

Java Swing Tutorial

Building a Java Swing Application

Prerequisite for Building java swing application using Window builder:

Install Eclipse if you already do not have it on your system.

(Note : for installing eclipse go to the following web site URL and follow the installation instructions

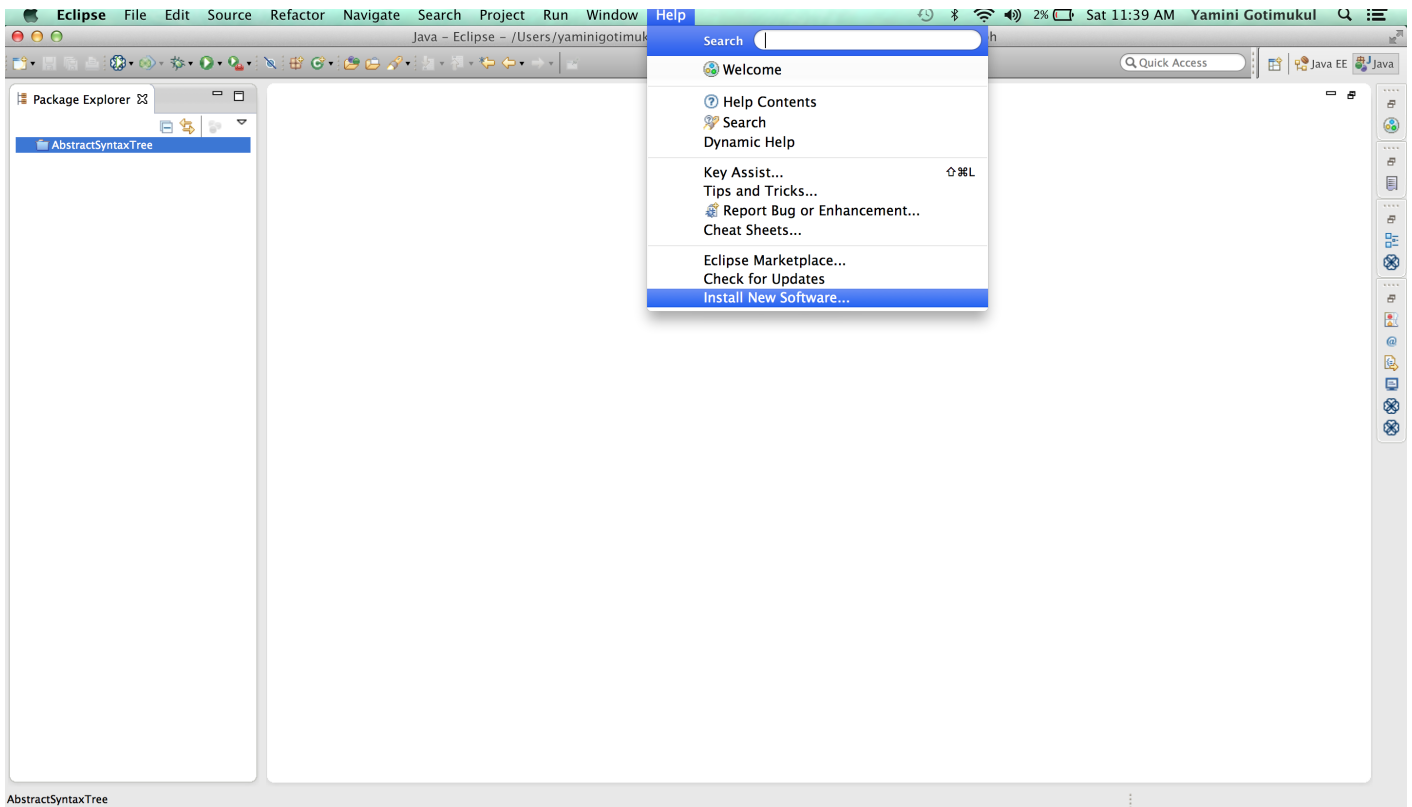
<http://www.eclipse.org/downloads/>)

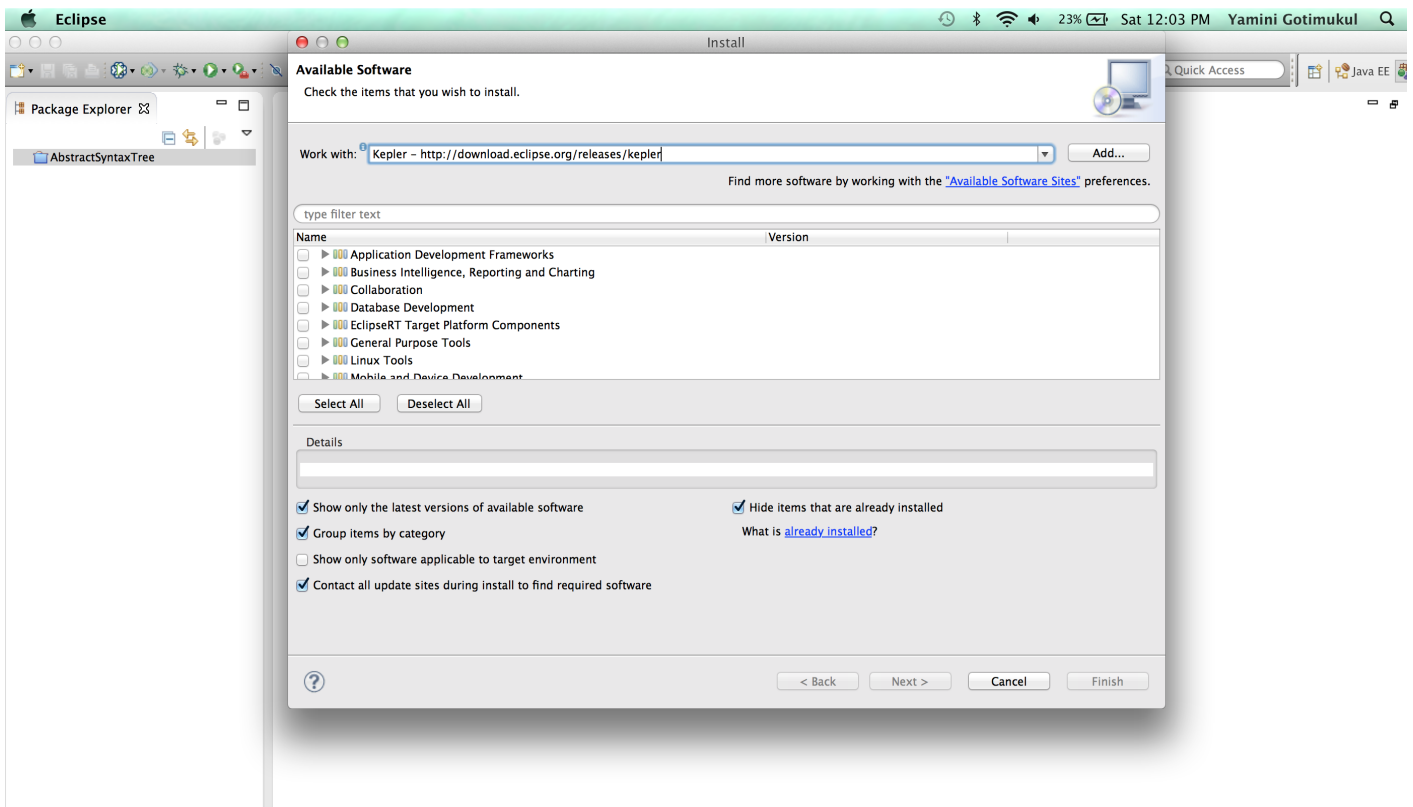
Installing Window builder in Eclipse

Step 1: Open Eclipse, Make sure all the open projects are closed

Step 2: Select the Help Menu --> Select the " Install new Software " and type in the following link in the " Work with " field

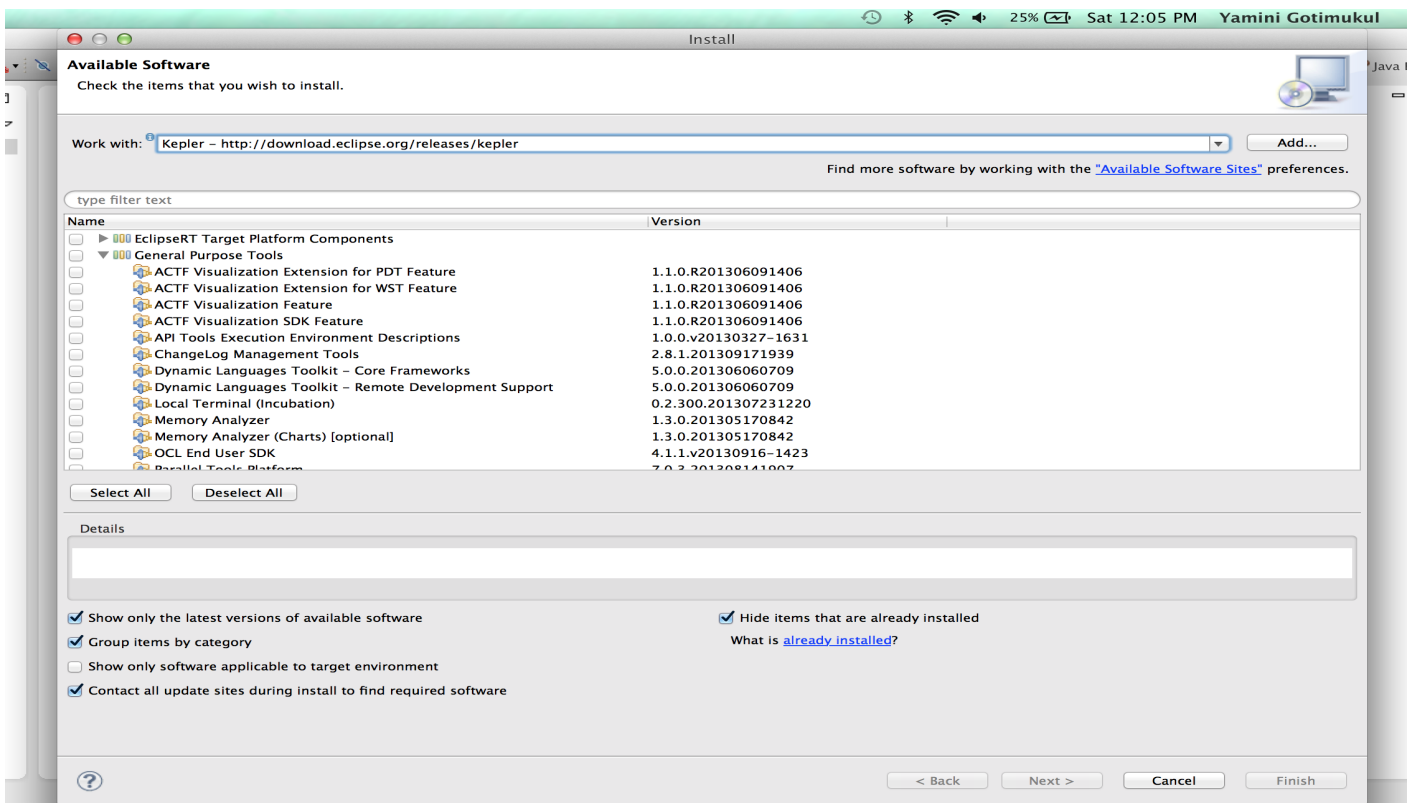
Kepler - <http://download.eclipse.org/releases/kepler>



