CS209 Lab10
fa2014
dutta/streller

PROBLEM STATEMENT:

Applets are designed to be embedded in HTML pages for display in web browsers.
The Applet or Object tag is used to embed an applet in a web page. You are to create Java applets and the HTML page within which an applet is embedded.
You will also need to use passing of parameters to an applet.

Notes:

Applets should be deployed in a JAR file on a web server for remote or local access. This topic, however, is outside of the scope of this current exercise.
There are several ways to run an applet, including the following, which we will do:

One way is to use the simple Appletviewer tool which only displays the applet and not the HTML page. This tool can be run either from within an IDE such as NetBeans or from the command line. We will use the Appletviewer from within NetBeans.
A second way is to use your web browser to display the HTML page containing the applet. Normally the HTML page would be downloaded from a remote web server in response to a request from you via your browser. However, if the page and applet are on your local machine, then you can view the page with its embedded applet, directly (without being deployed on a web server).
We will also use this method of testing and viewing applets.

CODE :

This Lab builds on your completed project from Lab 8 : you will create an applet version of your program from Lab8.

Create a JApplet in your project for this Lab Exercise.

Use Lab10 as your project name ( create a Java Application )
check the Create Main Class and enter lab10.lab10
After creation, delete the default class created ( Lab10.java )

In the project pane, on the package name, right click to add a new Japplet to the project
Call the class LoanArrayAppletEx.

The new JApplet should be listed in the left-side Project pane of your NetBeans project.

Copy the contents of your LoanArrayEx file (from Exercise 8) and paste the text into the Java JApplet class/file that you created.
( do not lose the init() method, and be sure to include all the import statements )

Change all occurrences of the name LoanArrayEx to LoanArrayAppletEx.

Check the first line of the declaration of the new class to ensure that the new class is a subclass of JApplet. The first line should read:

```
public class LoanArrayAppletEx extends JApplet
```

Eliminate the main method from the new class/file.
Eliminate the statement: super(“LOAN GUI DEMO”);
Save your work (save your files).

For this project, add all the other classes/files that were in Exercise 8 to this project

In the init() method
    Initialize max number of Loans to 13.
    Create the array.
    Set the border
    Set the Text
    Set the size

Compile the JApplet.

In the left-side Project pane, right-click on the JApplet file and select Compile File from the pop-up menu.

Correct any compile errors.

The required HTML file containing Applet tag is automatically created by NetBeans.

Remember that an applet must be embedded within an HTML page to run in a real application.

When you run a JApplet within NetBeans, the HTML file for the JApplet will be automatically created by NetBeans and placed within the build folder.

The HTML file will contain the required Applet tag.

Run the applet as instructed in the next step. Then check the build folder in the Files tab pane to see that the HTML file is created and within the build folder.

Run the applet using the Appletviewer within NetBeans.

As mentioned above, there are several ways to execute an applet.

First we will use the Appletviewer that is part of NetBeans.

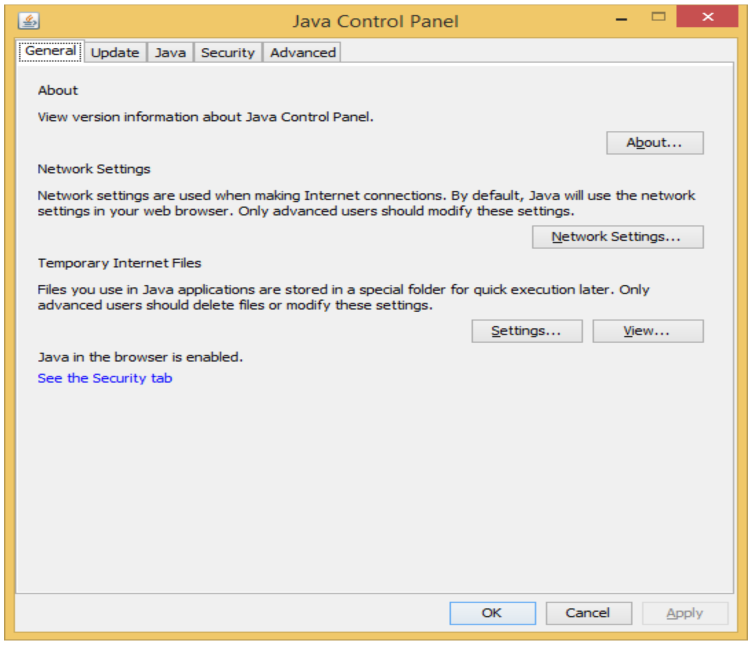
Note that the Appletviewer tool, which is part of the basic JDK toolset, only displays the applet, not the HTML page containing the applet, as shown in the figure below. However, the name of the HTML page is passed as a parameter to the Appletviewer, so that the Appletviewer has access to the information within the Applet tag element (e.g., codebase, parameters).

