10 Prove Assignment: Handling Exceptions

Purpose

Prove that you can write a Python program that handles exceptions, including FileNotFoundError, PermissionError, and KeyError.

Problem Statement

A local grocery store subscribes to an online service that enables its customers to order groceries online. After a customer completes an order, the online service sends a CSV file that contains the customer's requests to the grocery store. The store needs you to write a program that reads the CSV file and prints to the terminal window a receipt that lists the purchased items and shows the subtotal, the sales tax amount, and the total.

Assignment

During the prove milestone for the previous lesson, you wrote the part of this program that reads and processes two CSV files, one named `products.csv` that contains a catalog of products and one named `request.csv` that contains a customer's order. During this prove assignment, you will add code to finish printing a receipt and to handle any exceptions that might occur while your program is running. Specifically, your program must do the following:

1. Print the store name at the top of the receipt.
2. Print the list of ordered items.
3. Sum and print the number of ordered items.
4. Sum and print the subtotal due.
5. Compute and print the sales tax amount. Use 6% as the sales tax rate.
6. Compute and print the total amount due.
7. Print a thank you message.
8. Get the current date and time from your computer's operating system and print the current date and time.
9. Include a `try` block and `except` blocks to handle `FileNotFoundError`, `PermissionError`, and `KeyError`.

Helpful Documentation

- The prove milestone of the previous lesson describes the two CSV files that your program must process.
- The preparation content for this lesson explains how to handle exceptions.

- The `datetime.now() method` from the standard Python `datetime` module will get the current date and time from your computer's operating system. Here is an excerpt from the official documentation for the `datetime.now` method:

> `datetime.now(tz=None)`
>
> Return the current local date and time.
>
> *tz* is optional, but if it is not `None`, it must be a `tzinfo` (time zone information) object

These two Microsoft videos explain how to use methods from the standard `datetime` module.

 Date data types (8 minutes)
 Demonstration: Dates (9 minutes)

The following Python code imports the `datetime` class from the `datetime` module and calls the `datetime.now` method to get the current date and time from a computer's operating system. Then it uses an f-string to format and print the current date and time.

```
 1  # Import the datetime class from the datetime
 2  # module so that it can be used in this program.
 3  from datetime import datetime
 4
 5  # Call the now() method to get the current
 6  # date and time as a datetime object from
 7  # the computer's operating system.
 8  current_date_and_time = datetime.now()
 9
10  # Use an f-string to print the current
11  # day of the week and the current time.
12  print(f"{current_date_and_time:%A %I:%M %p}")
```

```
> python example_1.py
Tuesday 01:23 PM
```

After the computer executes line 8 in the above code, the variable *current_date_and_time* will hold the current date and time. Within the f-string at line 12, the string sequences that begin with the percent symbol (%) are called format codes. The format codes and their meaning are listed in the official Python `datetime` reference. As shown in the terminal window above, the previous example code will print the current day of the week and time to the terminal window.

Help from a Tutor

As a BYU-Idaho campus or online student you can get help from a tutor to complete your CSE 111 assignments. Each tutor is a current BYU-Idaho student employed by BYU-Idaho. Meeting with a tutor is free. It will not cost you any money to meet with a tutor. To get help from a tutor, you simply make an appointment and then meet with the tutor. Campus students meet with tutors in

the tutoring center. Online students meet with tutors in Zoom. To make an appointment, follow the instructions in the course tutoring guide.

Testing Procedure

Verify that your program works correctly by following each step in this testing procedure:

1. Run your program and verify that it prints a receipt formatted similarly to the one shown below. Your program must print the current date and time with exactly the same formatting as shown below. Also, verify that your program computes the number of items, subtotal, sales tax, and total as shown below.

```
> python receipt.py
Inkom Emporium

wheat bread: 2 @ 2.55
1 cup yogurt: 4 @ 0.75
32 oz granola: 1 @ 3.21
twix candy bar: 2 @ 0.85
1 cup yogurt: 3 @ 0.75

Number of Items: 12
Subtotal: 15.26
Sales Tax: 0.92
Total: 16.18

Thank you for shopping at the Inkom Emporium.
Wed Nov  4 05:10:30 2020
```

2. Verify that the `except` block to handle `KeyError` that you added to your program works correctly by doing the following:

   a. Temporarily add the following line to the end of your `requests.csv` file and then save the file.

   ```
   R002,5
   ```

   b. Run your program again and verify that it prints an error message like the one shown below.

   ```
   > python receipt.py
   Error: unknown product ID in the request.csv file
   'R002'
   ```

   Hint: if you wrote an `except` block in your program to handle `KeyError` and added "R002,5" to the `requests.csv` file and saved the `requests.csv` file and ran your program but your program isn't raising a `KeyError`, then look in your program to see if you wrote an `if` statement before the statement that finds a value in the *products* dictionary. Look for code similar to this:

   ```
       if prod_num in products_dict:
           prod_info_list = products_dict[prod_num]
   ```

```
⋮
```

If your program contains an `if` statement similar to the one above, then the `if` statement is probably preventing your program from raising a `KeyError`. Delete the `if` statement and unindent the lines of code inside the `if` statement and test your program again.

3. Verify that the `except` block to handle `FileNotFoundError` that you added to your program works correctly by doing the following:
    a. Temporarily delete or rename the `products.csv` file.
    b. Run your program again and verify that it prints an error message like the one shown below.

```
> python receipt.py
Error: missing file
[Errno 2] No such file or directory: 'products.csv'
```

Exceeding the Requirements

If your program fulfills the requirements for this assignment as described in the previous prove milestone and the Assignment section above, your program will earn 93% of the possible points. In order to earn the remaining 7% of points, you will need to add one or more features to your program so that it exceeds the requirements. Here are a few suggestions for additional features that you could add to your program if you wish.

- Write code to discount the product prices by 10% if today is Tuesday or Wednesday.
- Write code to discount the product prices by 10% if the current time of day is before 11:00 a.m.
- Write code to print a coupon at the bottom of the receipt. Write the code so that it will always print a coupon for one of the products ordered by the customer.
- Write code to print at the bottom of the receipt an invitation for the customer to complete an online survey.

Submission

To submit your program, return to I-Learn and do these two things:

1. Upload your program (the .py file) for feedback.
2. Add a submission comment that specifies the grading category that best describes your program along with a one or two sentence justification for your choice. The grading criteria are:
    a. Some attempt made

b. Developing but significantly deficient
c. Slightly deficient
d. Meets requirements
e. Exceeds requirements