

MWC3 - Python Development 2025

Scenario

You are working for a small company that ships products across the United States. For their needs, they need to be able to calculate the distance between two locations utilizing zipcodes. Amongst other things, they will also need to be updating their current back end architecture from relying on .csv files to the utilization of a database they can easily manage. Your task is to provide their basic software needs for their calculations, updates on their infrastructure, and providing an easily usable interface.

Application Criteria:

1. User Interface - users must be able to interact with the program either through the console or a GUI with relevant interactions/commands provided to the user.
2. Users must be able to upload from a .csv file to intake media in bulk.
3. When applicable, users must be able to merge or modify existing data with incoming data.
4. Users must be provided with proper documentation of how to use the provided application.

Directions

For this competition, you have been tasked with analyzing data in the form of a **.csv** file. Stored in this file are various points of data regarding zip codes in the United States. Please take a moment to view the information stored inside of the given **zipcodes.csv** file. Utilizing that information, you must design a program that can perform the following tasks:

- Be able to properly import information from the **zipcodes.csv**.
- Be able to properly add and remove information based on a given zip code in the file.
- Be able to quickly look up the information on a location based on a given zip code.
- Be able to calculate the distance between zip codes in miles.
- Find all other zip codes in a given radius (in miles)
- Find the furthest away zip code from a given starting zip code within a given radius.
- Search for information for all zip codes relating to a given string.
- Find all zip code information by giving an abbreviated state name.
- Map the total distance between a series of zip codes progressing from one point to the next.

*** hint: for calculations of distance, utilize the longitude and latitude**

In addition to the program functionality, the following needs to be done as well.

- A total conversion from utilizing a **.csv** file to your choice of database.
- Import additional information from a **.csv** file to your database.
- Write program functionality to do the following with the database:
 - Add information for a new entry to the database using a zipcode, city, state, longitude, and latitude
 - Remove information of a given entry using only a zip code
 - Add the functionality of the previous section to utilize the databases instead of the **.csv** file.
- Provide a graphical user interface to do all of the above functionality.

Requirements

Basic Requirements (20%)

- method(s) used to take in direct user input to modify an existing **.csv** file or import in bulk from another file.
- Method(s) used to modify or create an existing file if one does not exist.
- Store information in an appropriate and consistent format.

Intermediate Requirements (30%)

- Effective use of classes to demonstrate object-oriented design principles.
- Effective use of exception handling when necessary.
- Sufficiently export clean data to a readable format.
- Provide the user with the ability to view the data stored in the readable format.

Advanced Requirements (30%)

- Export data to a database.
- Calculations for program functionality use the database created.
- Provide appropriate methods of modifying the database.

Documentation requirements (10%)

- Usage of industry standard documentation (i.e., PEP 8).

- Comments and docstrings where necessary.

Documentation Requirements for the Application (10%)

- Documentation of any dependencies.
- Documentation on how to use the application.
 - If using a console interface, include a description of various commands and how to use them.
 - If using a GUI, write a brief tutorial for the GUI.

Upon completion, please upload all necessary materials and documentation

Rubric

| Tier 1 | |
|---|---------------|
| Description | Points |
| Methods to read .csv files and extract usable information | /5pts |
| Methods to store and modify necessary information | /5pts |
| Store information in a consistent and usable manner | /10pts |
| | Total: |

| Tier 2 | |
|---|---------------|
| Description | Points |
| Effective use of classes to demonstrate object-oriented design principles | /15pts |
| Appropriate usage of exception handling | /3pts |
| A clean interface in which users can use the application effectively and easily | /10pts |
| Sufficiently export error free data to an external file if needed | /2pts |
| | Total: |

| Tier 3 | |
|---|---------------|
| Description | Points |
| Effectively exports all .csv data to a database | /10pts |
| Provided methods needed for functionality work with the provided database | /15pts |
| Appropriate methods available for database modification | /5pts |

| | |
|--|---------------|
| | Total: |
|--|---------------|

| Documentation & Usability | |
|--|---------------|
| Description | Points |
| Proper documentation of any dependencies required to run the application | /5pts |
| Proper documentation of how to use and run the provided application | /5pts |
| Proper usage of comments, white space, references, and docstrings | /10pts |
| | Total: |