

Deborah Barndt

ITMD 513 Open Source Programming

Professor Dr. Sam

Hw8

3-26-19

```
1  '''
2  Deborah Barndt
3  3-25-19
4  EmailDictionary.py
5  hw8: Name and Email Dictionary
6
7  This program will keep names and email addresses in a dictionary as key-value
8  pairs. It will display a menu that lets the user look up a person's email
9  address, add a new name and email address, change an existing email address,
10 and delete an existing name and email address. The program will bind the
11 dictionary and save it to a file when the user exits the program. Each time the
12 program starts, it will retrieve the dictionary from the file and unbind it.
13 The program will run continuously until the user selects the option exit from
14 the menu.
15
16 Written by Deborah Barndt.
17 '''
18
19 import pickle
20
21 # Function to check to for an existing file or dictionary.
22 def checkExist():
23     try:
```

```
24     # Open the file contacts.dat in read mode.
25     contacts = open('contacts.dat', 'rb')
26
27     # Unpickle the contacts in the dictionary.
28     contactDict = pickle.load(contacts)
29
30     # Close the contacts.dat file.
31     contacts.close()
32
33     # If no dictionary exists, ask user if they would like to create one.
34     if (contactDict == {}):
35         print('There are no contacts in your dictionary.')
36
37         create = input('Would you like to add a contact? ')
38
39         # If user types yes, go to addContact().
40         if ('yes' in create.lower()):
41             addContact()
42
43         else:
44             return
45
46     else:
47         return contactDict
48
49     # If there is a reading error or the end of the file has been reached, give response to user.
50 except EOFError:
51     print('There are no contacts in the dictionary.')
52
```

```
53     # Ask the user if they would like to create a contact.
54     create = input('Would you like to add a contact? ')
55
56     # If user types yes, go to addContact().
57     if ('yes' in create.lower()):
58         addContact()
59
60     else:
61         return
62
63     # If no file exists, give response to user.
64     except FileNotFoundError:
65         print('There are no contacts in the dictionary.')
66
67     # Ask the user if they would like to create a contact.
68     create = input('Would you like to add a contact? ')
69
70     # If user types yes, go to addContact().
71     if ('yes' in create.lower()):
72         addContact()
73
74     else:
75         return
76
77
78     # Function to find a contact in the dictionary.
79     def findContact():
80         print('\nFind Contact\n')
81
```

```
82     # Check to see if the file or dictionary exists.
83     contactDict = checkExist()
84
85     # If the dictionary does not exist, return to main menu.
86     if (contactDict == None):
87         return
88
89     else:
90         try:
91             # Ask user to enter the contact name, all, or cancel.
92             search = input('Enter the contact name, all, or cancel: ')
93
94             # If user types cancel, return to main menu.
95             if (search.lower() == 'cancel'):
96                 return
97
98             elif (search.lower() == 'all'):
99                 # Print an organized list of contacts in the dictionary.
100                 for name, email in sorted(contactDict.items()):
101                     print('Name: ', name, '\nEmail: ', email, '\n')
102                 return
103
104             else:
105                 print('Name: ', search, '\nEmail: ', contactDict[search])
106                 return
107
108         # If invalid entry, respond to user.
109         except KeyError:
110             print('The contact does not exist in the dictionary.')
```

```
111
112     # Ask the user if they would like to check the list.
113     checkList = input('Would you like to check the list? ')
114
115     # If user types yes, return organized sorted list.
116     if ('yes' in checkList.lower()):
117         for name, email in sorted(contactDict.items()):
118             print('Name: ', name, '\nEmail: ', email, '\n')
119
120     else:
121         return
122
123
124 # Function to add a contact to the dictionary.
125 def addContact():
126     print('\nAdd Contact\n')
127
128     # Try to open the file.
129     try:
130         # Open the file contacts.dat in read mode.
131         contacts = open('contacts.dat', 'rb')
132
133         # Unpickle the contacts in the dictionary.
134         contactDict = pickle.load(contacts)
135
136         # Close the contacts.dat file.
137         contacts.close()
138
139     # Ask user to enter new contact name or cancel.
```

```
140     name = input('Enter new contact name or cancel: ')
141
142     # If user types cancel, return to main menu.
143     if (name.lower() == 'cancel'):
144         return
145
146     else:
147         # Ask user to enter new contact email address.
148         email = input('Enter new contact email address: ')
149         contactDict.update({name : email})
150
151         # Open the file contacts.dat in write mode.
152         contacts = open('contacts.dat', 'wb')
153
154         # Pickle the dictionary, and write it to the file.
155         pickle.dump(contactDict, contacts)
156
157         # Close the contacts.dat file.
158         contacts.close()
159
160     # If there is no file, then give response to user.
161     except:
162         print('There are no contacts in the dictionary.')
163
164     # Ask user to enter new contact name or cancel.
165     name = input('Enter new contact name or cancel: ')
166
167     # If user types cancel, return to main menu.
168     if (name.lower == 'cancel'):
```

```
169         return
170
171     else:
172         # Ask user to enter new contact email address.
173         email = input('Enter new contact email address: ')
174         contactDict = {}
175         contactDict.update({name : email})
176
177         # Open the file contacts.dat in write mode.
178         contacts = open('contacts.dat', 'wb')
179
180         # Pickle the dictionary, and write it to the file.
181         pickle.dump(contactDict, contacts)
182
183         # Close the contacts.dat file.