In the left-side Project pane, right-click on the LoanArrayAppletEx.java file and select Run File from the pop-up menu. This will bring up the applet in the Appletviewer. You should be able to interact with the applet and use all its capabilities.

We will now create the HTML page for this applet

Use either Firefox (preferred) or Chrome as your web-browser.

Due to potential security risks, applets are more of than not blocked by Java.
Open the Java Control Panel by typing in the appropriate search field, Java Configure



About

View version information about Java Control Panel.

Network Settings

Network settings are used when making Internet connections. By default, Java will use the network settings in your web browser. Only advanced users should modify these settings.

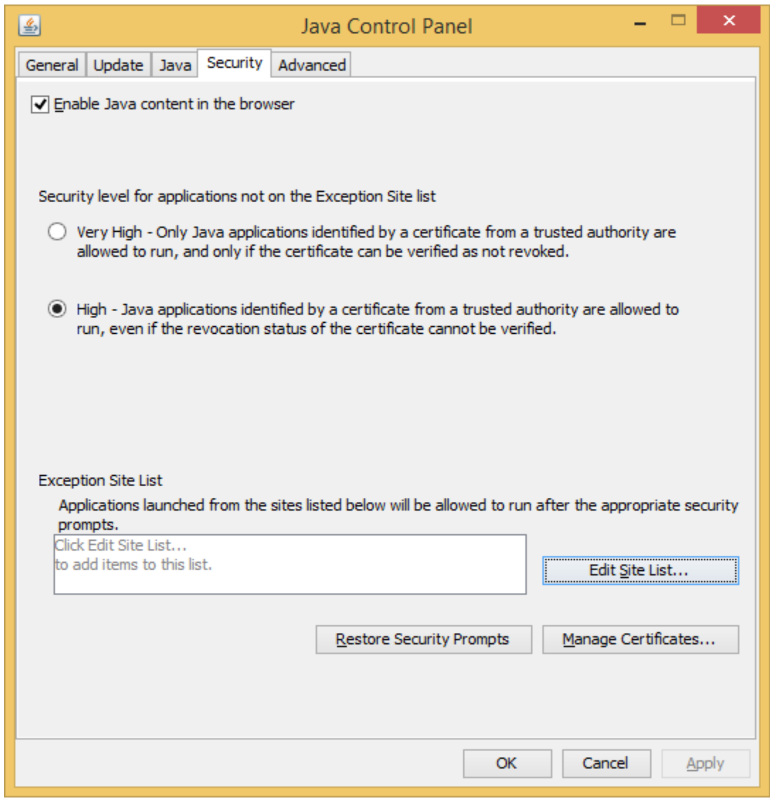
Temporary Internet Files

Files you use in Java applications are stored in a special folder for quick execution later. Only advanced users should delete files or modify these settings.

Java in the browser is enabled.

See the Security tab

And now open the security tab



[✓] Enable Java content in the browser

Security level for applications not on the Exception Site list

○ Very High - Only Java applications identified by a certificate from a trusted authority are allowed to run, and only if the certificate can be verified as not revoked.

● High - Java applications identified by a certificate from a trusted authority are allowed to run, even if the revocation status of the certificate cannot be verified.

Exception Site List

Applications launched from the sites listed below will be allowed to run after the appropriate security prompts.

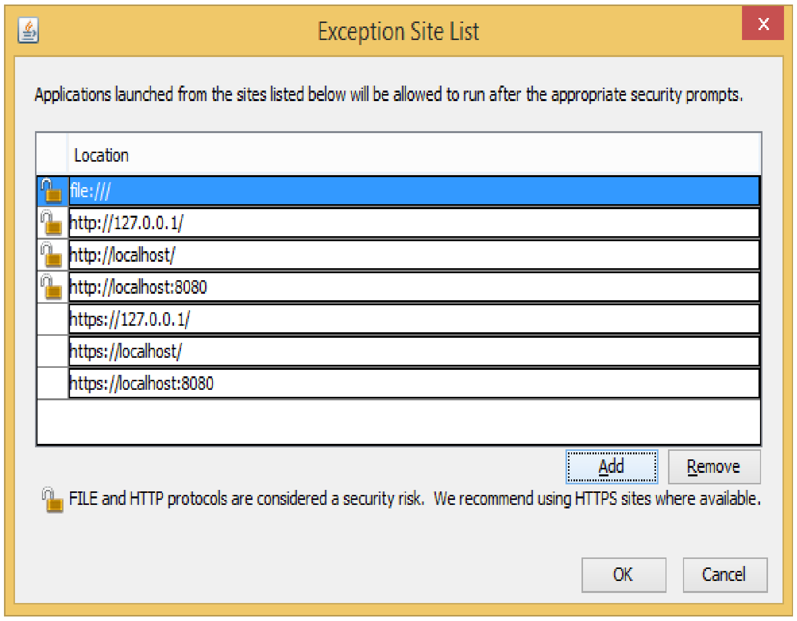
Click Edit Site List...
to add items to this list.