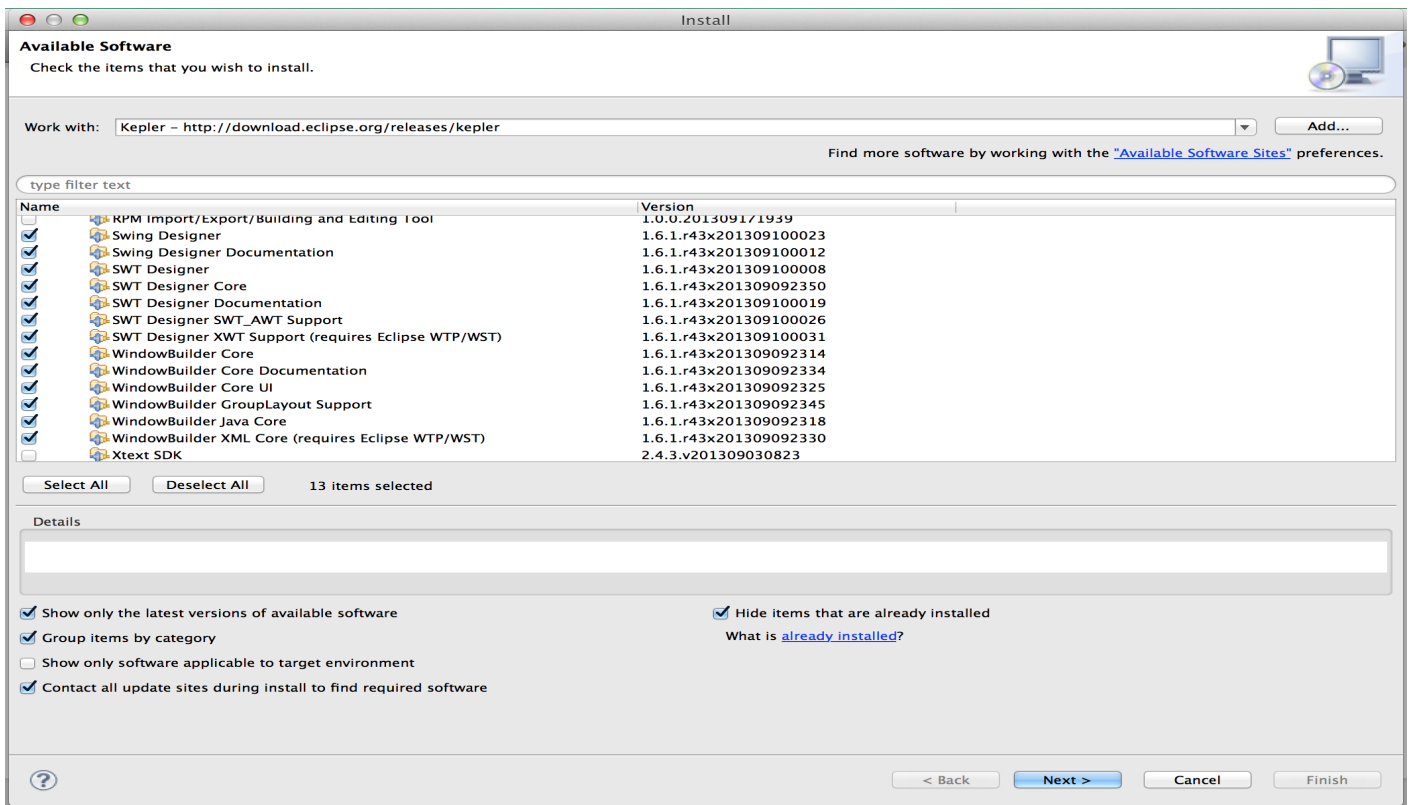


(Note : Depending on the version of eclipse use the link - when you type in the link it gets autocompleted for the version of eclipse you are using. This sample installation is on Eclipse Kepler (4.3 version)).

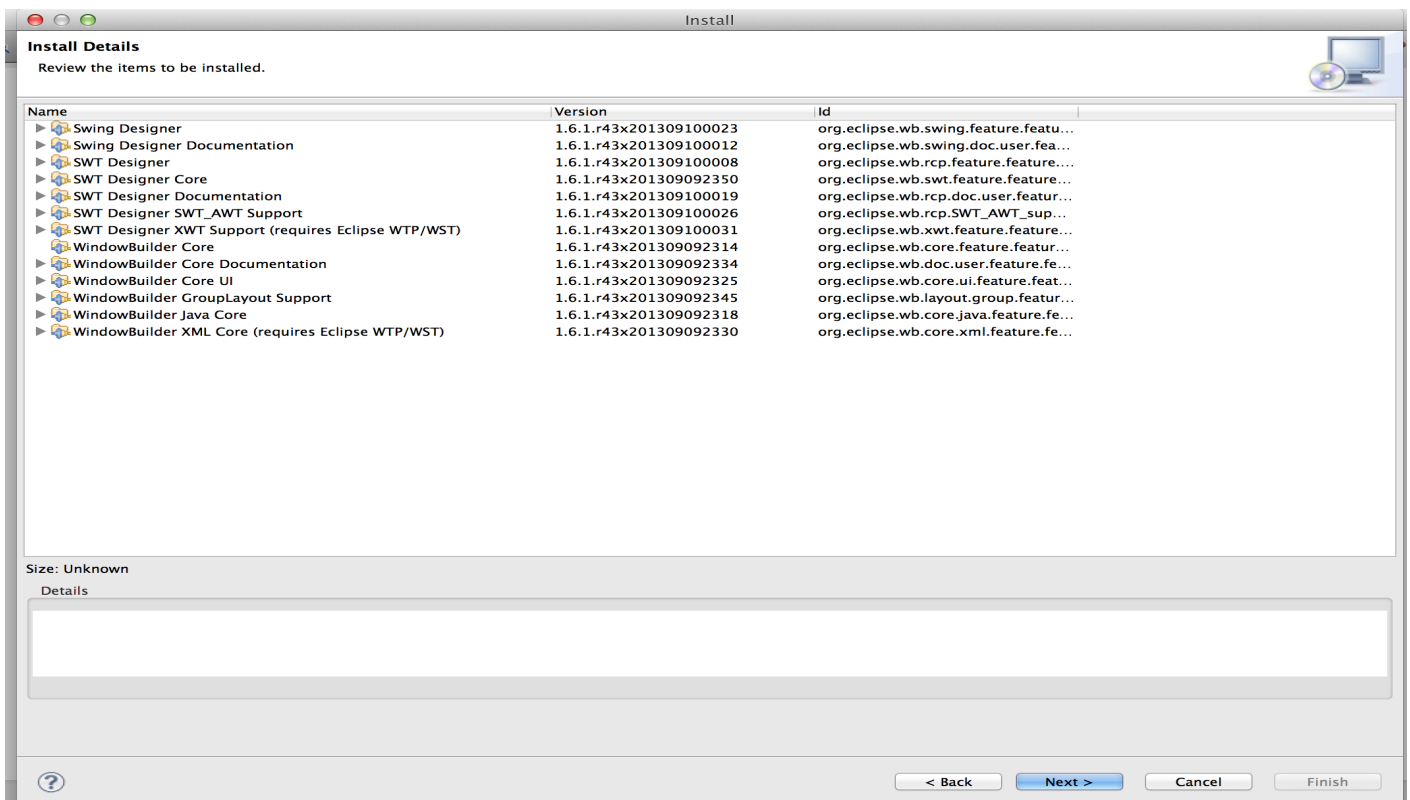
Step 3: Select the " General purpose tools " by expanding it.



Step 4: From the options under the " General purpose tools " select the item show in the screenshot and click on the next button.

