= rs.getInt("Number1");
          int        N2= rs.getInt("Number2");
          String   Op= rs.getString( "Operator" );
          float    Ex= rs.getFloat( "Expected" );
         //Use the test data for   testing
        }
```

Listing 5.2: java simple Calculator example

```
Public class SimpleCalculator{
    public int sum(int x,int y){
      int z=x+y;
      return z;
  }
public int sub(int x,int y){
      int z=x-y;
      return z;
  }
public int multi(int x,int y){
   int z=x*y;
    return z;
  }
public int Devision(int x, int y){
     int z=x/y;
     return z;
  }}
```

Appendix B

Listing 5.3: simple Calculator Data-driven test example

```
package javaapplication1;
import org.junit.*;
import java.sql.*;
public class SimpleCalculatorTest {
   public SimpleCalculatorTest() {}
   public void testSub() {
      int x = 0;
      int y = 0;
       SimpleCalculator instance = new SimpleCalculator();
       MyDriverScript drs= new MyDriverScript();
       String LogLevel="Info";
       int ntc=0,pass=0,fail=0;
       MylogLib log =new
MylogLib("D:\\last_Desertation\\practical_work\\Last_Excels\\logs\\SimpleCalculator_Data_log_7_2.doc");
     try
       {
        log.write("Info   Start Testing The     ");
         String aFile = "SimpleCalculator_Data.xls";
         String aPath = "D:\\last_Desertation\\practical_work\\Last_Excels\\";
         ResultSet rs =drs.getData(aFile,aPath);
        log.write("Info   Enviromment is Ready    ");
        while( rs.next() )
          {
            String testcase= rs.getString( "Test Case" );
            x= rs.getInt("Number1");
            y= rs.getInt("Number2");
            String Op= rs.getString( "Operator" );
            int expResult= rs.getInt( "Expected" );
                     int result;
            if (Op.charAt(0)=='-')
                {
  log.write(LogLevel+"   "+testcase+"   Testing  "+x+"        "+Op+"        "+y+"       ="+expResult+" ");
                 ntc++;
                 result = instance.sub(x, y);
                 assertEquals(expResult, result);
                  if(expResult== result)
                  {
                     log.write("Pass   "+testcase+"   is Done Correctly  Expected
result="+expResult+"=Actual Result="+result+"");
                    pass++;
                  }
                  else
                  {
                    log.write("Fail   "+testcase+"   is Fail  Expected resultt="+expResult+" but
ActualResult="+result+"");
                    fail++;
                  }
                }
            if (Op.charAt(0)=='+')
                {
                  log.write(LogLevel+"   "+testcase+"   Testing  "+x+"        "+Op+"        "+y+"
="+expResult+" ");
                 ntc++;
                  result = instance.sum(x, y);
```

```
                if(expResult== result)    {
                    log.write("Pass   "+testcase+"   is Done Correctly  Expected
resultt="+expResult+"=Actual Result="+result+"");
                    pass++;
                  }
                  else  {
                    log.write("Fail   "+testcase+"   is Fail  Expected result="+expResult+" but
ActualResult="+result+"");
                    fail++;
                  }   }
           if (Op.charAt(0)=='*'){
                 log.write(LogLevel+"   "+testcase+"   Testing  "+x+"        "+Op+"        "+y+"
="+expResult+" ");
                  ntc++;
                  result = instance.multi(x, y);
                  assertEquals(expResult, result);
                 if(expResult== result)
                  {
                    log.write("Pass   "+testcase+"   is Done Correctly  Expected
result="+expResult+"=Actual Result="+result+"");
                    pass++;
                  }
                  else{
                    log.write("Fail   "+testcase+"   is Fail  Expected resultt="+expResult+" but
ActualResult="+result+"");
                    fail++;
                  }   }
             if (Op.charAt(0)=='/')
               {
                  log.write(LogLevel+"   "+testcase+"   Testing  "+x+"        "+Op+"        "+y+"
="+expResult+" ");
                 ntc++;
                  result = instance.Devision(x, y);
                  assertEquals(expResult, result);
                   if(expResult== result)
                  {
                       log.write("Pass   "+testcase+"   is Done Correctly  Expected
resultt="+expResult+"=Actual Result="+result+"");
                       pass++;
                  }
                  else
                  {
                     log.write("Fail   "+testcase+"   is Fail  Expected result="+expResult+" but
ActualResult="+result+"");
                    fail++;
                  }} }    }
    catch( Exception e )
      {}
      finally
       {
        float m,n;
        m=Float.valueOf(pass);
        n=Float.valueOf(ntc);
              try{
          log.write("Info   Test Is done with "+pass+" out of "+ntc+" precentage="+(m/n)*100+"%");
         }
       catch( Exception e )
         { }     } }}
```

Appendix C

Listing 5.4: java code for the login code

