**CSCI 470/502 Assignment 8 - Java GUI Program Fall 2018**

**150 points**

For this assignment you may work in a team of one, two or three members. Once completed and a single solution agreed upon by the team member(s), ***only*** the team lead will submit the team's solution for the assignment. Remember that all team members' names must be included in the class documentation boxes.

In this assignment, you will re-implement the program from Assignment 4 but using Swing components to create a graphical user interface, or GUI, with which the travel agent can enter and receive data.

The Destination class should **remain the same** as that developed in Assignment 4. This reusability is one of the advantages of encapsulation, i.e., using object-oriented methodology.

In addition, the MileRedeemer class from Assignment 4 can be reused as well. Many methods of MileRedeemer can be invoked directly by your application, such as readDestinations() and redeemMiles(). But, there should be no part of your application that displays console-type output as you did in Assignment 4, not even the request for the name of the file!

The .jpg file included with the assignment on Blackboard shows a JFrame with two JPanels being used. Each of those may have nested JPanels for layout purposes. Each JPanel should have a background color set.

Recall the Color constructor:

Color(int red, int green, int blue)

where the arguments are in the range 0 to 255 and the higher the number, the lighter the color. Use colors that are not too intense and displeasing to a user, and be sure that the text is completely legible. You can learn how to add a Titled Border, etc. using the Java Tutorials [here](http://docs.oracle.com/javase/tutorial/uiswing/components/border.html).

The left panel contains a JList so that the user can go back and forth to look at the information of different tickets to different cities. The array of strings returned by the MileRedeemer method getCityNames() can be used to populate the JList. When a city in the JList is selected, its details (i.e., the members of its corresponding Destination object) should be displayed in the corresponding JTextFields. These JTextFields should not be editable. To listen for this event, implement the interface:

javax.swing.event.ListSelectionListener;

and provide the method

public void valueChanged(ListSelectionEvent e)

Add a new method to the MileRedeemer class to return the corresponding Destination object for a given city name, e.g.:

public Destination findDestination(String cityName)

The right panel takes in the accumulated miles using a JTextField and a JSpinner listing the possible departure months of the year. After the "Redeem Tickets" button is clicked, it outputs ticket details in a JTextArea, and the remaining miles in a JTextField. The components for output are not editable.

**The Input File**

Create a small GUI window to ask for the name of the .txt file of possible destinations. Use either JOptionPane or JFileChooser for this.

**How to populate the month Strings for the JSpinner**

The spinner's month Strings can be obtained using the following code. (Some logic is included to remove an extra, empty value.)

protected String[] getMonthStrings()   
{  
 String[] months = new java.text.DateFormatSymbols().getMonths();

int lastIndex = months.length - 1;

if (months[lastIndex] == null || months[lastIndex].length() <= 0)   
 {

String[] monthStrings = new String[lastIndex];  
 System.arraycopy(months, 0, monthStrings, 0, lastIndex);  
 return monthStrings;  
 }

else

{   
 return months;  
 }  
}

**Other Hints**

You may design a better looking GUI but it should have the same basic functionality and the same general look as that provided in the .jpg file.

Do ***not*** use a form or GUI builder in your IDE to build your GUI; you must code it all by hand.

You ***are*** allowed to use GridBagLayout, about which more can be found [here](https://docs.oracle.com/javase/tutorial/uiswing/layout/gridbag.html).

Any error messages or messages printed as a result of caught exceptions should be printed on the Java console.

**How to Submit the Assignment**

Submit each of your .java files on Blackboard as before. Once again, do ***not*** include your input file.