# CS 150: Fall 2015
## Project III: Due: 11:55pm, Saturday December 12, 2015

Version as of: 19:21 Sunday 22$^{nd}$ November, 2015

## Introduction

We now conclude the semester with a project on delivering goods to the customers instead of having to pick it up at the Easton Farmer's market. Project 3 will extend your work from Project 2.

## Project Description

Many vendors of farm goods find it beneficial to have customers with signed seasonal contracts for goods, instead of finding out each week how many customers might want their goods. One strategy is for the vendor to deliver the goods to the customer thus saving the customer from having to drive to Easton each week.

The data you will be provided with consists of the following:

- an interconnection graph that shows the connecting roads between cities and the distance between the cities,

- a list of vendors and the cities where the vendor has supplies,

Your program will work as follows:

1. read (from a file - *citymap.txt*) a connectivity graph,

2. read (from a file) - *goods.txt*, the profit (per unit) from each good,

3. read (from a file) - *locations.txt*, the locations of each vendor's stores/warehouses/farms,

4. read (from the console) the cost per distance unit in the connectivity graph,

5. as in Project 2, generate the customers and assign a city to each customer. The cities are extracted from the file *citymap.txt*. The only command that you will need from Project 2 is **run**.

6. for each good (vendor), find the routes that will maximize the profit (or minimze the loss) from delivering to all the customers for that good,

### Project Data

Your program will read in data from the following files:

1. connectivity (graph information) from a file *citymap.txt*. The format of the file is:

   ```
   <city name> - <city name> <edge weight>
   ```

2. goods information from a file *goods.txt*. The format of the file is:

   ```
   <good> - <profit per unit>
   ```

3. location information from a file *locations.txt*. The format of the file is:

   ```
   <good> - <city1 name> <city2 name> ...
   ```

## Project Analysis and Report

Your report should include:

- a description of the algorithm(s) that you used,

- assumptions made,

- the quality of the solution, i.e., how good is the solution in your opinion?

- analysis of the program's performance on the provided set of data. How does the program's performance change as the size of the data (graph) changes and as the number of customers changes?

- a **complete** list of the people with whom you have discussed the project.

## Project Constraints

The following constraints apply to the project:

1. The project is to be completed individually. The only person you can consult about code issues is the instructor.

# Project Simplifications

You can simplify your project in the following ways:

1. only handle one good,

2. for each good/vendor, only have one warehouse/store/farm

# Grading

Your project will be graded on the following criteria (assuming the program compiles and runs):

1. correctness of the program

2. documentation (methods and classes) including javadoc

3. unit testing

4. object oriented design

5. quality of the evaluation and analysis