Make sure the security setting is as low as possible

Now add to the Exception Site List the following (by clicking on Edit Site List)

http://localhost/
http://127.0.0.1/
https://localhost/
https://127.0.0.1/
file:///

Only the file:/// entry is needed unless you're running a local server such as a web server, Tomcat, Glassfish, etc., and plan on serving up Java applets and/or WebStart applications. In that case, you may need to specify a non-standard port number. For example, by default the Glassfish server uses port 8080, so you need to add "http://localhost:8080/" and "https://localhost:8080/".

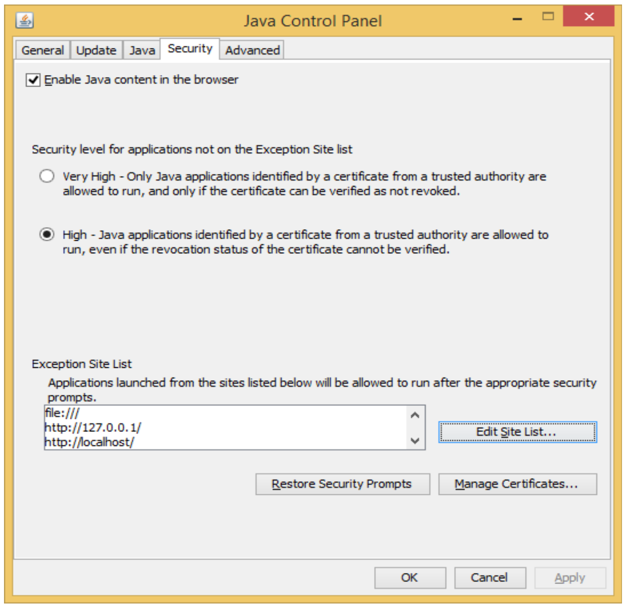

Exception Site List

Applications launched from the sites listed below will be allowed to run after the appropriate security prompts.

| Location |
|---|
| file:/// |
| http://127.0.0.1/ |
| http://localhost/ |
| http://localhost:8080 |
| https://127.0.0.1/ |
| https://localhost/ |
| https://localhost:8080 |

FILE and HTTP protocols are considered a security risk. We recommend using HTTPS sites where available.

The Java Control Panel is now :



[✓] Enable Java content in the browser

Security level for applications not on the Exception Site list

( ) Very High - Only Java applications identified by a certificate from a trusted authority are allowed to run, and only if the certificate can be verified as not revoked.

(●) High - Java applications identified by a certificate from a trusted authority are allowed to run, even if the revocation status of the certificate cannot be verified.

Exception Site List

Applications launched from the sites listed below will be allowed to run after the appropriate security prompts.

file:///
http://127.0.0.1/
http://localhost/

Now that java has been configured to allow html applets run, we can now proceed to create the html page

The first step is to modify the applet source code so that the parameters are received from the HTML page and not from the applet source code.
Now that these changes have been made, the applet will no longer correctly display in the appletviewer.

> Add an init() method to the JApplet class/file and add code within the init() method body that will get the applet parameter named TextAreaTitle that you added to the Applet tag element in the HTML file. An empty init method would consist of the following:
>
> ```
> public void init() { }
> ```
>
> You need to fill in the method body with appropriate code including a statement that calls the getParameter method:  getParameter("TextAreaTitle")
>
> Note: The applet's getParameter method can only be invoked after an instance of the applet has been created. So this method cannot be invoked in the constructor, but instead should be invoked in the init() method.
>
> Assign the value returned by the getParameter method to a member variable in your applet class named textAreaTitle. The variable should have class scope.
>
> Move the statement that sets the TitledBorder on the JTextArea. This statement should be moved to the applet's init() method, after the method call to get the parameter. Change the statement so that it uses the value of the textAreaTitle variable to set the text of the TitledBorder. Do not use a hardcoded literal string to set the title on the TitledBorder as we did previously.
>
> See the following sections of your textbook for relevant information:
>
> > Section 18.3 for details on the HTML Applet tag.
> > Section 18.7 for details on the specification of parameters in the HTML Applet tag element.
> > Section 18.7 of your textbook for details on how to get the parameters in the JApplet code.

Modify the JApplet and HTML page to pass a second parameter to the applet.

> Add a parameter named maxNumberOfLoans in the Applet tag element in the HTML file. As the value of the parameter use the text string: "15".
>
> Add code to the init() method that will get the applet parameter named maxNumberOfLoans.
>
> Also in the init() method, add code to assign the value returned by the getParameter method to the existing final member variable in your applet class named MAX_LOANS. This variable should be used to declare the size of the Loan array. The value of the variable is also displayed as the value of the second JTextField labeled "Max # of Loans:".

