



CST 183

Programming Assignment 9

Fall 2023
Instructor: T. Klingler

Objective

To build a complete working Java program that offer practice with a more involved Java graphical user interface, multiple classes, and a large data file.

Overview & Instruction

Write a Java application to simulate shipping tool for Michigan zip codes.

Your company maintains several shipping centers around Michigan. You need to build an application to calculate the shipping cost between one of the shipping centers and any other community in the state. Your shipping centers are in:

- University Center (48710)
- Mackinaw_City (49701)
- Grand Rapids (49501)
- Marquette (49855)
- Traverse_City (49684)

From post offices at these locations, your company can ship to any other post office in Michigan (except of course itself).

Build a JavaFX user interface that allows the user to enter a shipping center and a zip code for the product destination. Then, calculate and provide the shipping cost for the order. Create a drop-down list that includes the small list of shipping center cities. Also include an empty default choice in the drop-down list for an initial setting. Next, build a numerical key pad as an interface to key in the digits of the zip code. Include a (non-editable) text field to display the digits of the zip code as they are keyed in. Finally, include buttons to *Calculate*, *Clear*, and *Quit*. The *Calculate* button should determine the shipping cost and the *Clear* button should empty the zip code text field and reset the drop-down list to the empty "non-choice".

Upon entering a valid zip code, the user should receive a dialog box containing the post office name, the distance from the shipping source (see below), and the cost to ship. For simplicity, assume it requires only 6 cents per mile to ship a product.

For error checking, be sure that the zip code entered (1) has five digits, (2) exists in Michigan, and (3) does not match the selected shipping center. A simple message dialog can be used to display an error the user.

Your application requires a file [zipMlcity.txt](#) that contains all zip codes in Michigan including the location (latitude,longitude) and name of the post office. A sample line of input from the file will be:

```
48706 43.60880 -83.95300 MI Bay_City
```

(note: Western Hemisphere longitudes are negative).

Your application should be object oriented containing:

- The main application JavaFX GUI
- Include one class to store information for one zip code
- Include another class that will act as a "data manager" for the list of zip code info objects

The data manager class should perform the following tasks "behind the scenes":

- Read the raw zip code information from the provided data files.
- Store the information in one or more arrays within your class(es).
- Search for the name, latitude, and longitude for the zip code entered.
- Perform required distance calculations.
- Combine zip code and county information as needed for given user input.

To calculate the distance from Delta College, you will need to integrate the *great circle distance* formula. This formula is provided and demonstrated for you in the following example:

[DistanceCalc.java](#).

Note the following additional specifications:

- Avoid any use of **JOptionPane** dialogs and include JavaFX **Alert** actions instead.
- Underscores are added to city names to simplify input operations. Be sure to remove all underscores from city names for output.
- Be sure your GUI can handle an immediate mouse click without any user input (i.e. without crashing).
- Include at least one use of CSS formatting for your GUI
- Utilize basic arrays of objects and please avoid use of built-in Java data structures (like **ArrayLists**).
- There is code for keypads "out there" on the web. You are encouraged to build this feature without online referencing. If you do search for examples, be careful to borrow inspirations without directly copying code. If borrowing ideas for keypad from online sources, be sure to cite the source in your code comments.

Deliverables

Deliver the following to the online course management system **dropbox** as your final product:

- **Upload** your **source code** (.java) files

Notice

This is an individual assignment. You must complete this assignment on your own. You may not discuss your work in detail with anyone except the instructor. You may not acquire from any source (e.g., another student or an internet site), a partial or complete solution to a problem or project that has been assigned. You may not show another student your solution to an assignment. You may not have another person (current student, former student, tutor, friend, anyone) "walk you through" how to solve the assignment.