```
package javaapplication1;
public class login extends javax.swing.JFrame {
    public login() {initComponents();}
    private void initComponents() {
        jTextField1 = new javax.swing.JTextField();jButton1 = new javax.swing.JButton();
        jPasswordField1 = new javax.swing.JPasswordField();  jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
 setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("User Login"); setResizable(false);jButton1.setText("login");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
         public void actionPerformed(java.awt.event.ActionEvent evt)
        jButton1ActionPerformed(evt);}});  jLabel1.setText("User Name") jLabel2.setText("password");
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);layout.setHorizontalGroup(layout.createParallelGroup(javax.swin
g.GroupLayout.Alignment.LEADING).addGroup
(layout.createSequentialGroup()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING).addGroup(lay
out.createSequentialGroup().addGap(32, 32, 32)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING).addCompone
nt(jLabel1,javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE, 67,
javax.swing.GroupLayout.PREFERRED_SIZE)).addPreferredGap(javax.swing.LayoutStyle.Compone
ntPlacement.RELATED).addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING).addComponent(jPasswordField1, javax.swing.GroupLayout.PREFERRED_SIZE, 155,
javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 155,
javax.swing.GroupLayout.PREFERRED_SIZE))).addGroup(layout.createSequentialGroup().addGap(8
5, 85, 85).addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 94,
javax.swing.GroupLayout.PREFERRED_SIZE))).addContainerGap(73, Short.MAX_VALUE)) );
layout.setVerticalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING).
addGroup(layout.createSequentialGroup().addContainerGap()
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING).addCompone
nt(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 17,
javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE).addCompon
ent(jPasswordField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE).addComponent(jLabel2)).addPreferredGap(javax.swin
g.LayoutStyle.ComponentPlacement.RELATED).addComponent(jButton1).addContainerGap(22,
Short.MAX_VALUE)));pack();}
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    if(jTextField1.getText().equalsIgnoreCase("ali"))   {
      if(jPasswordField1.getText().equalsIgnoreCase("123")){
           System.out.println("Wellcome  ali");  }} }
public static void main(String args[]) {
  java.awt.EventQueue.invokeLater(new Runnable() {
  public void run() {new login().setVisible(true);} });}
public void setuser(String x,String y){
    jTextField1.setText(x);   jPasswordField1.setText(y);
   jButton1.doClick();}
    private javax.swing.JButton jButton1;  private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;   private javax.swing.JPasswordField jPasswordField1;
    private javax.swing.JTextField jTextField1;}
```

Appendix D

Listing 5.5: Using keyword-driven test data for login

```
package javaapplication1;
import org.junit.*;
import static org.junit.Assert.*;
import java.sql.*;
public class loginTest {
   public loginTest() { }
   @BeforeClass
   public static void setUpClass() throws Exception {}
   @AfterClass
   public static void tearDownClass() throws Exception {}
   @Before
   public void setUp() {}
   @After
   public void tearDown() {}
   @Test
   public void testMain() {
      Connection c = null;
       Statement stmnt = null;
         String x = "";
         String y = "";
         String[] arg =new String[10];
       login instance = new login();
        try
          {
            Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
            c = DriverManager.getConnection( "jdbc:odbc:loginkeyword", "", "" );
            stmnt = c.createStatement();
            String query = "select * from [TestCases$] ;";
            ResultSet rs = stmnt.executeQuery( query );
              int m=0,n=0,result=0,expResult=1;
          while( rs.next() )
                {
                    String testcase= rs.getString("userkeyword");
                      System.out.println(" Testing Login Data ");
                      rs.next() ;
                      x=rs.getString("keyword");
                                rs.next();x="ali";
                                rs.next();y="123";
                                rs.next();
                                 instance.setuser(x,y);
                                 System.out.println("testing Complete");
                }
        } catch( Exception e ){
             System.err.println( e );
          }
      finally
      { try{
            stmnt.close();c.close();
          } catch( Exception e )
{System.err.println( e ); }}}}
```

Appendix E

Listing 5.6: Using keyword-driven test data for calculator