Modify the JApplet by adding a JScrollPane to the GUI.

> Create an instance of the JScrollPane class. When invoking the JScrollPane constructor, pass the JTextArea object as parameter to the constructor so that the JTextArea is a component on the JScrollPane (the JScrollPane holds the JTextArea as its content).
>
> Then add the JScrollPane to the JPanel instead of the JTextArea.
>
> This addition of the JScrollPane should automatically provide scroll bars when needed in the JtextArea.

Compile the code to create "updated" class files.

Modify the JApplet and HTML page so that a parameter is passed to the applet.

Open the HTML page for editing in the NetBeans edit pane.

Add a parameter named textAreaTitle in the Applet tag element of the HTML file. As the value of the parameter use the text string: “DOLLARS & SENSE LOAN List”. The additional tag should consist of the following. This tag must be nested within the Applet tag:

```
<param name = textAreaTitle   value = " DOLLARS & SENSE LOAN List">
```

Compile the code to create “updated” class files.

Due to further Java applet  security restrictions, applets now must be signed. We will use self-signed applets for our local work.

Create Create a directory within your NetBeans project called Sandbox
Create a Manifest file
Create a text file, in this directory, called Lab10AppletManifest.txt with these three lines in it:

```
Permissions: all-permissions
Codebase: *
Application-Name: LoanArrayApplet
```

Make sure each line ends with a NEWLINE, including the last line

Copy the NeatBeans directory ...\build\classes\lab10 into Sandbox
(This directory contains all of the class file for your applet lab)

Now package the applet into a JAR file.  The applet must be in a JAR file before a certificate can be attached to it.  Call the JAR file “Lab10_Applet.jar”.  Use the jar utility to create a new JAR (Java ARchive) file, and include the manifest entries from the file you created and all the class files by executing the below statement in the cmd shell which has its current directory Sandox

```
jar -cvfm Lab10Applet.jar   Lab10AppletManifest.txt   lab10
```

Inside Sandbox you should now see this jar file

Move the NetBeans created HTML file to the Sandbox
remove  the code within the APPLET tags
and insert

```
<APPLET CODE="lab10.LoanArrayAppletEx" WIDTH="350" HEIGHT="500"
ARCHIVE="Lab10Applet.jar">
 <param name = textAreaTitle   value = "Loans">
<param name = maxNumberOfLoans  value = "666">
</APPLET>
```

You must now sign the certificate

Using the cmd shell, or the power shell, navigate to the Sandbox directory and create a key pair in user.home/.keystore by executing

```
keytool -genkeypair -alias mykey
```

the alias “mykey” is user's choice but use “mykey” anyway
use for the password : password ( so we are all on the same page )
for all other entries just return to leave no values.

This will create a private key to sign your code with, and a public key that can be used to validate the digital signature. Since Java 6, it will also generate a self-signed certificate. The keys will be put in the default file in the default location (the file “.keystore” within your “HOME” directory). A file containing a set of keys is called a keystore. You can specify an option to use a different keystore if you want.

The alias you give is the name for the pair of keys. (Sometimes, you may have multiple key pairs, and might use different keys for different applications. So you need to give each pair a name, or alias.) There are defaults for everything (including the alias, which does default to “mykey”), but you can use additional command line options to override the defaults.

Now that you have a signed certificate, you can sign your JAR file with it:

In the cmd or powershell enter

```
jarsigner -signedjar SignedLab10Applet.jar Lab10Applet.jar mykey
```

This will ask you for the password(s) needed to access your private key. Use the password previously entered. (If you don't include the option “-signedjar newJarFilename.jar”, jarsigner will overwrite the original JAR file listed!)

Finally, you need to update the applet tag in the HTML file to refer to the new (signed) JAR file:

In the HTML file, update the archive to

```
ARCHIVE="SignedLab10Applet.jar"
```

Now within this directory, create an new folder called YourLastName_Sandbox.

Place only the signed jar and your html file in this directory

Test your html file

This hml file should now work locally and in any location.
Move YourLastName_Sandbox. to any location to veriffy

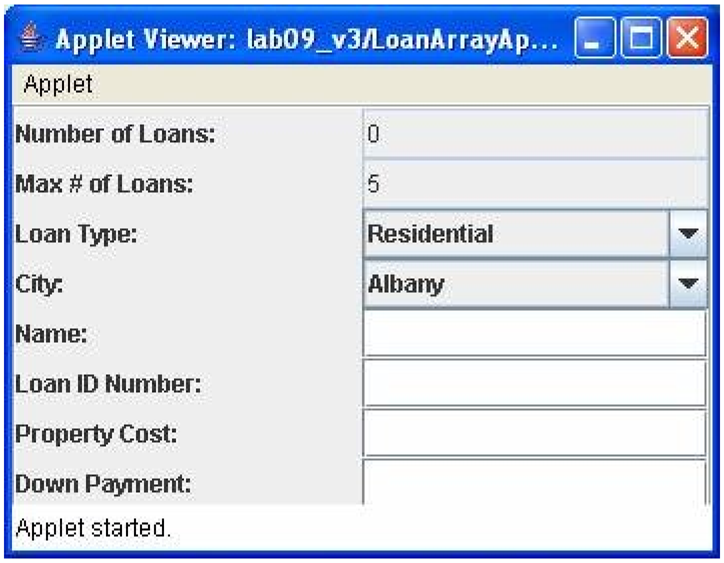
Modify the HTML page to adjust the size of the applet.