184         contacts.close()
185         return
186
187
188 # Function to edit a contact in the dictionary.
189 def editContact():
190     print("\nEdit Contact\n")
191
192     # Check to see if the file or dictionary exists.
193     contactDict = checkExist()
194
195     # If the dictionary does not exist, return to main menu.
196     if (contactDict == None):
197         return
```

```
198
199     else:
200         # Ask the user to enter a contact name to edit or cancel.
201         edit = input('Enter contact name or cancel: ')
202
203         # If user types cancel, return to main menu.
204         if (edit.lower() == 'cancel'):
205             return
206
207     else:
208         if (edit not in contactDict):
209             create = input('The contact is not in the dictionary. Would you like to create this new contact? ')
210
211             # If user types yes, ask user to enter new email address.
212             if ('yes' in create.lower()):
213                 newEmail = input('Enter new email address: ')
214                 contactDict[edit] = newEmail
215                 print('Name: ', edit, '\nEmail: ', contactDict[edit])
216
217             # Open the file contacts.dat in write mode.
218             contacts = open('contacts.dat', 'wb')
219
220             # Pickle the dictionary, and write it to the file.
221             pickle.dump(contactDict, contacts)
222
223             # Close the contacts.dat file.
224             contacts.close()
225             return
226
```



```
227         else:
228             return
229
230     # If contact exists, allow user to edit.
231     else:
232         # Ask the user for the new email address for contact.
233         newEmail = input('Enter new email or cancel: ')
234
235         # If user types cancel, return to main menu.
236         if (newEmail.lower() == 'cancel'):
237             return
238
239         else:
240             contactDict[edit] = newEmail
241             print('Contact email updated.\n')
242             print('Name: ', edit, '\nEmail: ', contactDict[edit])
243
244             # Open the file contacts.dat in write mode.
245             contacts = open('contacts.dat', 'wb')
246
247             # Pickle the dictionary, and write it to the file.
248             pickle.dump(contactDict, contacts)
249
250             # Close the contacts.dat file.
251             contacts.close()
252             return
253
254
255     # Function to delete a contact in the dictionary.
```

```
256 def deleteContact():
257     print('\nDelete Contact\n')
258
259     # Check to see if the file or dictionary exists.
260     contactDict = checkExist()
261
262     # If the dictionary does not exist, return to main menu.
263     if (contactDict == None):
264         return
265
266     else:
267         try:
268             delete = input('Enter contact name or cancel: ')
269
270             # If user types cancel, return to main menu.
271             if (delete.lower() == 'cancel'):
272                 return
273
274             else:
275                 # Delete the contact from the dictionary.
276                 contactDict.pop(delete)
277
278                 print('The contact ', delete, ' has been deleted from the dictionary.')
279
280                 # Open the file contacts.dat in write mode.
281                 contacts = open('contacts.dat', 'wb')
282
283                 # Pickle the dictionary, and write it to the file.
284                 pickle.dump(contactDict, contacts)
```

```
285
286         # Close the contacts.dat file.
287         contacts.close()
288
289         # If there are no contacts in the dictionary, give response to user.
290         if (contactDict == None):
291             print('There are no contacts in the dictionary.')
292
293         else:
294             return
295
296     # Exception for invalid input.
297     except KeyError:
298         print('The contact does not exist in the dictionary.')
299
300     # Ask the user if they would like to check the list.
301     checkList = input('Would you like to check the list? ')
302
303     # If user types yes, return organized sorted list.
304     if ('yes' in checkList.lower()):
305         for name, email in sorted(contactDict.items()):
306             print('Name: ', name, '\nEmail: ', email, '\n')
307
308     else:
309         return
310
311
312 # Function that contains the main menu to the dictionary.
313 def main():
```

```
314     option = ""
315
316     # Loop for the main menu to run continuously.
317     while option.lower() != 'exit':
318         print('-----')
319         print('    Email Dictionary Main Menu')
320         print('-----')
321         option = input('Please select an option from the menu by number or name:\n'
322                        '1. Add New Contact\n'
323                        '2. Find Contact\n'
324                        '3. Edit Contact\n'
325                        '4. Delete Contact\n'
326                        '5. Exit\n\n'
327                        'Select Option: ')
328
329         # If statement for each of the options on the menu.
330         if (option.lower() in '1. add new contact'):
331             addContact()
332
333         elif (option.lower() in '2. find contact'):
334             findContact()
335
336         elif (option.lower() in '3. edit contact'):
337             editContact()
338
339         elif (option.lower() in '4. delete contact'):
340             deleteContact()
341
342         elif (option.lower() in '5. exit':
```

```
343         break
344
345     else:
346         print('Your input is invalid. Please enter a number or name from the menu.')
347
348
349
350 # Call main function to begin program.
351 main()
352
353
354 Output Result:
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: G:\ITMD 513 Open Source Programming Python\hw8\EmailDictionary.py =
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 6
Your input is invalid. Please enter a number or name from the menu.
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 2

Find Contact

Enter the contact name, all, or cancel: all
Name: John
Email: jsmith@gmail.com

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
```

