

Input/Output, Error Handling and Parsing Textual Formats

1) Write a command line app which takes input from the command window. This application is related to a phone contact list. The first input is always the name of a ".txt" file used to store the phone records.

a) Implement the following features according to the following inputs given to the command line

- i) If the following input is a name (a name can contain more than one word), it should find the corresponding mobile number(s). There can be more than one person with the same name as shown below. In case of multiple persons with given input name, the name of the person and their number should be printed. Also print a no contact found message when appropriate.

Example:

Input:

C:\Downloads\contact_list.txt Mary Anne

Output*:

Mary Anne 808-779-1466

Mary Anne David 843-798-6698

Mary Anne Doe 801-557-2819

*Notice the output should be formatted such that numbers appear in the same column. Adapt the name number spacing according to the output.

- ii) If the following input is a number the corresponding name should be printed out. If multiple names are found, an exception (Stored Data Duplicate Number Exception) should be thrown but the first found name should be printed. Also print a no contact found message when appropriate. (Assume

the correct input for a number contains only 10 digits and no symbols, although the output and stored values does.)

- iii) If the following inputs is a name followed by a number it should be added to the contacts list in the alphabetical order of names. If the contacts list does not exist, the text file and necessary directories should be created or an unable to create message should be given. Note that irrespective of the input, the first letter of each name should be capitalized and the rest should be in lower case. Duplicate numbers should not be added.

b) The following Exception classes should be handled in your application in addition to the file read/write Exceptions. Always catch the most relevant Exception first and give an appropriate message to it. The Exceptions should be Runtime Exceptions where appropriate.

When in doubt use the first character, number of inputs and length of input to decide if the input is a name or number.

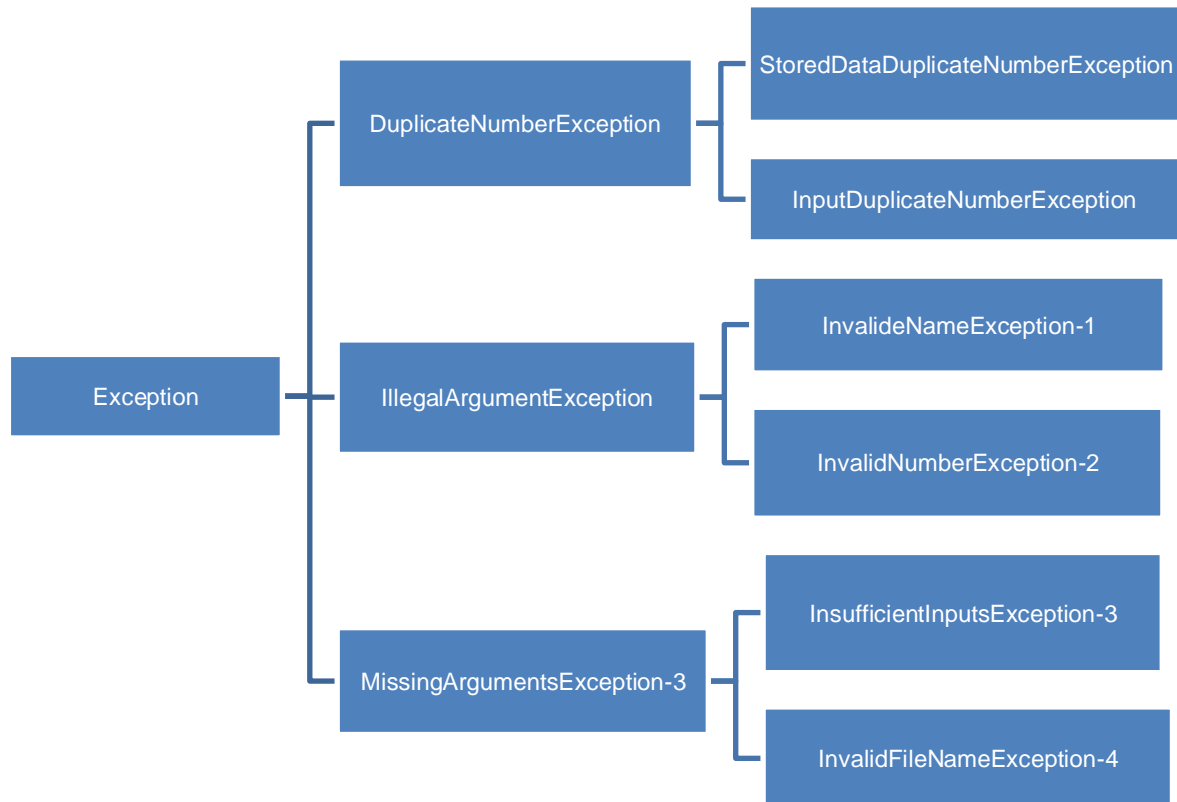


Figure 1: Exception Hierarchy

1. Inclusion of symbols and numbers
2. Inclusion of symbols and letters or number of digits not 10
3. The app receives less than 2 arguments
4. The first input is not a proper name of a text file

- c) Irrespective of an Exception, the program should always output a greeting message as below.
"Thank you for using contact list app" (you may change the message)
- d) Implement a functionality for the command line application so that it will add contact details from a json file to the contact_list file. Your second and third input passed via command line should be add_json and <json_file_name>.

Example: C:\Downloads\contact_list.txt add_json example.json

Note: The text in your contact list ".txt" file should be formatted as shown below

Mary Anne: 808-779-1466

Mary Anne David: 843-798-6698

Mary Anne Doe: 801-557-2819

A list of contacts and an example Jason file are attached.