```
package javaapplication1;
import org.junit.*;
import static org.junit.Assert.*;
import java.io.*;
import java.sql.*;
public class NewClacGUITest
 {
   public NewClacGUITest() { }
   public void testMain() {
      System.out.println("main");
      String[] args = null;
          Connection c = null;
       Statement stmnt = null;
       NewClass instance = new NewClass();
         instance.main(args);
       try  {
           //File logfile=new File("Calulator.log"");
           String loglevel="info";
           Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
        c = DriverManager.getConnection( "jdbc:odbc:GUI_CALAC_DATA", "", "" );
        stmnt = c.createStatement();
       String query = "select * from [TestCases$] ;";
           ResultSet rs = stmnt.executeQuery( query );
           String testcase="";
           float num1,num2;
           String  op1="";
           float expectedResult;
           float ResultFromApplication;
            while( rs.next() )   {
                    testcase= rs.getString("TestCase");
                     instance.num1=rs.getFloat("Number1");
                     instance.op=rs.getString("Operator").charAt(0);
                       System.out.println(instance.op);
                     instance.num2=rs.getFloat("Number2");
                     expectedResult=rs.getFloat("Expected");
                     instance.flag=true;
                     instance.Calctxt.setText(String.valueOf(instance.num2));
                    System.out.println("Start testing "+testcase+"    "+instance.num1+" "+instance.op+"
"+instance.num2+" Expected ="+expectedResult);
                     instance.jButton10.doClick();
                  if (expectedResult==instance.result)  {
                       System.out.println("testing Complete");
                       System.out.println("Start testing "+instance.num1+" "+instance.op+"
"+instance.num2+" Expected ="+expectedResult);
                      }
                      else
       System.out.println("testing not Complete result="+instance.result+"");
                  }   }
        catch( Exception e ){
             System.err.println( e );
         }
      finally{
        try{ stmnt.close(); c.close();
        }catch( Exception e ){
           System.err.println( e );
        } }  }}
```

Appendix F

Listing 6.1: code for data-driven Calculator Example

```
package javaapplication1;
import org.junit.*;
import java.io.*;
import java.sql.*;
public class NewClacGUITest
 {
   public NewClacGUITest() { }
   public void testMain() {
       System.out.println("main");
       String[] args = null;
           Connection c = null;
        Statement stmnt = null;
        NewClass instance = new NewClass();
          instance.main(args);
        try  {
            String loglevel="info";
            Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
         c = DriverManager.getConnection( "jdbc:odbc:GUI_CALAC_DATA", "", "" );
         stmnt = c.createStatement();
        String query = "select * from [TestCases$] ;";
            ResultSet rs = stmnt.executeQuery( query );
            String testcase="";
            float num1;
            float num2;
            String  op1="";
            float expectedResult;
            float ResultFromApplication;
             while( rs.next() )  {
                    testcase= rs.getString("TestCase");
                     instance.num1=rs.getFloat("Number1");
                     instance.op=rs.getString("Operator").charAt(0);
                       System.out.println(instance.op);
                     instance.num2=rs.getFloat("Number2");
                     expectedResult=rs.getFloat("Expected");
                     instance.flag=true;
                     instance.Calctxt.setText(String.valueOf(instance.num2));
                    System.out.println("Start testing "+testcase+"    "+instance.num1+" "+instance.op+"
"+instance.num2+" Expected ="+expectedResult);
                     instance.jButton10.doClick();
                  if (expectedResult==instance.result)
                      {
                       System.out.println("testing Complete");
                       System.out.println("Start testing "+instance.num1+" "+instance.op+"
"+instance.num2+" Expected ="+expectedResult);
                      }else
        System.out.println("testing not Complete result="+instance.result+"");
                  }   }
         catch( Exception e ){
              System.err.println( e );
          }
       finally{
          try{ stmnt.close(); c.close();
          }catch( Exception e ){
             System.err.println( e );
          } }    }}
```

Appendix G

Listing 6.2: keyword-driven for Calculator examples

```
package javaapplication1;
import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;
import static org.junit.Assert.*;
import java.io.*;
import java.sql.*;
public class NewClassTest {
    public NewClassTest() { }
    public static void setUpClass() throws Exception { }
    public static void tearDownClass() throws Exception {}
    public void setUp() {}
    public void tearDown() {}
    public void testMain() {
        System.out.println("main");
        String[] args = null;
        Connection c = null;
        Statement stmnt = null;
        NewClass instance = new NewClass();
        instance.main(args);
         boolean myflag=false;
         char    c1;
          try
            {
              //File logfile=new File("Calulator.log"");
              String loglevel="info";
              Class.forName( "sun.jdbc.odbc.JdbcOdbcDriver" );
              c = DriverManager.getConnection( "jdbc:odbc:Key_Calc_Gui", "", "" );
              stmnt = c.createStatement();
              String query = "select * from [TestCases$] ;";
              ResultSet rs = stmnt.executeQuery( query );
              String testcase="";
              float expectedResult=0;
              String  key="";
              int n=1;
               while( rs.next() )
                   {
                    testcase= rs.getString("TestCase");