Be sure that you have copied the HTML from the build folder into your current Lab 9 folder contained in the source folder

In the left-side Files pane, double-click on the HTML file. As a result, its contents should be displayed in the NetBeans edit pane.

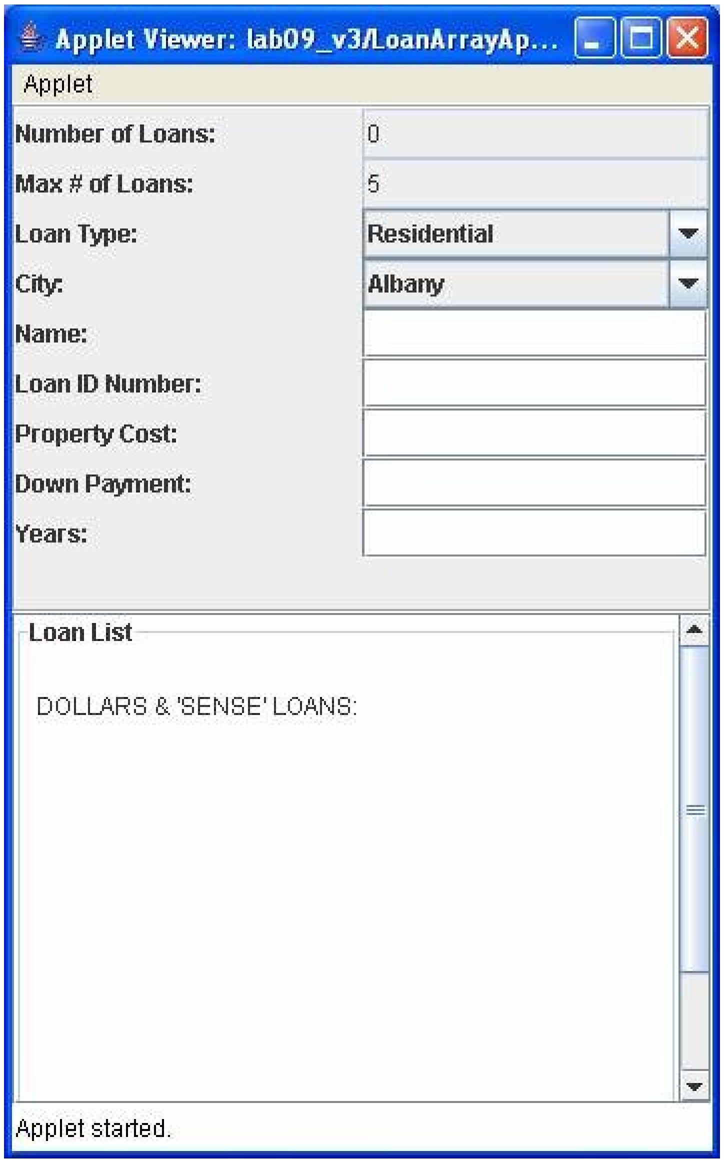
Change the height of the applet by changing the value of the height attribute to something that will show the entire applet within the HTML page. Use something like 500 for the height.

Reload the HTML page and be sure that the applet has been resized.



| | |
|---|---|
| Number of Loans: | 0 |
| Max # of Loans: | 5 |
| Loan Type: | Residential |
| City: | Albany |
| Name: | |
| Loan ID Number: | |
| Property Cost: | |
| Down Payment: | |

Applet started.