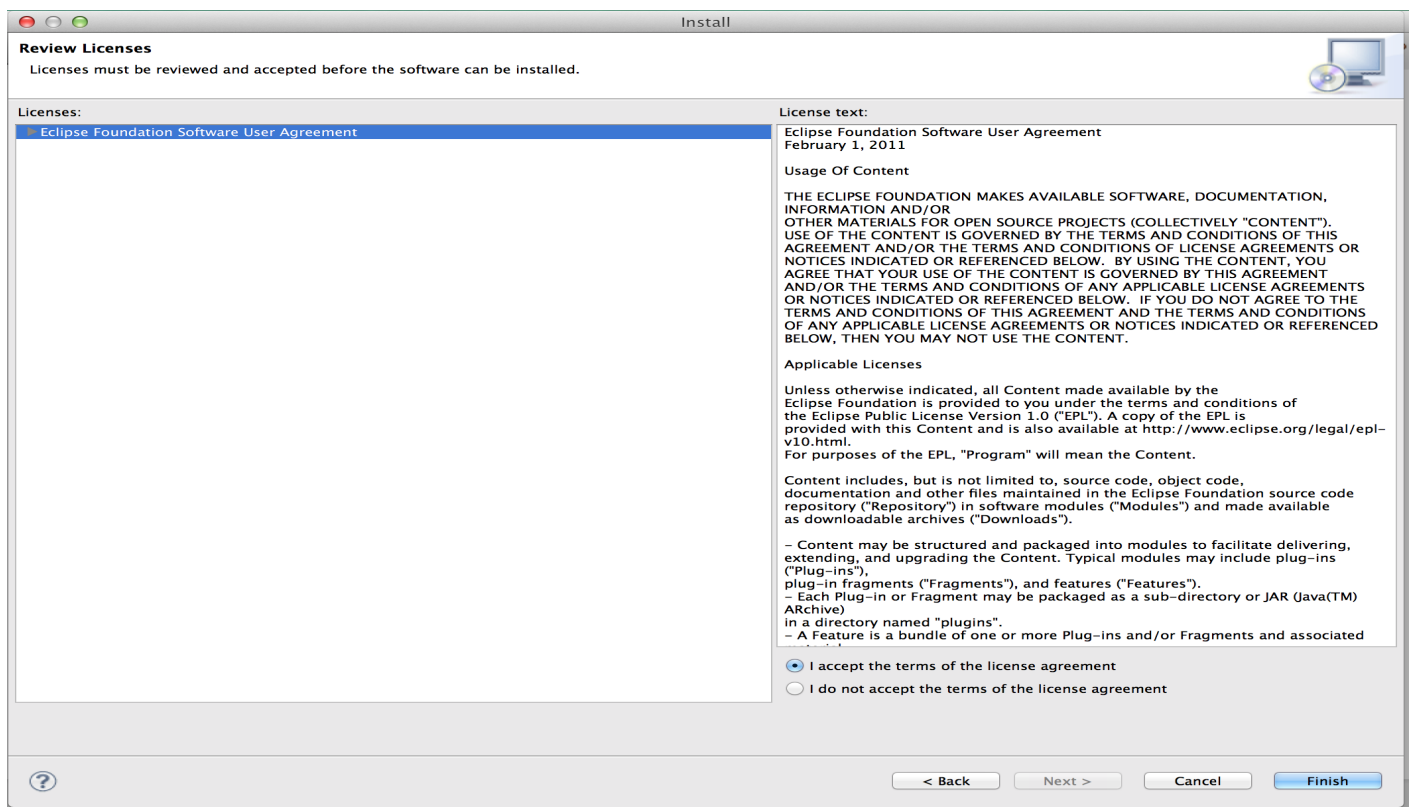


Step 5: Click on the next button and the review items to be installed window will be displayed

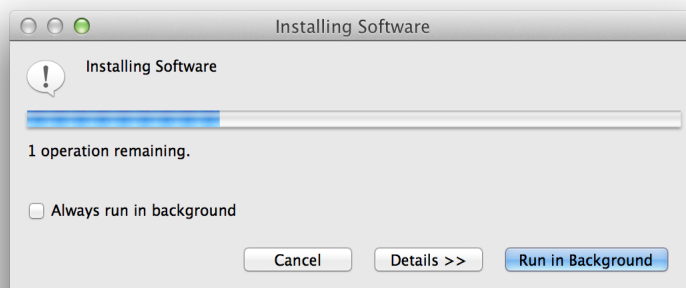


Click on the next button

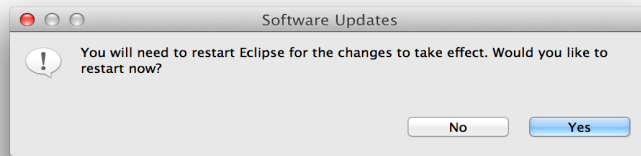
Step 6: Select on the " I agree to terms of the license agreement " radio button and click on the finish radio button.



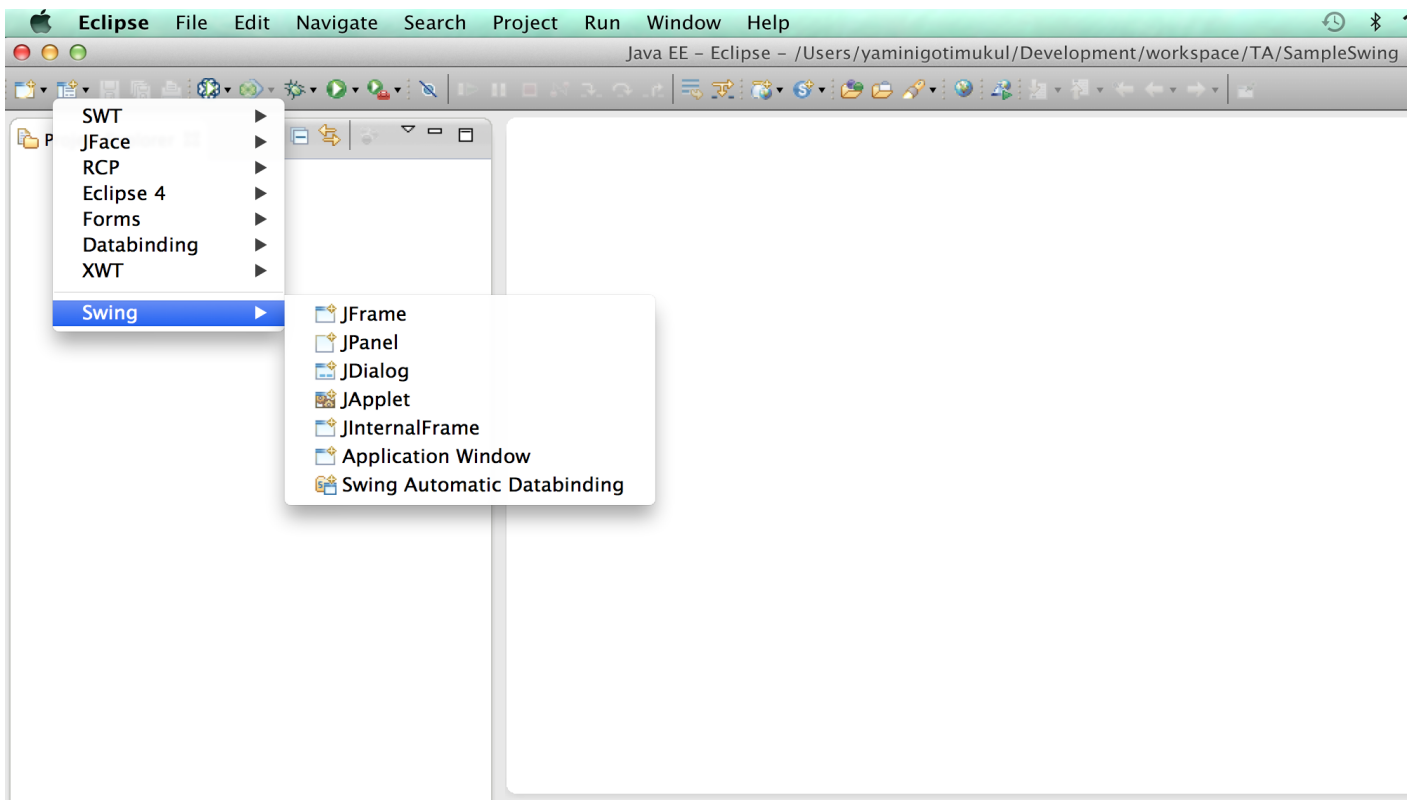
Click on the Finish button



Step 7: Restart eclipse once installation is complete by selecting yes in the following window.



