CSE 174 - Assignment 4
Deadline: Monday, Dec 6th at 11:59 PM

You are to write a program that reads data from a file, and creates and works with an ArrayList of Customer objects.

- Create a new class/file called **Assignment4.**
- Download the API Documentation for Customer Class.
- Download the Customer.java (DO NOT CHANGE ANYTHING INSIDE THIS FILE), list.txt, and customers.txt files and put them in the same folder as your Assignment4.java file.

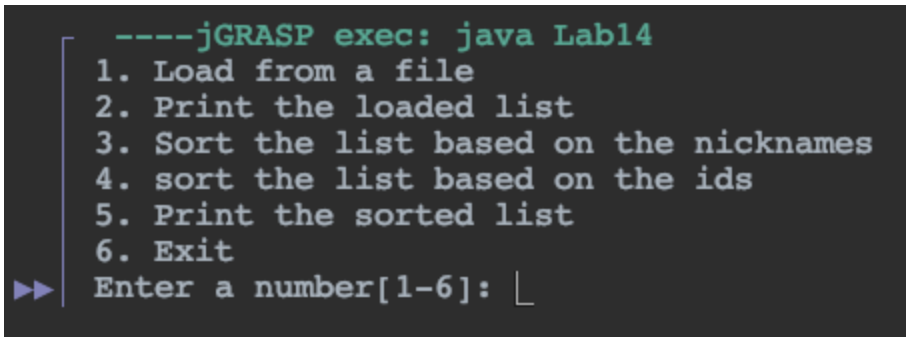
For this assignment you are allowed to use everything you have learned so far to generate the expected output. There are only 2 limitations that you must follow:

1. The length of methods (including the main method) should be **less than 35 lines.**
2. Your class must have 2 public sort methods called **sort1** and **sort2** which are explained later.

Other than that, you can use anything that you have learned in this course.
The quality of your code will be evaluated by the assessor and points will be taken off if you don't follow the proper programming practices. So, avoid using things that we have never used in the course.

Part1: Run the program

When your program is run, the following menu needs to be displayed:



At this point nothing has been read from the file, so if the user chooses menu options from 2-5, nothing should happen and the user should be prompted with a proper message as the following:

```
 ----jGRASP exec: java Lab14
1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 2
No data has been loaded yet!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 3
No data has been loaded yet!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 4
No data has been loaded yet!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
```

```
Enter a number[1-6]: 5
No data has been loaded yet!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 6
End!

 ----jGRASP: operation complete.
```

Part2: Reading from a file & creating an ArrayList of Customer objects

When the user enters 1, it asks the user to enter a filename and starts reading data from the given file and creates an ArrayList of Customers. The following is the result:

```
----jGRASP exec: java Lab14
1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 1

Enter the name of the file: list.txt
Loading from the file is done!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]:
```

When option 1 is done:

- The user can enter 2 to display the list.
- Choosing number 2 only displays 10 elements at a time. If the user enters character 's', it will stop showing the rest of the list. If the user enters anything else, it will show the next 10 elements of the list and so on.
- At this level if the user enters number 5, nothing should be printed because the list hasn't been sorted yet.

  The results are as follows:

```
1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 2

**** Printing the list ****
1. [Eric, 7383837]
2. [Jordan, 378954]
3. [Mike, 473674]
4. [Alen, 888888]
5. [Meisam, 42736482]
6. [Alex, 43654365]
7. [Ethan, 237423]
8. [Jim, 6689745]
9. [Jordan, 378954]
10. [Fatima, 654782]
Enter something to continue/enter s to stop: s

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 5
Nothing is sorted yet!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]:
```

Part3: Sorting

Your class must have 2 sort methods called **sort1** and **sort2**. These two methods both **accept an ArrayList of Customer objects** and **return nothing**. From the sorting algorithms that you have learned so far (bubble sort, selection sort, and insertion sort) you can pick any one you like for sort1 and sort2 methods. For instance, you can use bubble sort for sort1 and insertion sort for sort2, or you can use the same sort algorithm for both sort1 and sort2 methods.