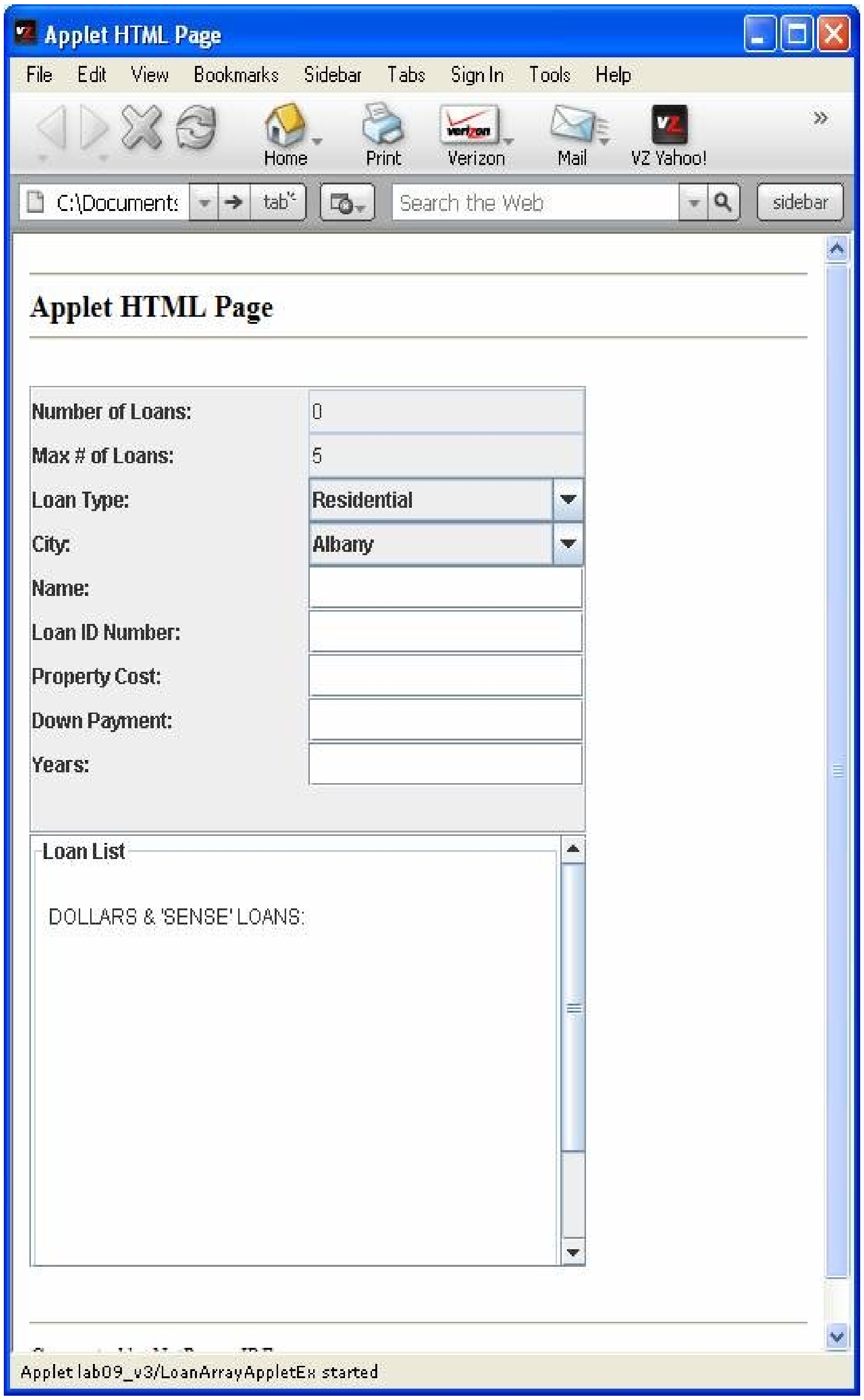


| | |
|---|---|
| Number of Loans: | 0 |
| Max # of Loans: | 5 |
| Loan Type: | Residential |
| City: | Albany |
| Name: | |
| Loan ID Number: | |
| Property Cost: | |
| Down Payment: | |
| Years: | |

Loan List

```
DOLLARS & 'SENSE' LOANS:
```

Applet started.


Applet HTML Page
File Edit View Bookmarks Sidebar Tabs Sign In Tools Help
Home Print Verizon Mail VZ Yahoo!
C:\Document: tab Search the Web sidebar

Applet HTML Page

| | |
|---|---|
| Number of Loans: | 0 |
| Max # of Loans: | 5 |
| Loan Type: | Residential |
| City: | Albany |
| Name: | |
| Loan ID Number: | |
| Property Cost: | |
| Down Payment: | |
| Years: | |

Loan List

DOLLARS & 'SENSE' LOANS:

Applet lab09_v3/LoanArrayAppletEx started

Be sure that the height attribute is large enough to display the entire applet.

Save the file and test it. To test it, right-click on the HTML file, select View,
and be sure that the HTML page is displayed in the web browser and the applet is
displayed and works correctly.

Modify the HTML page to add three header lines.

Bring up the HTML page in NetBeans for editing.