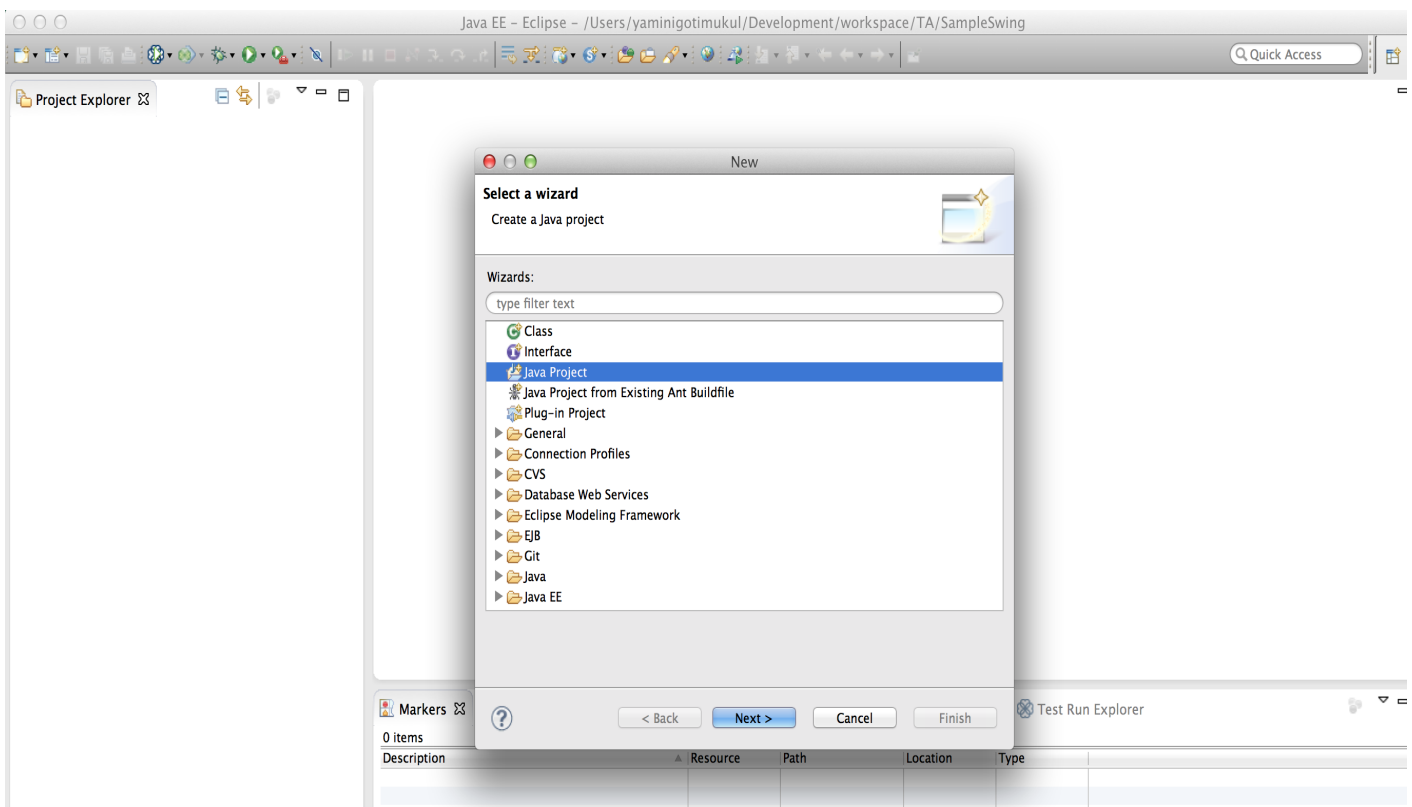
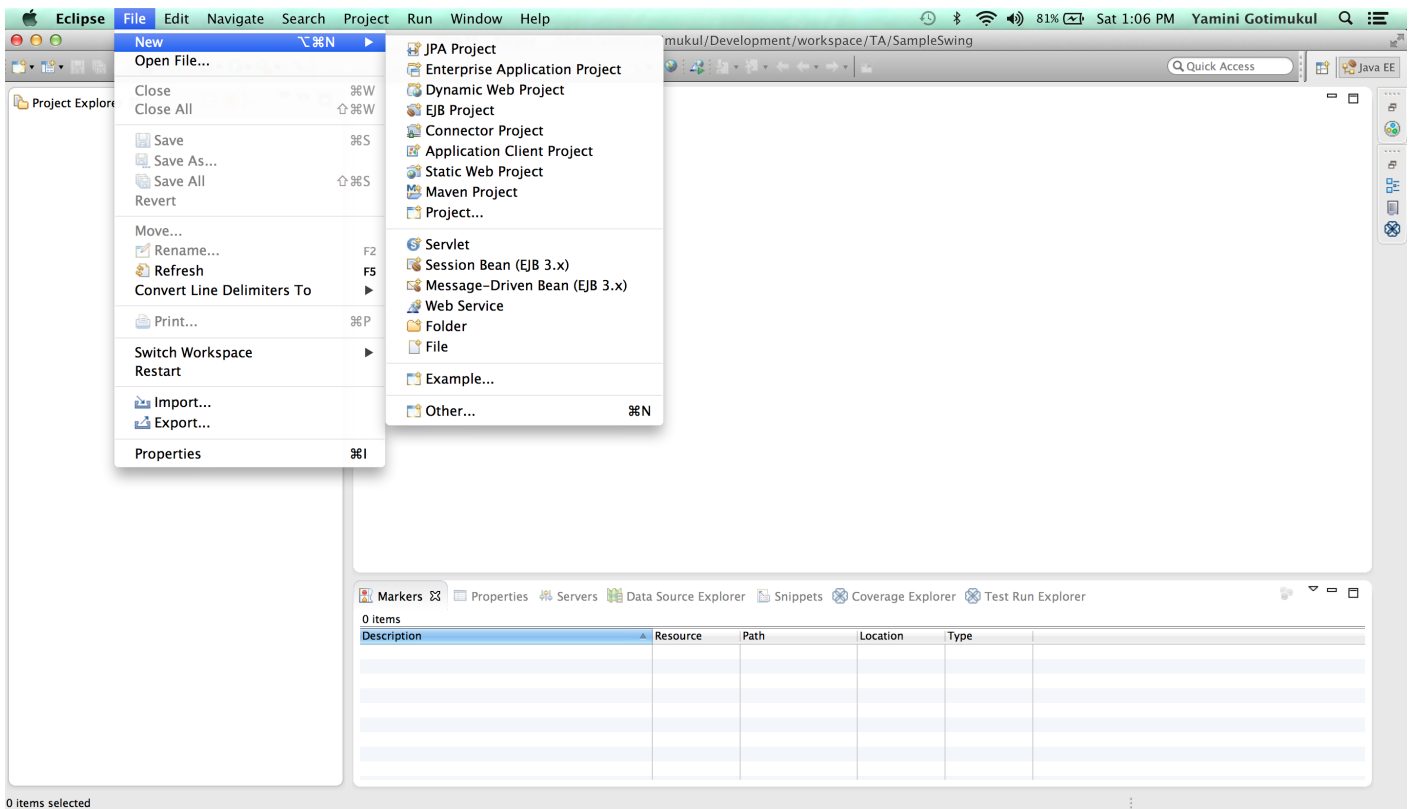
Step 8: To verify that installation is complete select the new visual class option you can see options Swing and SWT



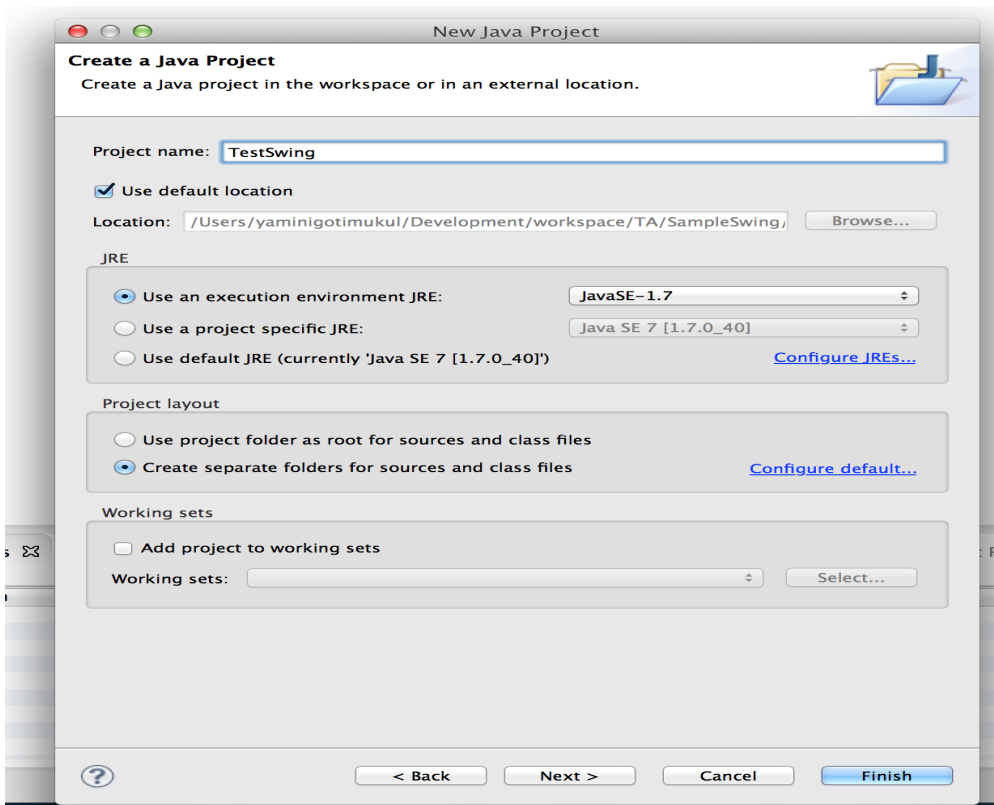
Creating a sample swing application

Step 1: Create a new Java project by selecting " java project " from the Select a wizard and click on Next button.

(Note: Steps for new java project creation, File menu --> New --> Other --> Select java project)

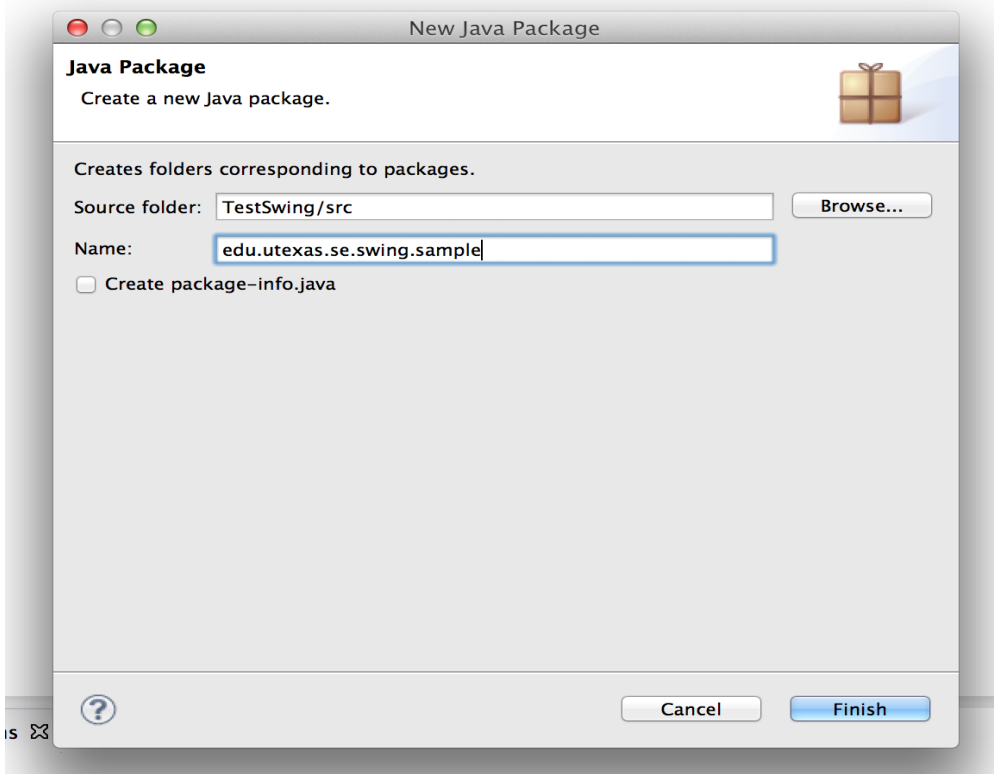


Step 2: Give a project name click finish.



