Exercise 1.7: Finalizing Your Python Program

Learning Goals

- Interact with a database using an object-relational mapper
- Build your final command-line Recipe application

Reflection Questions

1. **What is an Object Relational Mapper and what are the advantages of using one?**
   An Object-Relational Mapper (ORM) converts the contents and structure of a database into classes and objects you can interact with.
   There is no need to know how SQL queries work (and SQL queries can work differently depending on the DBMS); without an ORM transferring and converting databases can be time consuming and lead to syntax errors, because you have to manually adapt your SQL queries to the new DBMS.

2. **By this point, you've finished creating your Recipe app. How did it go? What's something in the app that you did well with? If you were to start over, what's something about your app that you would change or improve?**
   I really enjoyed exercise 1.6. I think I did well working with databases in Python.
   I would improve:
   - Handling exceptions and errors.
   - Code legibility and efficiency – I repeat several validations throughout the code (*e.g.* in the create_recipe() and edit_recipe() functions I repeat the input validations).

3. **Imagine you're at a job interview. You're asked what experience you have creating an app using Python. Taking your work for this Achievement as an example, draft how you would respond to this question.**
   The Specialization Course I chose during the CareerFoundry Program was "Python for Web Developers". In this course I had to build a command line version of a Recipe app that creates, reads, updates and deletes recipes, as well as searching for them by ingredients.

   In summary I learned about:
   - Installing Python on Windows, creating and managing virtual environments, and using pip to install and managing packages.
   - Data types (and methods), operators, and functions; how to create structures for storing individual recipes (dictionaries) and storing all recipes (list), and by creating my first script that used if-elif-else statements, for loops and functions to take recipes from the user and display them.

- File handling in Python, namely taking data from the user and writing it in a binary file, and vice-versa.
- Using Python's exception handling features to handle common errors (try-except block, finally block, else block, handling exception in file handling).
- Object-oriented programming (OOP) in Python and how to build custom classes for recipes, which contains its own data attributes for name, ingredients, cooking time, and difficulty, as well as other custom methods to interact with this data.
- Interacting with a MySQL database (insert, read, update, and delete recipes, as well as searching for recipes by ingredient).
- Using an Object Relational Mapper from SQLAlchemy to manage the contents of the database from the application.

4. **You've finished Achievement 1! Before moving on to Achievement 2, take a moment to reflect on your learning in the course so far:**
    a. **What went well during this Achievement?**
    b. **What's something you're proud of?**
    c. **What was the most challenging aspect of this Achievement?**
    d. **Did this Achievement meet your expectations? Did it give you the confidence to start working with your new Python skills?**
    e. **What's something you want to keep in mind to help you do your best in Achievement 2?**

I really enjoyed programming in Python. In my opinion, it's a programming language that is easy to learn. I like its syntax; it's very easy to read/write (and understand).
The most challenging aspect of this achievement was working with files – it didn't come naturally to me. Working with databases seems more logical.