Assignment 10 – Dictionaries

Learning Objective

Create a dictionary to manage items on a menu. Allow the user to insert, update, and delete items from the dictionary.

Assignment Description

Write a program that manages a menu. The menu is represented using a dictionary. The **keys** in the dictionary are names of food items (strings), and each **value** is the price (float) of the food item.

When you display the menu, the price will be printed as a float. You do not need to worry about displaying 2 numbers after the decimal point. It just depends on the prices assigned to the food item. All of the following examples are acceptable:

```
Pizza costs $10.0
Hamburger costs $8.99
Tea costs $2.5
```

Steps

1. In PyCharm (Community Edition), open your existing ITP115 project.
2. Under the Assignments directory, create a new directory called **a10_*last_first*** where *last* is your last/family name and *first* is your preferred first name. Use all lowercase letters.
3. In the directory, create a new Python file called **assignment10.py**.
4. At the top of the file, put comments in the following format and replace the name, email, and section with your actual information:

```
# Name, USC email
# ITP 115, Fall 2022
# Section: number or nickname
# Assignment 10
# Description:
# Describe what this program does.
```

5. Define the **displayChoices()** function.

This content is protected and may not be shared, uploaded, or distributed.

- Parameter: None
- Return value: None
- Print out the choices to the user:

```
Manage the Menu
   a) Add a menu item
   b) Update the price
   c) Show the price
   d) Delete a menu item
   e) Show the menu
   x) Exit
```

- This function will be called in the main() function.

6. Define the **isFloat(numStr)** function.
   - Parameter: numStr is a string containing input from the user
   - Return value: a Boolean (True or False) depending on whether numStr contains a float
   - To see if numStr contains a float, remove one decimal point (i.e., period) and then check to see if the string only contains digits.
   - Use the str.replace(oldValue, newValue, count) method to replace one decimal point (".") with the empty string.
   - Use the str.isdigit() method to see if the string only contains digits.
   - This function will be called in functions where the user enters the price of a food item.
7. Define the **addItem(menuDict)** function.
   - Parameter: menuDict is a dictionary representing the menu
   - Return value: None
   - Get input from the user for the food item. Here is an example with the user input in green text:

```
Enter a food item to add:   pasta
```

   - The user may add extra spaces before and after the food item. Remove them.
   - The user may not enter the food item in the correct case. Use the str.title() method to convert to title case which is how the food is saved in the menu.
   - Check if the food item is already in the dictionary. If so, print a message to the user. Here's an example when the user entered "pizza":

```
Pizza is already on the menu
```

- If the food item is not in the dictionary, get input from the price from the user. Loop until the user enters a valid float. Use the isFloat() function. Here is an example with the user input in green text:

```
Enter the price: ten
Enter the price: 10,99
Enter the price: 10.99
```

- Once you have a valid string that contains a float, convert it to a float. Add the food item and price into the dictionary. Display a message to the user. Here is an example:

```
Pasta has been added to the menu
```

- This function will be called in the main() function.

8. Define the **updatePrice(menuDict)** function.
   - Parameter: menuDict is a dictionary representing the menu
   - Return value: None
   - Get input from the user for the food item. Here is an example with the user input in green text:

```
Enter a food item to update:   pizza
```

   - Check if the food item is in the dictionary. If so, display a message to the user with the current price. Here is an example:

```
Pizza costs $10.99
```

   - Then get a price for the food item. Use a loop to make sure it is a float. Here is an example with the user input in green text:

```
Enter the price:   8.99
Enter the price: 8.99
```

   - After you get a valid price, update the dictionary and display a message to the user. Here is an example:

```
Pizza now costs $8.99
```

   - If the food item is not in the dictionary, display a message to the user. Here is an example:

```
Salad is not on the menu
```

- This function will be called in the main() function.

9. Define the **showPrice(menuDict)** function.
   - Parameter: menuDict is a dictionary representing the menu
   - Return value: None
   - Get input from the user for the food item. Here is an example with the user input in green text:

```
Enter a food item to find:  CHICKEN
```

   - Check if the food item is in the dictionary. If so, display a message with the food item and price. Here is an example:

```
Chicken costs $9.99
```

   - If the food item is not in the dictionary, display a message. Here is an example:

```
Salad is not on the menu
```

   - This function will be called in the main() function.