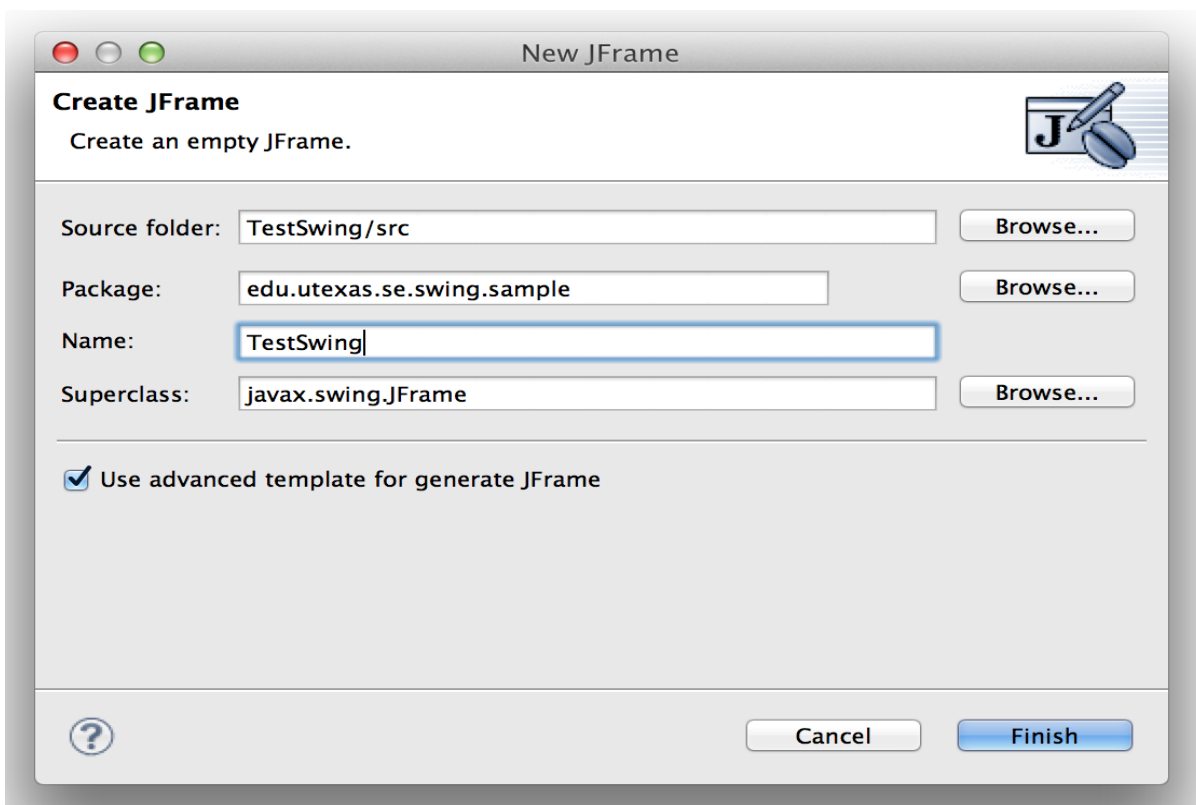
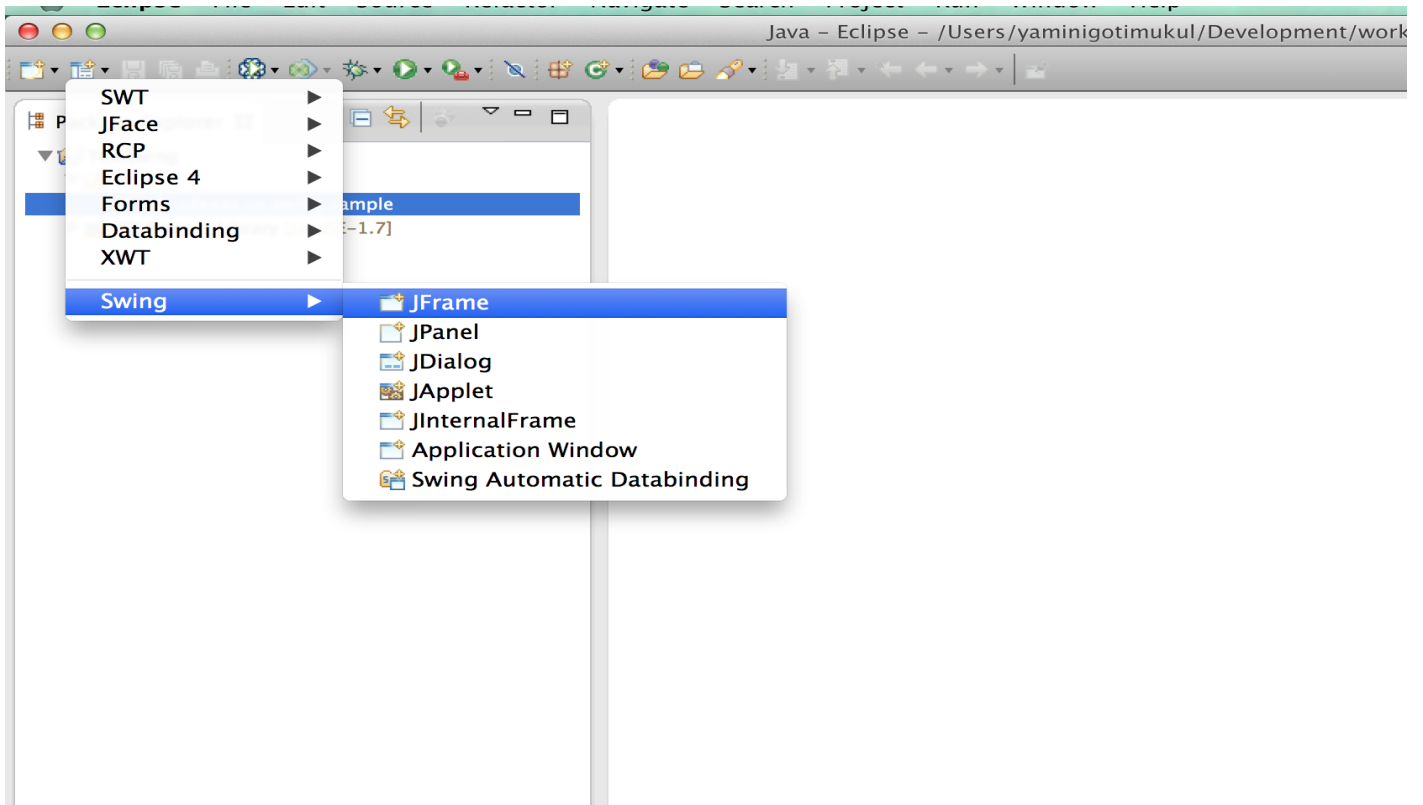
Step 3: Create a new package in the src folder.

(Note: Steps for new java package, File menu --> New --> Package)