Ln: 247 Col: 4

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 1

Add Contact

Enter new contact name or cancel: Dr. Sam
Enter new contact email address: shamsuddin@iit.edu

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 1

Add Contact

Enter new contact name or cancel: cancel

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Ln: 247 Col: 4
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Enter new contact name or cancel: cancel
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 1

Add Contact

Enter new contact name or cancel: Deborah Barndt
Enter new contact email address: dbarndt@hawk.iit.edu
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 2

Find Contact

Enter the contact name, all, or cancel: Dr. Sam
Name: Dr. Sam
Email: shamsuddin@iit.edu
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
```



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 2

Find Contact

Enter the contact name, all, or cancel: cancel

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 2

Find Contact

Enter the contact name, all, or cancel: all
Name: Deborah Barndt
Email: dbarndt@hawk.iit.edu

Name: Dr. Sam
Email: shamsuddin@iit.edu

Name: John
Email: jsmith@gmail.com

-----
Ln: 247 Col: 4
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 3

Edit Contact

Enter contact name or cancel: John
Enter new email or cancel: jsmith404@gmail.com
Contact email updated.

Name: John
Email: jsmith404@gmail.com
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 1

Add Contact

Enter new contact name or cancel: Gary Sterna
Enter new contact email address: gsterne@gmail.com
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 3

Edit Contact

Enter contact name or cancel: Joe
The contact is not in the dictionary. Would you like to create this new contact?
yes
Enter new email address: joetrader@gmail.com
Name: Joe
Email: joetrader@gmail.com

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 4

Delete Contact

Enter contact name or cancel: Joe
The contact Joe has been deleted from the dictionary.

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
```

Ln: 247 Col: 4

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
The contact Joe has been deleted from the dictionary.
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 2

Find Contact

Enter the contact name, all, or cancel: all
Name: Deborah Barndt
Email: dbarndt@hawk.iit.edu

Name: Dr. Sam
Email: shamsuddin@iit.edu

Name: Gary Sterna
Email: gsterne@gmail.com

Name: John
Email: jsmith404@gmail.com

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 5
>>> |
```

Ln: 247 Col: 4

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
>>>
= RESTART: G:\ITMD 513 Open Source Programming Python\hw8\EmailDictionary.py =
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 3

Edit Contact

Enter contact name or cancel: Joe
The contact is not in the dictionary. Would you like to create this new contact?
no
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 3

Edit Contact

Enter contact name or cancel: cancel
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
```

Ln: 358 Col: 4

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: find contact

Find Contact

Enter the contact name, all, or cancel: all
Name: Deborah Barndt
Email: dbarndt@hawk.iit.edu

Name: Dr. Sam
Email: shamsuddin@iit.edu

Name: Gary Sterna
Email: gsterne@gmail.com

Name: John
Email: jsmith404@gmail.com

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: delete contact

Delete Contact

Ln: 358 Col: 4
```

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Delete Contact

Enter contact name or cancel: John
The contact John has been deleted from the dictionary.
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: find contact

Find Contact

Enter the contact name, all, or cancel: all
Name: Deborah Barndt
Email: dbarndt@hawk.iit.edu

Name: Dr. Sam
Email: shamsuddin@iit.edu

Name: Gary Sterna
Email: gsterne@gmail.com

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: exit
>>> |
```

Ln: 358 Col: 4

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
>>>
= RESTART: G:\ITMD 513 Open Source Programming Python\hw8\EmailDictionary.py =
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: find contact

Find Contact

Enter the contact name, all, or cancel: John
The contact does not exist in the dictionary.
Would you like to check the list? yes
Name: Deborah Barndt
Email: dbarndt@hawk.iit.edu

Name: Dr. Sam
Email: shamsuddin@iit.edu

Name: Gary Sterna
Email: gsterne@gmail.com

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: update contact
Your input is invalid. Please enter a number or name from the menu.
```

Ln: 409 Col: 4


```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Select Option: find contact

Find Contact

Enter the contact name, all, or cancel: John
The contact does not exist in the dictionary.
Would you like to check the list? yes
Name: Deborah Barndt
Email: dbarndt@hawk.iit.edu

Name: Dr. Sam
Email: shamsuddin@iit.edu

Name: Gary Sterna
Email: gsterne@gmail.com

-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: update contact
Your input is invalid. Please enter a number or name from the menu.
-----
Email Dictionary Main Menu
-----
Please select an option from the menu by number or name:
1. Add New Contact
2. Find Contact
3. Edit Contact
4. Delete Contact
5. Exit

Select Option: 5
>>> |
```

Ln: 409 Col: 4