CODING STANDARD

In general make the code:

Readability

Reusability

Modifiability

Easy to debug

Name identifiers properly, ex:
variables: lowerCamelCase;
constants: UPPER_CASE;

Indent blocks of code, int main should have one indent and then more indents for any methods/functions, etc.

Commenting Code:
a. Use comments to describe what the program does.
b. Data tables, they should have the use of the variable.
ex: int count // used to keep count of integers.
c. Describe what each section is doing, input, processing, output.
ex: //INPUT - Grabbing integer from the user.
d. Describe any very complex parts of code.
e. Class header/Author box on top of source code, providing relevant information.

Commenting Program Description:
a. Program Title
b. General Description
c. Describe IN & OUT variables/constants
ex:
//input - used to get input name from user

//AR_SIZE - the size of the array

int main():
a. Main should have constants above any declarations.
b. Variables come next with a brief description.
c. Initialization of variables.

d. In/Out/Processing: Details of what is going on in each significant section.
e. Double-space each code segment.

Initialization
a. Initialization should happen during or after declarations for the sake of consistency. ex:
int count;   OR   int count = 0;

count = 0;

TEAM RULES

- USE GLOBAL VARIABLES! Very good practice!
- Use namespaces to declare identifiers with namespace scope.
- Capitalize the class names.
- Classes and methods/functions should have clear descriptions with pre and post-conditions.
- Inform other team members if you cannot finish your stories

Agile Stories

- Story 1 BASELINE STORY - As an admin I need to create a data structure to access and manipulate the data
    - Descriptions
        - The list needs to have 30 european cities and up to 6 food items per city and their respective prices
    - Assumptions
        - No prior requisites
    - Assignee
        - Muzzy
    - Tasks/Tests
        - Create datatable
        - Add all tables
        - Create a test function to make sure database outputs correct value
    - Definition of Done

        - The database has been created in DB editor with SQLITE.
        - The database contains multiple tables with a city and their dishes

- Story 2- As a prospective traveler I want to be able to see a list of European cities and their distances from berlin
    - Descriptions
        - There should be a UI that displays the table that the traveler wants to see nice and neatly
        - The UI should display correct values from database
    - Assumptions
        - The database has been created
        - Main login page needs to be established
        - Test function for database has run and all values are correct
    - Assignee
        - N/A
    - Tasks/Tests
        - Create a main page to display all cities and their distances from berlin
        - Create a function that displays a table from the database
        - Need to see distances from Berlin and a list of all cities
    - Definition of Done
        - We are done when the prospective traveler can see the list of cities and distances from Berlin.

- Story 3 - As an admin I need to be able to modify and edit the data table
    - Descriptions
        - There should be buttons on the admin page that enable the admin to delete certain cities and foods
        - Only the admin should be able to access these functions
    - Assumptions
        - Admin login ui page has been created
        - Database has been created
    - Assignee
        - N/A
    - Tasks/Tests
        - Create buttons on the admin page that allow the admin to change the data
    - Definition of Done
        - The admin can change the datatable easily
        - Only the admin can access these features
        - Display all of the information from the data table

- Story 4 - As a prospective traveler I should be able to plan a custom trip.
    - Descriptions
        - The program should allow a traveler to select the starting European city they wish to visit.Then allow the traveler to select all other cities they wish to visit.
        - The program should plan the trip with the selected cities in the most efficient order.
        - The program should also display the total distance for the custom trip.
    - Assumptions
        - The Program should correctly path a custom trip.
        - The data table has been created
        - The main ui has been created
    - Assignee
    - Tasks/Tests
        - Create the ability to set a custom course.
        - Have the custom course plotted correctly.
    - Definition of Done
        - Done when the user is able to set a custom course and have the program correctly plot the most efficient route.

- Story 5 - As a prospective traveler there should be a button that allows me to plan a trip to all 11 initial european cities
    - Descriptions:
        - The program should chard out a course starting at berlin and it should take the user to all other cities in the most efficient order (recursively gotta choose the city to the closest previous one)
        - Total distance traveled will be displayed
        - UI should have a button that plans a trip to all of the cities
    - Assumptions
        - Datatable has been established
        - The ui has been created
    - Assignee
        - N/A
    - Tasks/Tests
        - Create button on main ui page
        - Display total distance traveled
    - Definition of Done
        - A course to all 11 initial cities from berlin is planned