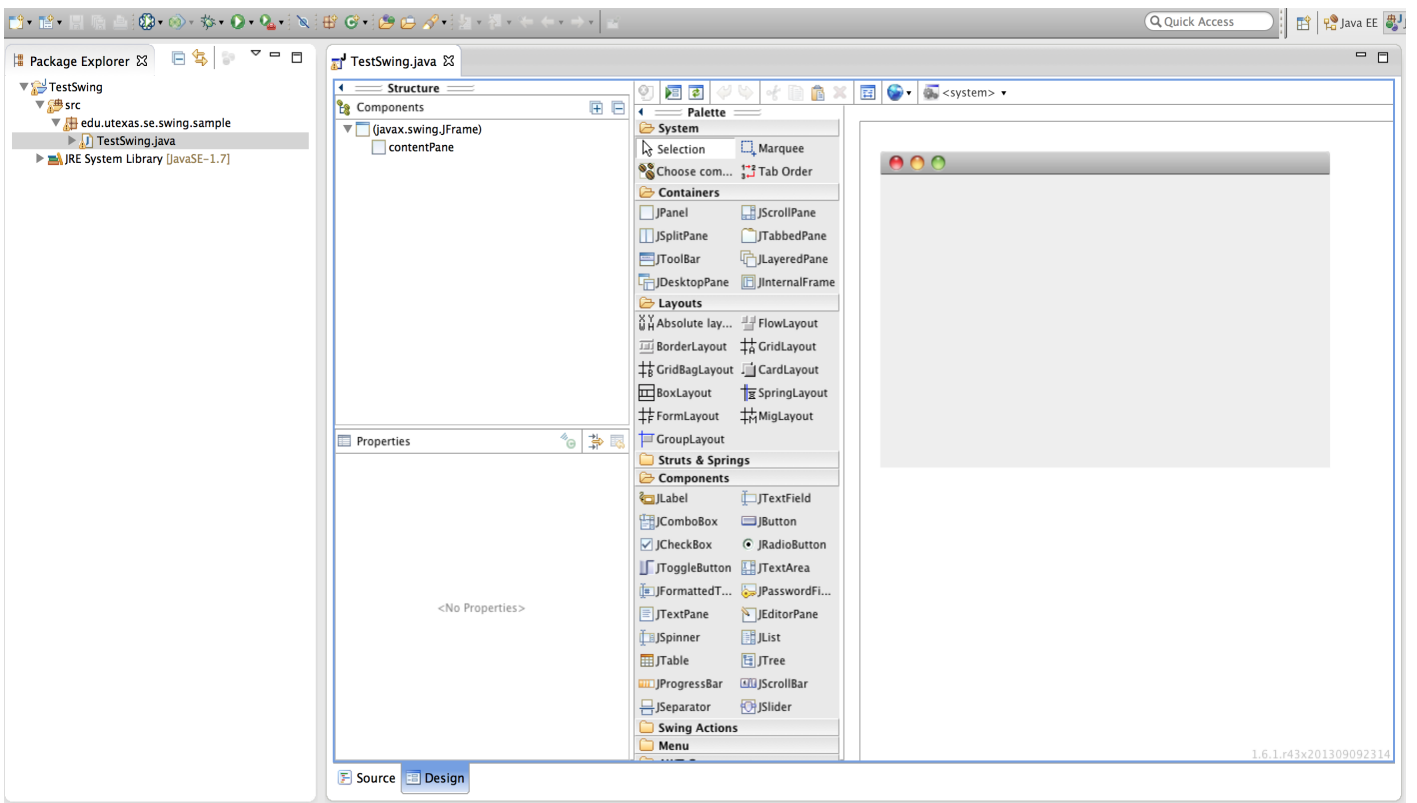
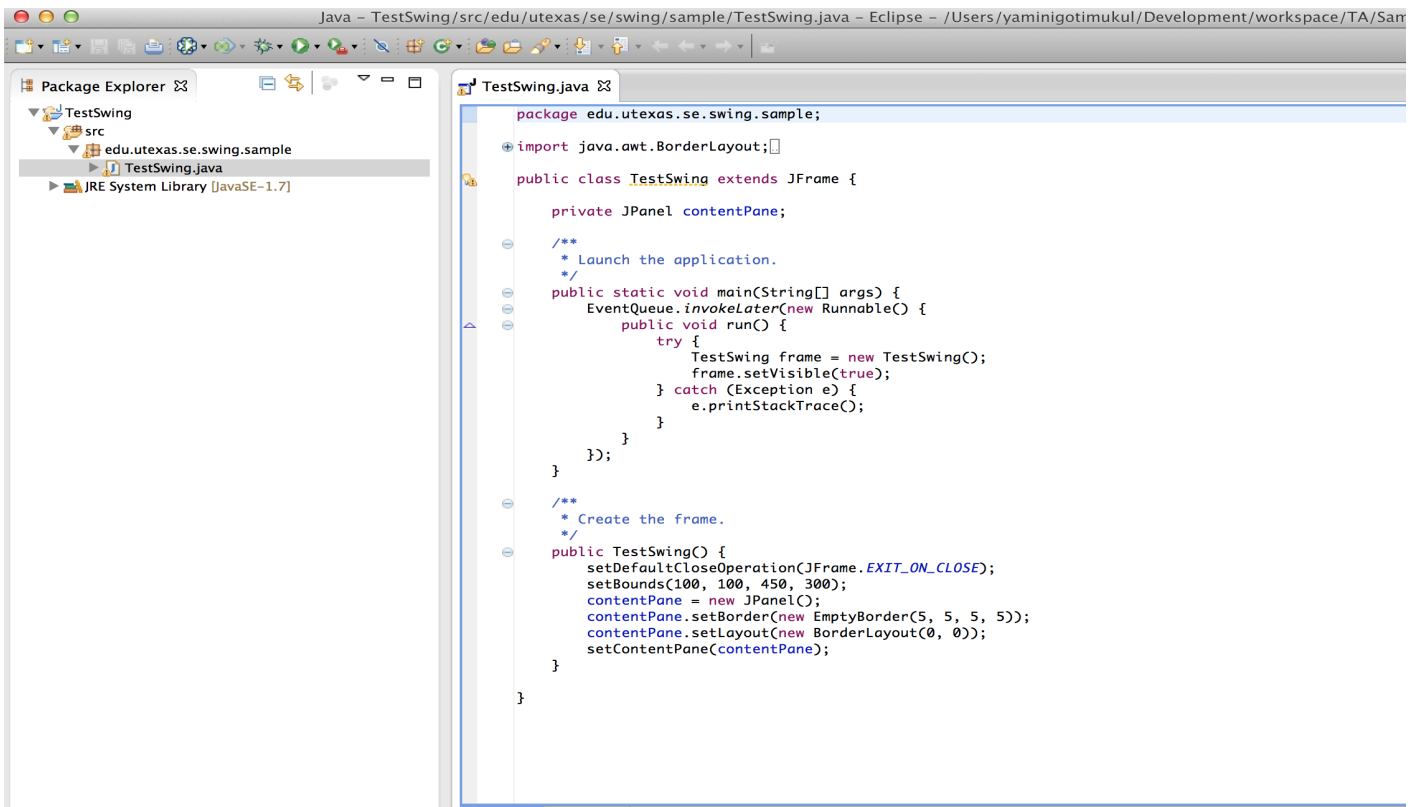


Click on the Finish button

Step 4: Now select the package and click on the new visual class option in the tool bar and select Swing --> select JFrame and give a name to the class and click finish button.



Step 5: A class with some existing code is created and now to use the window builder click on the " Design tab " at the bottom of the editor.

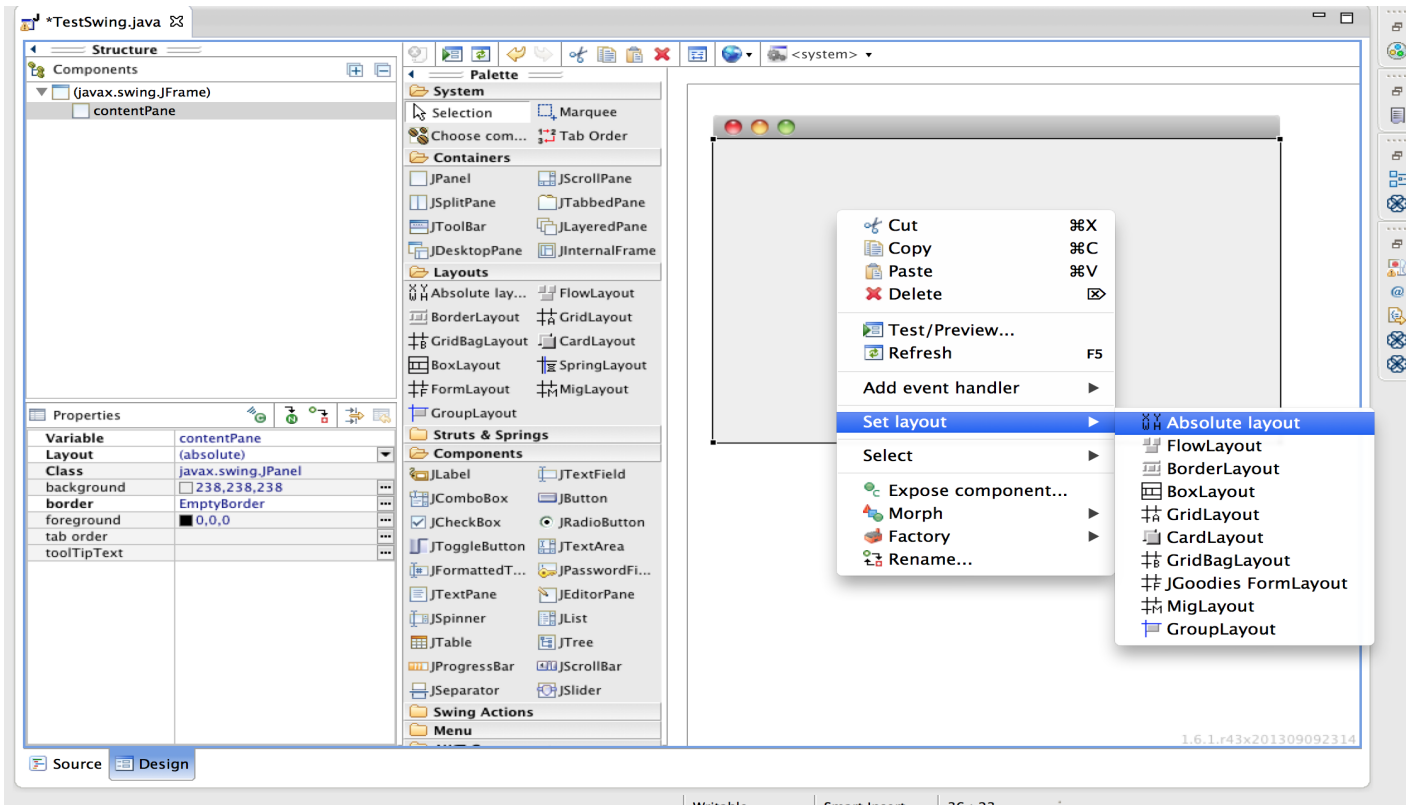


Step 6: Now the UI panel selection window opens up. You can drag drop the UI item on the sample screen and click on the " Source tab " to edit the UI related code

