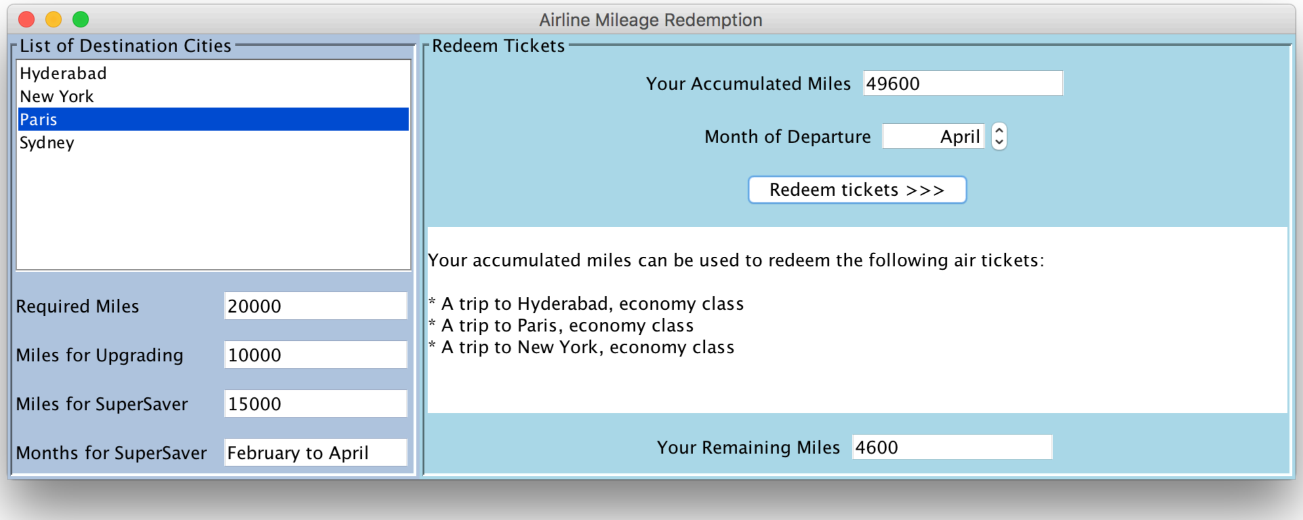
CSCI 470/680 Assignment 2

For this assignment, you will re-implement the Airline Ticket Redeemer from Assignment 1 as a Java GUI application with Swing components.

The `Destination` class should remain the same as in Assignment 1. This reusability is one of the advantages of encapsulation.

In addition, the `MilesRedeemer` class from Assignment 1 can be reused as well. Many methods of `MilesRedeemer` can be invoked directly, such as `readDestinations()` and the method that contains the redemption algorithm. Of course, the application will need to supply new code for obtaining the input and displaying the output.



Airline Mileage Redemption

List of Destination Cities

Hyderabad
New York
Paris
Sydney

Required Miles: 20000
Miles for Upgrading: 10000
Miles for SuperSaver: 15000
Months for SuperSaver: February to April

Redeem Tickets

Your Accumulated Miles: 49600
Month of Departure: April

```
Your accumulated miles can be used to redeem the following air tickets:

* A trip to Hyderabad, economy class
* A trip to Paris, economy class
* A trip to New York, economy class
```

Your Remaining Miles: 4600

Two `JPanel` objects are used in the application shown above (They have further nested `JPanel` objects for layout purposes). Each `JPanel` should have a background color set. Recall the `Color` constructor:

```
Color(int r, int g, int b)
```

where the arguments are in the range 0...255 and higher numbers are lighter colors. Use colors that are not too intense, and be sure that the text is legible. You can learn how to add a Titled Border, etc. using the Java Tutorials here.

The left (`LINE_START`) panel contains a `JList` so that the user can go back and forth to look at the information for different tickets to different cities. The array of strings returned by the `MilesRedeemer` 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 `JTextField` objects. These `JTextField` objects should not be editable. To listen for this event, the applet needs to implement the interface `javax.swing.event.ListSelectionListener`, and provide the method

```
public void valueChanged(ListSelectionEvent e)
```

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

```
public Destination findDestination(String cityName)
```

The right (CENTER) panel takes in the accumulated miles using a JTextField and a JSpinner. 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.

How to Populate the Months 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.)

```
private String[] getMonthStrings() {

    String[] months = new java.text.DateFormatSymbols().getMonths();
    int lastIndex = months.length - 1;
    if (months[lastIndex] == null || months[lastIndex].length() <= 0) {
        //last item empty
        String[] monthStrings = new String[lastIndex];
        System.arraycopy(months, 0, monthStrings, 0, lastIndex);
        return monthStrings;
    } else {
        //last item not empty
        return months;
    }
}
```

Other Hints

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

How to Submit the Assignment

Submission of the source code files should be done via Blackboard as in Assignment1.