        - The total distance traveled will be displayed
- Story 6- As a prospective traveler I should have the ability to plan the shortest trip starting at Paris
    - Descriptions:
        - There should be a button on the ui that shows the shortest trip starting at paris
        - The total distance traveled should be displayed

    - Assumptions
        - Ability for user to visit European cities has been completed
        - The database has been established
        - The main ui has been created
    - Assignee
        - N/A
    - Tasks/Tests
        - Obtain the “number” of European cities to visit
        - Visit the “number” of cities specified (including Paris)
        - Plan the trip starting at Paris then visit the closest city to Paris, then visit the closest city to that city and so on (recursively choose the city closest to the previous city)
        - Display the total distance traveled
    - Definition of Done
        - We are done when the user can plan a trip from Paris that gives the shortest distance traveled.
        - There is a button on the ui that gives this option

- Story 7 - As a prospective customer, I should be able to visit all 13 cities starting at Berlin
    - Descriptions
        - Provide the capability to visit the 13 European cities starting at Berlin.
        - Plan the trip starting at Berlin then visit each of the other European cities in the most efficient order (recursively choose the city closest to the previous city)
        - Display the total distance traveled
        - Allow the traveler to purchase multiple traditional food items when visiting the European cities
    - Assumptions
        - The database has been established
        - The ui has been created

        - The ability to chart a course starting at berlin and visiting the rest of the initial 11 cities has been created
    - Assignee
        - N/A
    - Tasks/Tests
        - Access the database
        - Enable ability to see all of the food and their prices
        - Ability to purchase food at each city
    - Definition of Done
- Story 8 -As a prospective traveller I should be able to purchase multiple traditional food items.

    - Descriptions
        - The UI should display the data
        - There should be a button to allow the user to buy multiple food items
    - Assumptions
        - We must have the database complete with the food items done
        - We must let the user have a “cashier” type checkout in the GUI
        - Must have a test function created to make sure database values are correct
    - Assignee
        - Blake Dickerson
    - Tasks/Tests
        - Run a test to make sure the database values are correct
        - Create a cashier method
        - Allow user to checkout
        - Display total amount purchased in each city
        - Display grand total for all cities visited
    - Definition of Done
        - We are done once the user can purchase multiple food items and see the total purchased in each city and the grand total for all cities.
- Story 9 As a team member I need to create a test plan for the project
    - Descriptions
        - Test plan for the project needs to be created
        - Either white box/ black box testing needs to be done

    - Assumptions
        - We must have some sort of code done so we can test it
        - We must target a specific part of code that needs work
    - Assignee

        - All of us
    - Tasks/Tests
        - Decide on what kind of test we want to do (black box/white box)
        - Decide on how we implement the test
    - Definition of Done
        - We are done when each team member has done a test plan

- Story 10 -As a team member I need to create the documentation for the code along with the big(0) notation for 5 member functions
    - Descriptions
        - 5 member functions need to have the big(0) notation
        - Doxygen needs to be used for the code coments

    - Assumptions
        - We must have 5 member functions complete
        - We must know what big(O) is
        - We must know how to figure out the big(O) for the given function
    - Assignee
        - N/A
    - Tasks/Tests
        - Pick 5 member functions
        - Run the math for big(O) on the functions
        - Determine if we can make the functions better
    - Definition of Done
        - We are done when the 5 member functions have the big(O) notation done
- Story 11 -As a team member I need to create the uml diagrams along with the case diagrams for the code
    - Descriptions
        - The uml diagrams need to have a class diagram
        - Test cases
        - And use cases
        - Activity diagram needs to be established as well
    - Assumptions
        - We need to have all the classes complete
        - We need to have all documentation complete
    - Assignee
        - N/A
    - Tasks/Tests
        - Three use cases

        - One activity diagram
        - Three state diagrams with your project
    - Definition of Done
        - We are done when we have all the uml diagrams complete

- Story 12 - As a prospective traveler, all of the information should be seen on a UI
    - Description:
        - There should be a main page created on qt
        - The main page should have an admin login section and a button for the client
    - Assumptions:
        - No prior assumptions
    - Assignee
        - N/A
    - Tasks/Test
        - Create main page with a login button for the admin
    - Definition of Done
        - The login page is set up and the admin login button takes you to the admin page

BACKLOG

MEETING LOG

8-31-20 The agile stories were being created and the team rules were being outlined
9- 2 -20 The agiles stories were being modified and changed
9-9-20   Planning poker was demonstrated however, the stories needed to be reworked.
9 - 14 -20