                    if(!(testcase.equalsIgnoreCase("&")))
                      {
                        System.out.println(" Start Testing   Keyword  "+testcase+" ");
                      }
                        key =rs.getString("keyword");
                          if (key.equalsIgnoreCase("input"))
                            {
                               if(myflag)
                                 {
                                   instance.num2=rs.getFloat("Argument");
                                    myflag=false;
                                 }
                              else
                                  {
                                   instance.num1=rs.getFloat("Argument");
```

```
                    myflag=true;
                   }
                 }
               if (key.equalsIgnoreCase("push"))
                  {
                     c1=rs.getString("Operator").charAt(0);
                     if(c1 =='+')

                        {
                         instance.op='+';
                        }
                     if(c1 =='-')   {
                         instance.op='-';
                        }
                     if(c1 =='*')    {
                         instance.op='*';
                        }
                     if(c1 =='/')   {
                         instance.op='/';//;(char)c1;
                        }
                    /*   if((myflag))
                       {
                         instance.flag=true;
                         instance.Calctxt.setText(String.valueOf(instance.num2));
                         instance.jButton10.doClick();
                         instance.num1=instance.result;
                         myflag=true;
                        }*/
                   }
               if (key.equalsIgnoreCase("ckeck"))
                  {
                     expectedResult=rs.getFloat("Argument");
                     instance.flag=true;
                     float  oldn1=instance.num1;
                     instance.Calctxt.setText(String.valueOf(instance.num2));
                      instance.jButton10.doClick();
                   if (expectedResult==instance.result)
                     {
                      System.out.println("Succesful Test: "+ oldn1+"   "+instance.op+"
"+instance.num2+" Expected ="+expectedResult+" The REAl Result="+instance.result);
                     }
                    else
                         System.out.println("Not Succesful Test: "+instance.num1+" "+instance.op+"
"+instance.num2+" Expected ="+expectedResult+" The REAl Result="+instance.result);
                  }    }   }
         catch( Exception e ){
             System.err.println( e );
          }
       finally {
          try
          {
            stmnt.close();
            c.close();
          }
        catch( Exception e )    {
            System.err.println( e );
           }
      }
   }}
```

Appendix H

Listing 6.3: Test keyword-driven web Example

```
package javaapplication1;
package javaapplication1;
import java.net.*;
import java.io.*;
import java.util.*;
public class KeywordWebTesting {
  public static void main (String[] args)
  {
  long n,NOW1,NOW=new Date().getTime();
  System.out.println(new java.util.Date()+" Driver |  info |  Start The test Excution");
        String p="sudan";
        String fr="yfp-t-501";
        String toggle="1";
        String cop="mss";
        String ei="UTF-8";
      try{
String
x="http://search.yahoo.com/search?p="+p+"&fr="+fr+"&toggle="+toggle+"&cop="+cop+"&ei="+ei+
"";
          System.out.println(new java.util.Date()+" Driver |  Trace  |  Lunch"+x);
          URLConnection urlConn =
           new URL(x).openConnection();
            InputStream in = urlConn.getInputStream();
             byte buf[] = new byte[4096];
             int nSize = in.read(buf);
            String Ck="";
           System.out.println(new java.util.Date()+" Driver |  info |  Start The test Excution");
         if(nSize>=0)
           {
              NOW1 =new Date().getTime();
              n=NOW1-NOW;
            System.out.println(new java.util.Date()+" Driver |  Trace  |  trace    Successful In "+new
Date(n).getSeconds()+" second ");
            while(nSize>=0)
                 {
                   Ck=new String(buf,0,nSize);
                  // System.out.print(Ck);
                   nSize = in.read(buf);
                 }
          System.out.println("Succesfully Ended In "+new java.util.Date(n)+" second ");
            }
          else
            {
  System.out.println(new java.util.Date()+" Driver |  Trace  |  trace Not  Successful ");
  System.out.println("Not Connected  ");
            }
        System.out.print("\r\n");
       }
     catch(Exception e)
      {
       System.out.println("Test Is Endede beouse of Exception: "+e.getMessage());
      }
      long NOW12 =new Date().getTime();
      long  n1=NOW12-NOW;
   System.out.println("Testing Duration Is ========"+new Date(n1).getSeconds());
  }}
```