CS 245: Object-Oriented Programming

Map Maker

Due April 29 @ 6pm

In this homework assignment, you will develop a database-driven mapping application. Given a set of city names, latitudes, and longitudes stored in a mysql database, your application will plot the cities in their proper locations on a Turtle Graphics canvas.

Your program will begin by printing a header and asking questions:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Map Maker \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter name of database server: **localhost**

Enter the name of the database: **map**

Enter the name of the table: **location**

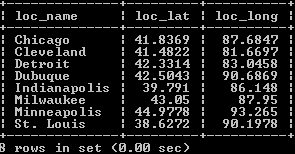
Enter your username: **root**

Enter your password: **mypassword**

After you answer the questions as shown above, your program will connect to the server, database, and table identified above using the username and password credentials you specified. It will pull the location data from the database, and then it will draw the following map based on that data.



The source of this data is a mysql database that you will have to create. That database contains the following data.



If your program is unable to connect to the database (perhaps because the user typed in the wrong names for the server, database, or tablename), then your program should print a message indicating that fact.

Your program must consist of the following classes, all stored in the same file named lastname\_mapmaker.py (where, of course, you replace “lastname” with your last name).

* Location: the model class that stores the name, latitude, and longitude of a location
* MySql\_Database: the class that represents our connection to the database and enables us to execute queries against it. It should have username, password, server, and database name properties, as well as a connect function to establish a connection and an execute function to perform a query against the data. The MySql\_Database class will also include a data member called *cursor* that allows us to interact with the data that was fetched through the most recent query.
* MySql\_Map\_Data\_Fetcher: a controller class that serves as the intermediary between our application and the database, allowing us to fetch the data and create from it a list of Location objects. MySql\_Map\_Data\_Fetcher establishes a connection with the database and executes a query to get the location data stored within it. Its get\_locations function (you could call it fetch instead to be consistent with what we did in class) will return a list of Location objects from the retrieved data. This list will serve as the basis for drawing the locations on the map. Like we did in class, you could have this descend from another class called MySql\_Fetcher.
* Banner, a class from your last assignment that descends from Shape, another class from your last assignment. The city names will be drawn on the screen as Banner objects.
* Turtle\_Map\_Controller: a controller class that will draw the locations as Banner objects using Turtle Graphics. It will have a draw\_map function that takes in the locations and places them on the map in the correct places.

You main code will have to create a MySql\_Map\_Data\_Fetcher. That object will, in turn, create the MySql\_Database object, which will create the connection. MySql\_Map\_Data\_Fetcher’s get\_locations function will ask the MySql\_Database to execute a query. That will cause the MySql\_Database’s cursor property to point to a list of returned rows. The get\_locations function will create a Location for each returned row and return these as a list. Your program will then create a Turtle\_Map\_Controller object that will open a Turtle Graphics window and use its draw\_map function to place the names of cities in the appropriate locations based on the contents of the list of Location objects.

Your program will be graded per the following rubric.

|  |  |
| --- | --- |
| You’ve defined the Location class and equipped it with name, location, and lat members as well as a constructor and a to\_string function. | Perfect: 3 points  Good: 2 points  Lacking: 1 point  No: 0 points |
| You’ve defined the Mysql\_Database class to include server, database, table name, user name, and password values. You’ve implemented server name as a property that will set the value of server to *localhost* if the user tries to set the server to blank. You’ve included an \_\_init\_\_ function that initializes these data members. You’ve included a connect function that establishes a connection, and an execute function which executes a query passed to the function. | Perfect: 3 points  Good: 2 points  Lacking: 1 point  No: 0 points |
| You’ve defined the MySql\_Map\_Data\_Fetcher class. It includes a dbase data member that refers to the associated MySql\_Database object which represents the connection to the database. MySql\_Map\_Data\_Fetcher must establish the connection. If it is unsuccessful, it should set dbase to None. Once a connection is established, it should be able to retrieve data. To do that, MySql\_Map\_Data\_Fetcher has a function called get\_locations that retrieves Location data from the database and returns the locations as a list of Location objects. | Perfect: 3 points  Good: 2 points  Lacking: 1 point  No: 0 points |
| You’ve defined the Shape and Banner classes so that you can draw the city names on the screen as Banner objects. | Perfect: 3 points  Good: 2 points  Lacking: 1 point  No: 0 points |
| You’ve defined the Turtle\_Map\_Controller which will draw the cities on that canvas using its draw\_map function. This will correctly position the cities on the canvas based on their latitudes and longitudes. | Perfect: 3 points  Good: 2 points  Lacking: 1 point  No: 0 points |
| In the main part, you’ve printed the heading, asked the appropriate questions, created the necessary objects, and tied everything together so that the map can be drawn. | Perfect: 3 points  Good: 2 points  Lacking: 1 point  No: 0 points |
| Your code is well-commented | 1 point |
| You’ve submitted your work with the required name. | 1 point |

So, this program is worth 20 points.