10. Define the **deleteItem(menuDict)** function.
    - Parameter: menuDict is a dictionary representing the menu
    - Return value: None
    - Get input from the user for the food item. Here is an example with the user input in green text:

```
Enter a food item to find: soda
```

    - Check if the food item is in the dictionary. If so, delete the item from the dictionary. Then display a message. Here is an example:

```
Soda was deleted from the menu
```

    - If the food item is not in the dictionary, display a message. Here is an example:

```
Salad is not on the menu
```

    - This function will be called in the main() function.

11. Define the **showMenu(menuDict)** function.

- Parameter: menuDict is a dictionary representing the menu
- Return value: None
- Loop through the menu to display each item and price. Here is an example:

```
Chicken costs $9.99
Pizza costs $8.99
Hamburger costs $8.99
Coffee costs $4.49
Tea costs $2.49
```

- This function will be called in the main() function.

12. Define and call the **main()** function.
    - Create a dictionary with at least 3 food items (strings) and their corresponding prices (floats). Make sure that the food item uses title case. Here are some potential food items:
    - Create a do-while loop to repeat while the choice is not equal to "x".
    - In the while loop, call the displayChoices() function and get input from the user. Allow the user to enter in upper or lower case. Here is an example with the user input in green text:

```
Choice: a
```

    - Use branching to call the appropriate function based on the choice. If the user enters "x" or "X", then print out a message such as "Thank you". If the choice is not valid, then print a message. Here is an example:

```
Invalid choice
```

    - Don't forget to call the main() function.
13. Be sure to comment your code. This means that there should be comments throughout your code. Put a comment block before each function stating the parameters, return values, and what that function does. Points will be deducted for not having comments.
14. Follow coding conventions. You should use lowerCamelCase or snake_case for variable names. You are welcome to create any variables that you need.
15. Test the program. Look at the Sample Output below. Assignments that do not run are subject to 20% penalty.

16. Prepare your submission:
    - Find the **a10_*last_first*** folder on your computer and compress it. This cannot be done within PyCharm.
    - On Windows, use **File Explorer** to select the folder. Right click and select the Send to -> Compressed (zipped) folder option. This will create a zip file.
    - On Mac OS, use **Finder** to select the folder. Right click and select the Compress "*FolderName*" option. This will create a zip file.
17. Upload the zip file to your Blackboard section:
    - On Blackboard, navigate to the appropriate item.
    - Click on the specific item for this assignment.
    - Click on the **Browse Local Files** button and select the zip file.
    - Click the **Submit** button.

Grading

- This assignment is worth 40 points.
- Make sure that you the program runs. Points will be taken off if the graders have to edit the source code to test your program.
- Make sure to submit your assignment correctly as described above. Points will be taken off for improper submission.

Item	Points
displayChoices()	3
isFloat()	3
addItem()	6
updatePrice()	6
showPrice()	4
deleteItem()	6
showMenu()	4
main()	8
Total	**40**

Sample Output

```
Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: e
Chicken costs $9.99
Pizza costs $10.99
Hamburger costs $8.99
Coffee costs $4.49
Tea costs $2.49
Soda costs $3.49

Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: A
Enter a food item to add:   pasta
Enter a price: ten
Enter a price: 10,99
Enter a price: 10.99
Pasta has been added to the menu

Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: b
Enter a food item to update:   pizza
Pizza costs $10.99
Enter the price:   8.99
Enter the price: 8.99
Pizza now costs $8.99
```

```
Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: c
Enter a food item to find:  CHICKEN
Chicken costs $9.99

Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: D
Enter a food item to delete: soda
Soda was deleted from the menu

Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: a
Enter a food item to add: pizza
Pizza is already on the menu

Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: b
Enter a food item to update:  salad
Salad is not on the menu

Manage the Menu
   a) Add a menu item
```

```
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: C
Enter a food item to find: SALAD
Salad is not on the menu

Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: d
Enter a food item to delete:  salad
Salad is not on the menu

Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: f
Invalid choice

Manage the Menu
   a) Add a menu item
   b) Update a price
   c) Show price
   d) Delete a menu item
   e) Show menu
   x) Exit
Choice: X
Thank you
```