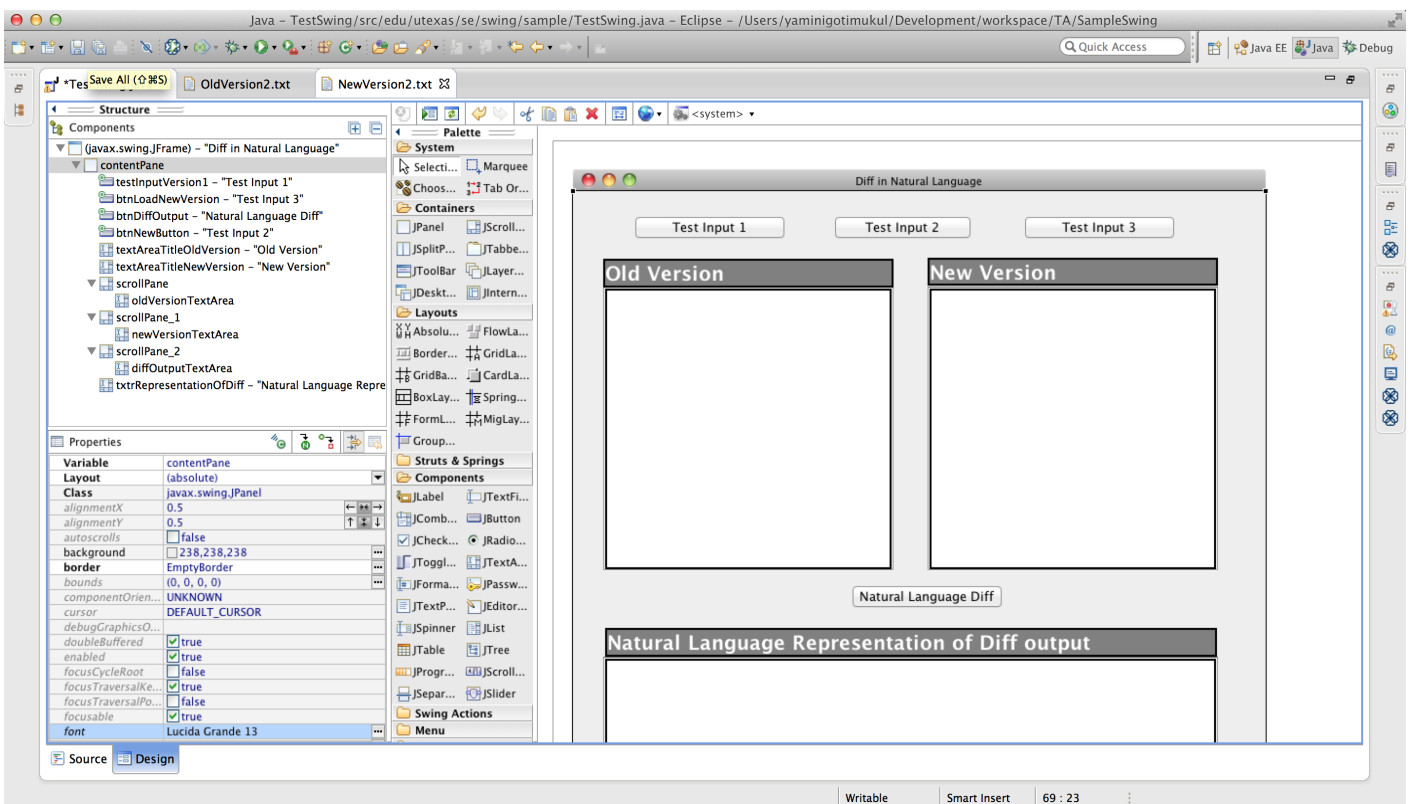
Step 7: The sample swing application is developed to add elements --> a JButton and a JTextArea, these elements can be added to the window by selecting the elements in the palette and dropping the element in right sample window .The application displays line wise diff output of two versions of the program in natural language format.

Step 8: This application is developed using the absolute layout.

To set layout, right click on the window frame and select " set layout " and select the appropriate layout and for the sample application " absolute layout " is selected



Step 9: Now drag and drop the button onto the JFrame. For the sample application, JButton, JTextArea have been selected. The following is the screenshot of the sample window application (After adjusting the size of the elements)



Note: To add the scroll bar to the text area first drag drop the jscrollpane and then add the jtextArea element to the jscrollpane

Step 10: You can change the attributes of the elements by double clicking on them, it opens source tab with cursor at the selected elements code. The properties like font, background color can be changed in the properties pane in the left corner or by modifying the source code

(Note: To add code for handling event on the button (like button clicking events) there are two ways, either double click on the element on the sample JFrame OR right click on the element and select add event handler --> action--> action performed. The corresponding event handling code gets added to the source.)

```

*TestSwing.java
private JTextField textField;

/**
 * Launch the application.
 */
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {
                TestSwing frame = new TestSwing();
                frame.setVisible(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    });
}

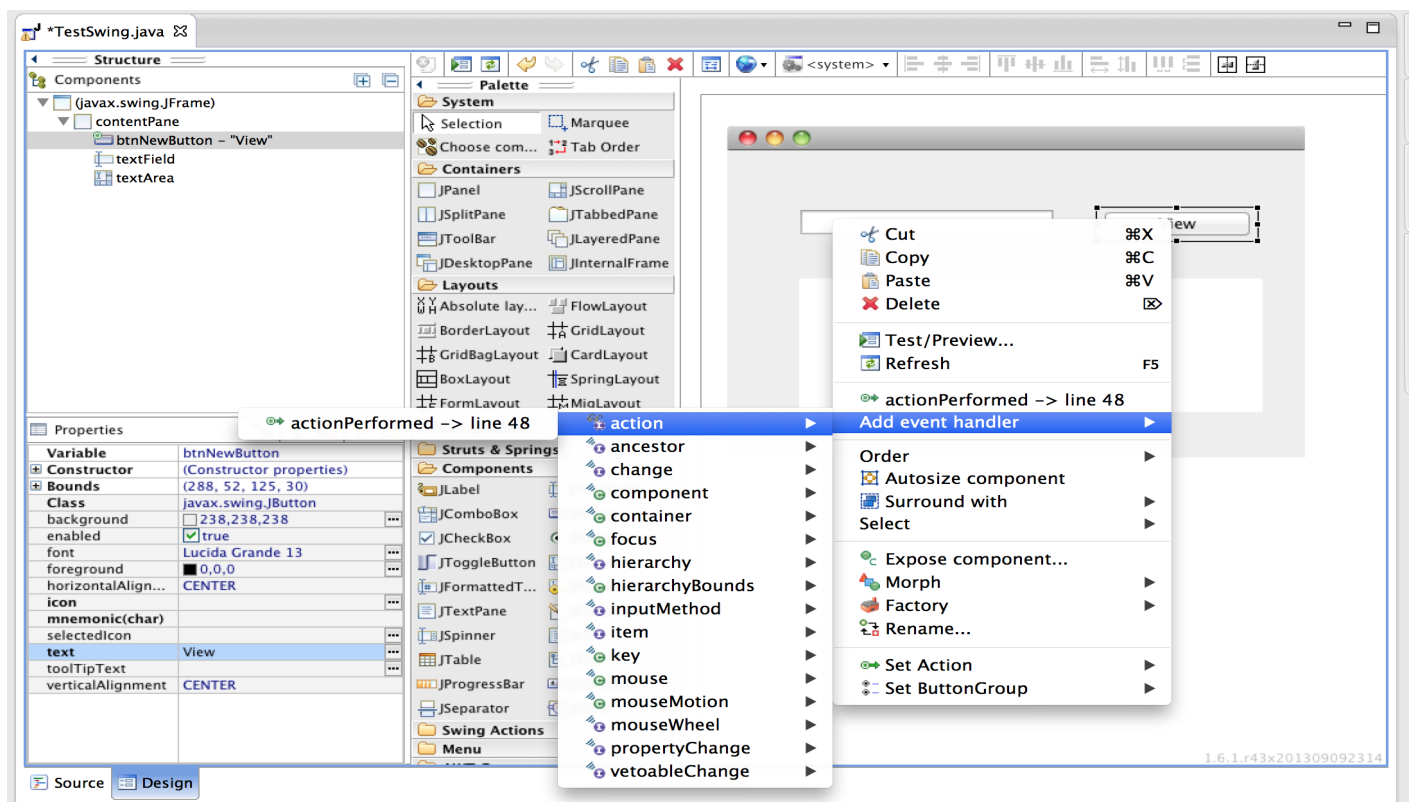
/**
 * Create the frame.
 */
public TestSwing() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(null);

    JButton btnNewButton = new JButton("View");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
        }
    });
    btnNewButton.setBounds(288, 52, 125, 30);
    contentPane.add(btnNewButton);

    textField = new JTextField();
    textField.setBounds(54, 51, 203, 28);
    contentPane.add(textField);
    textField.setColumns(10);

    JTextArea textArea = new JTextArea();

```



Step 11: You can view the sample code in the source folder in the following link

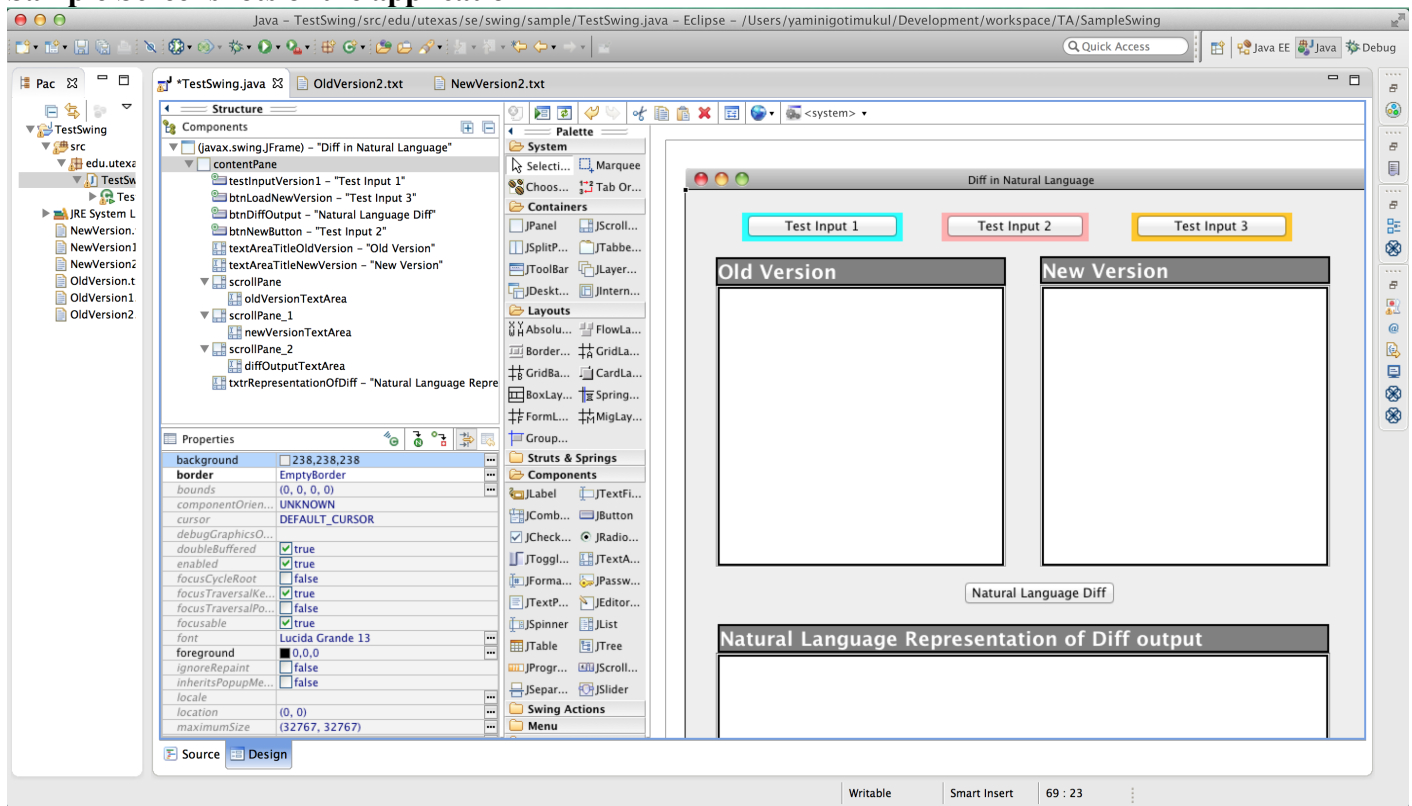
<https://bitbucket.org/YaminiGotimukul/sampleswing/src/d551905ff5d00ded164a1e99466263943ba378b5?at=master>