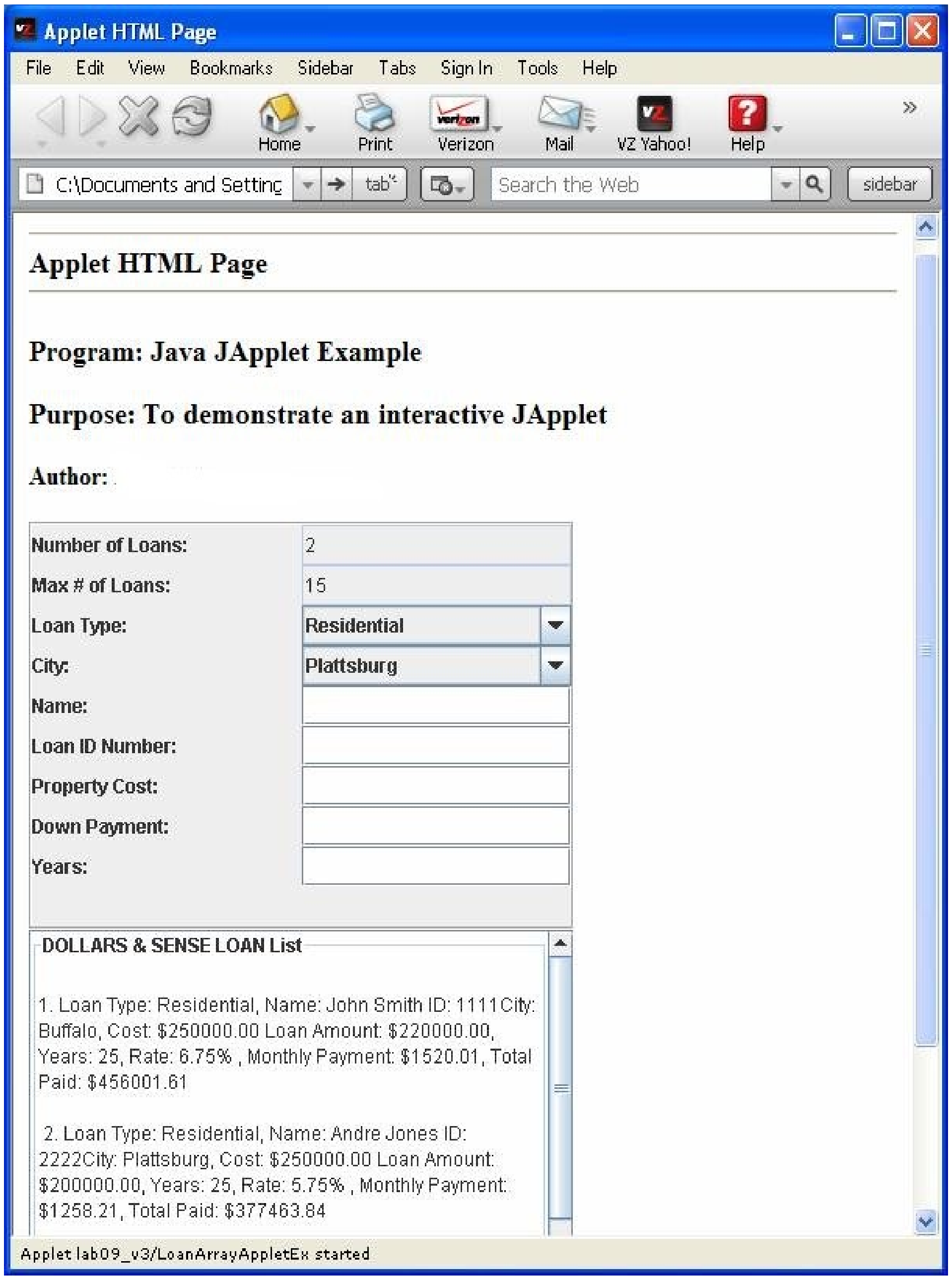
Add a heading on the HTML page, above the applet. Use the h3 tag. The heading should display the following: "Program: Java JApplet Example". The new HTML element should look like the following:

```
<h3>Program: Java JApplet Example</h3>
```

dd another heading on the next line that displays:
"Purpose: To demonstrate an interactive JApplet".
Make the header line look like the one created above.

Add another heading on the next line using an h4 tag that displays
"Author: " followed by your name.

Compile, execute, and debug the program.


Applet HTML Page
File Edit View Bookmarks Sidebar Tabs Sign In Tools Help
Home Print Verizon Mail VZ Yahoo! Help
C:\Documents and Setting
Search the Web
sidebar

Applet HTML Page

Program: Java JApplet Example

Purpose: To demonstrate an interactive JApplet

Author:

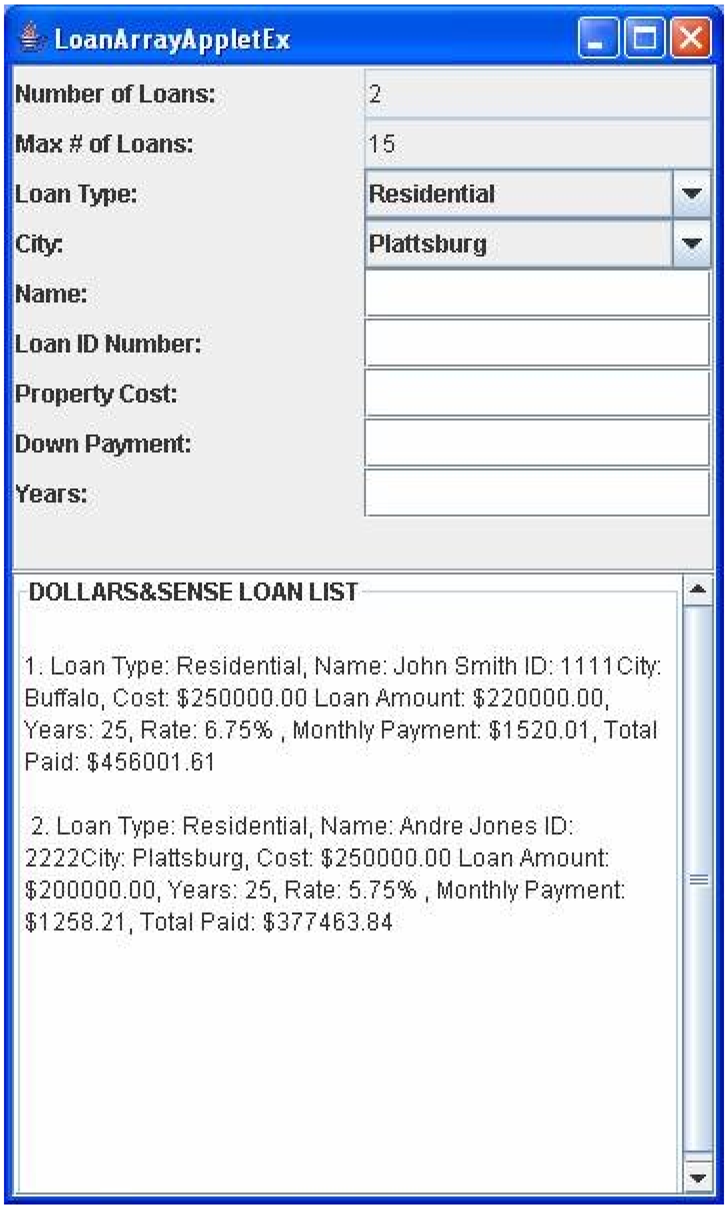
| | |
|---|---|
| Number of Loans: | 2 |
| Max # of Loans: | 15 |
| Loan Type: | Residential |
| City: | Plattsburg |
| Name: | |
| Loan ID Number: | |
| Property Cost: | |
| Down Payment: | |
| Years: | |

DOLLARS & SENSE LOAN List

```
1. Loan Type: Residential, Name: John Smith ID: 1111City:
Buffalo, Cost: $250000.00 Loan Amount: $220000.00,
Years: 25, Rate: 6.75% , Monthly Payment: $1520.01, Total
Paid: $456001.61

 2. Loan Type: Residential, Name: Andre Jones ID:
2222City: Plattsburg, Cost: $250000.00 Loan Amount:
$200000.00, Years: 25, Rate: 5.75% , Monthly Payment:
$1258.21, Total Paid: $377463.84
```

Applet lab09_v3/LoanArrayAppletEx started



Number of Loans: 2
Max # of Loans: 15
Loan Type: Residential
City: Plattsburg
Name:
Loan ID Number:
Property Cost:
Down Payment:
Years:

```
DOLLARS&SENSE LOAN LIST

1. Loan Type: Residential, Name: John Smith ID: 1111City:
Buffalo, Cost: $250000.00 Loan Amount: $220000.00,
Years: 25, Rate: 6.75% , Monthly Payment: $1520.01, Total
Paid: $456001.61

 2. Loan Type: Residential, Name: Andre Jones ID:
2222City: Plattsburg, Cost: $250000.00 Loan Amount:
$200000.00, Years: 25, Rate: 5.75% , Monthly Payment:
$1258.21, Total Paid: $377463.84
```

As usual create ReadMe.pdf file.

Create a file named ReadMe.pdf
In this document, insert your name at the top, and on the next line insert the assignment number
Then enter any comments regarding the assignment and your program.
Then insert at least two window captures of your JFrame window showing the inputs and output from the execution of the program.

The first window capture should be captured when you have entered data on the fourth loan; it should show the data in the JTextFields and selected items in the JComboBoxes.
The second window capture should show the updated list of loans in the JTextArea that is the result of hitting the Enter key in one of the JTextFields for this new fourth loan.

Zip the project folder and all its contents and YourLastName_Sandbox.

Change the name the zip file so that its name consists of your name along with the assignment number, as follows: "  LastName_ Lab_10_cs209.zip".
Do not use spaces in the name of the file, use underscores or  hyphens instead.

<u>Deliverables:</u>

Send to streller@ecc.edu an email this the exact subject

cs209_ Lab_10

In this email attached the above named zip file

LastName_ Lab_10_cs209.zip

Due Date :   7:00pm 13  November 2014