Step 12: To run the sample application right click on the project folder in the left navigation and click Run As --> Java Application

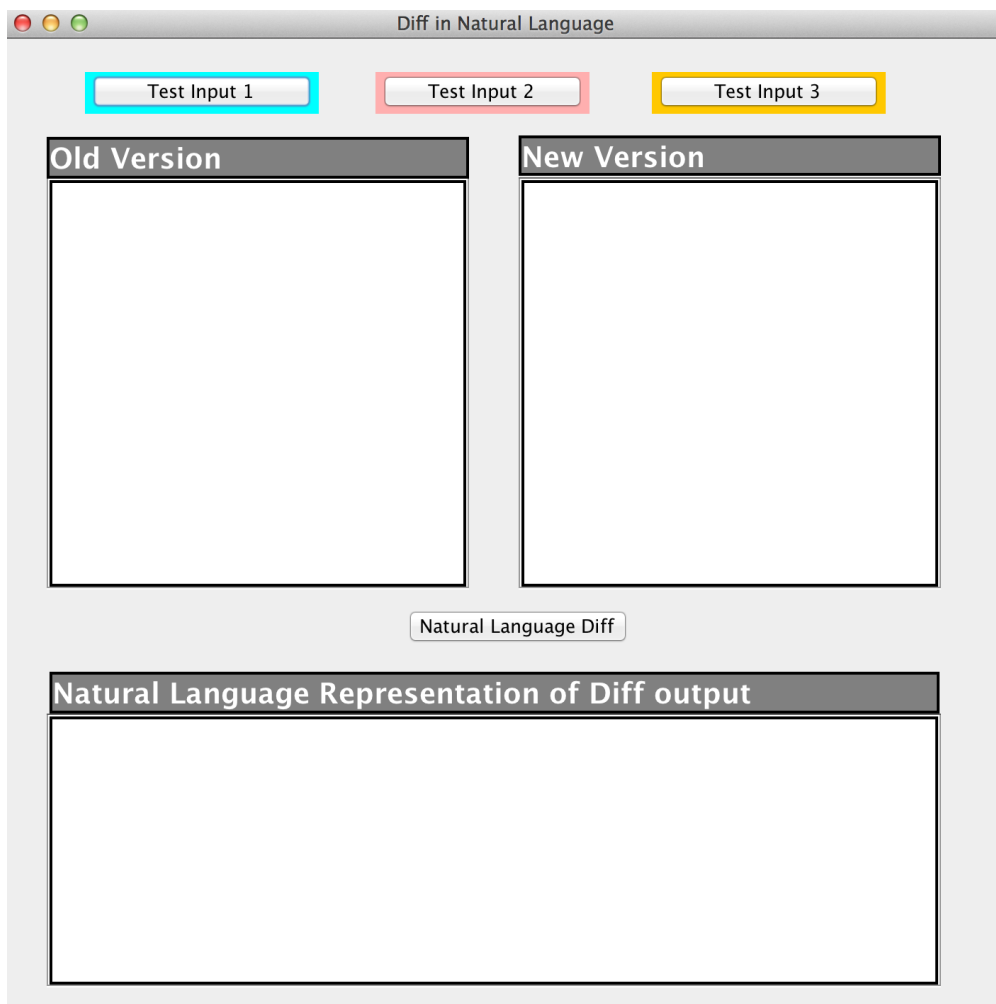
(Note : You can clone the application or copy the source code from the link specified in Step 11 before running the application)

Step 13: You can choose any of the three inputs (by clicking on the buttons)and get the diff output in the natural language

Sample Screenshots of the application



Sample application window



Sample diff outputs for different input programs
Test Input 1

Diff in Natural Language

Test Input 1 Test Input 2 Test Input 3

Old Version

```

1.class A{
2.public A(){
3.public void foo()
4.}
5.}

```

New Version

```

1.class A{
2.public A(){
3.public void bar()
4.}
5.}

```

Natural Language Diff

Natural Language Representation of Diff output

Line 3 : The method public void foo() in old version changed to public void bar()

Test Input 2

Diff in Natural Language

Test Input 1 Test Input 2 Test Input 3

Old Version

```

1.public class A {
2.
3.void m1() {...}
4.}
5.
6.public class B extends A {
7.
8.void m2() {...}
9.}
10.
11.public class E1 extends Except
12.public class E2 extends E1 {}
13.public class E3 extends E2 {}

```

New Version

```

1.public class A {
2.
3.void m1() {...}
4.}
5.
6.public class B extends A {
7.void m1() {...}
8.void m2() {...}
9.}
10.
11.public class E1 extends Except
12.public class E2 extends E1 {}
13.public class E3 extends E1 {}

```

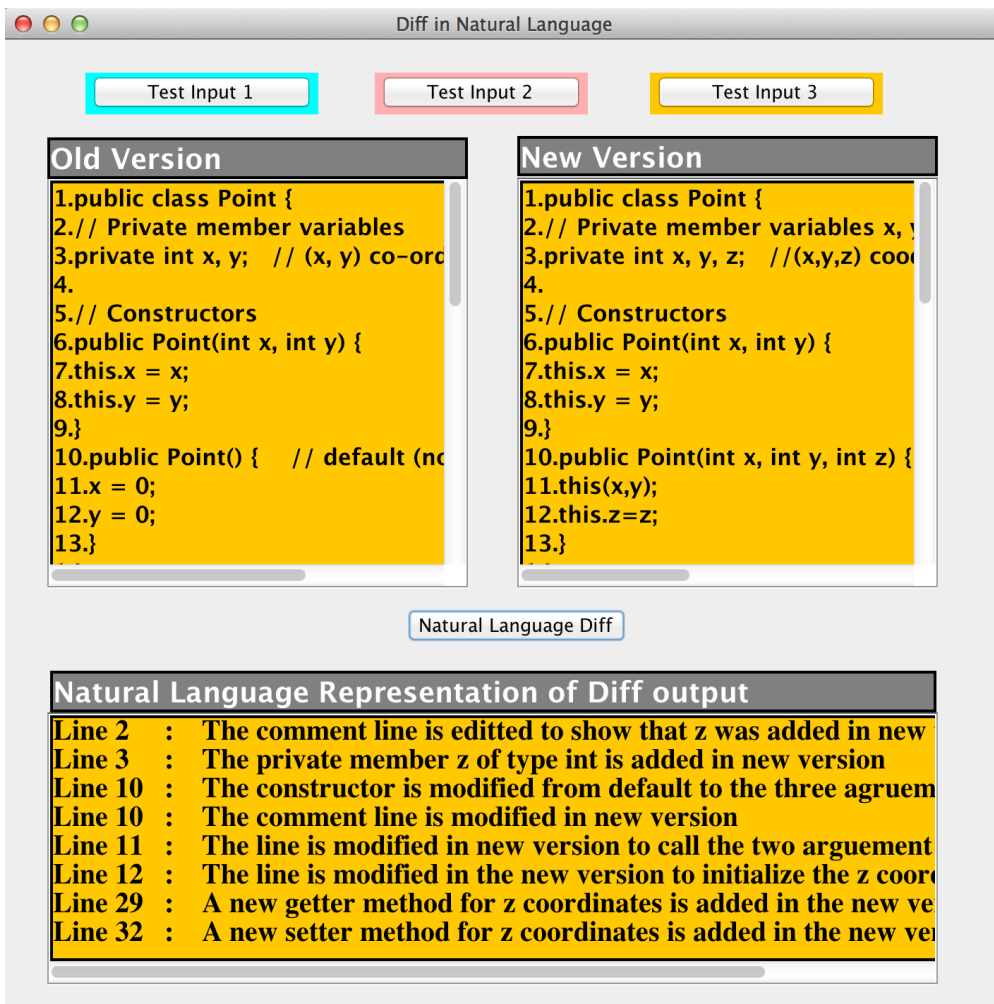
Natural Language Diff

Natural Language Representation of Diff output

Line 7 : The method void m1() is overridden in Class B in new version

Line 13 : Exception hierarchy in E3 new version is changed from E2 to E1

Test Input 3



(Note : The input is taken from 6 different file, if you want to run the application on your system keep the input files in the project root folder(refer screenshot under step 13))

Reference online Swing tutorial links are as following

1. <http://zetcode.com/tutorials/javaswingtutorial/>
2. <http://www.javabeginner.com/java-swing/java-swing-tutorial>