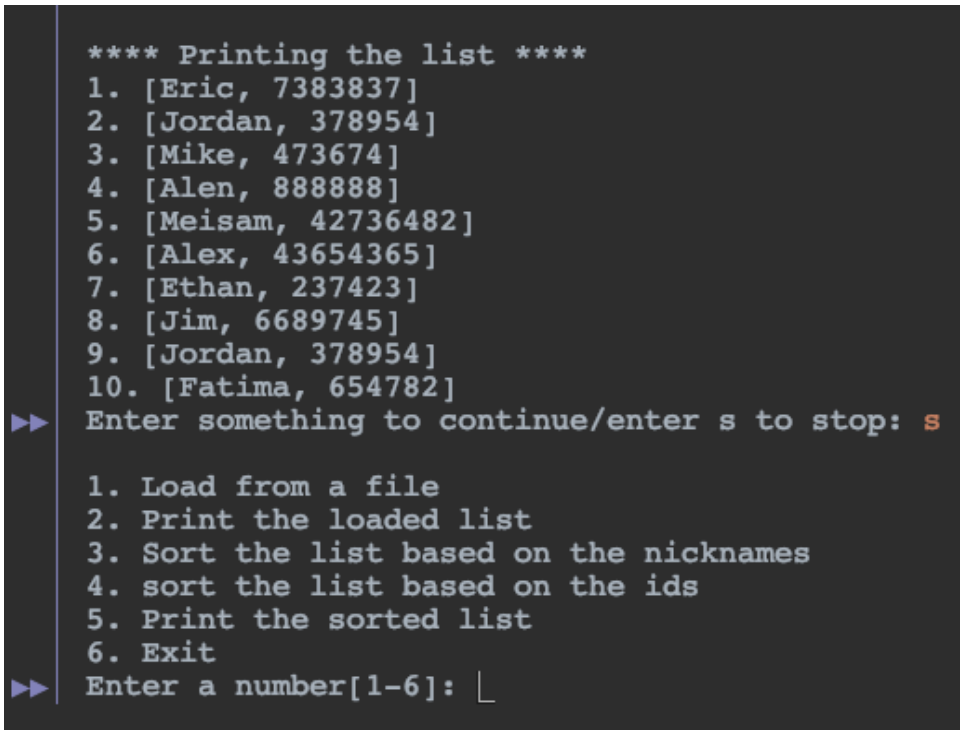
The **sort1** method should sort the list based on **nicknames**, and the **sort2** method should sort the list based on the **id numbers**.

- When the user enters 3, the list should be sorted by nicknames.
- When the user enters 4, the list should be sorted by id numbers.
- The **last** sorted list can be displayed by entering 5.
- If the user enters 2, the **original, unsorted** list still needs to be displayed as before.

The results are as follows:

```
1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 3
Sorting is done!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 5

**** Printing the list ****
1. [Alen, 888888]
2. [Alex, 43654365]
3. [Eric, 7383837]
4. [Ethan, 237423]
5. [Fatima, 654782]
6. [Jeniffer, 987451]
7. [Jim, 6689745]
8. [Jordan, 378954]
9. [Jordan, 378954]
10. [Kason, 32984723]
Enter something to continue/enter s to stop: s

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 4
Sorting is done!
```

```
1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 5

**** Printing the list ****
1. [Ethan, 237423]
2. [Jordan, 378954]
3. [Jordan, 378954]
4. [Mike, 473674]
5. [Fatima, 654782]
6. [Alen, 888888]
7. [Jeniffer, 987451]
8. [Steven, 1231231]
9. [victor, 1283719]
10. [Patty, 2736282]
Enter something to continue/enter s to stop: s

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 2
```



Part4: Loading another file

If the user enters 1 again to load another file, everything will be repeated from part2. Please go back and read part2 of this assignment again.

Part5: Test your code with more sample runs

Just follow the following sample runs to test your code and make sure that your code is generating the same results. .

```
----jGRASP exec: java Lab14
1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 1

Enter the name of the file: customers.txt
Loading from the file is done!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 5
Nothing is sorted yet!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 2
```

```
**** Printing the list ****
1. [hygmfaizz, 9916134364]
2. [ehluopilk, 9916152142]
3. [khkv, 9916165273]
4. [atxkmtob, 9916167866]
5. [izimhg, 9916169430]
6. [jjdmsg, 9916178693]
7. [elfjb, 9916180232]
8. [rkinbcvx, 9916194910]
9. [gtjd, 9916224758]
10. [tjpsgjqgw, 9916234418]
Enter something to continue/enter s to stop: c
11. [dxidry, 9916266182]
12. [bhnwnszimr, 9916271265]
13. [gbpcsds, 9916275669]
14. [nudwzo, 9916280038]
15. [fnknfyrnn, 9916300170]
16. [ucniyrv, 9916302067]
17. [mwapy, 9916316819]
18. [jlsb, 9916320501]
19. [gmwfsbql, 9916332767]
20. [sltdoid, 9916332876]
Enter something to continue/enter s to stop: c
21. [cpslj, 9916336504]
22. [jfohdky, 9916345657]
23. [ldtqwoftbe, 9916348748]
24. [japqt, 9916365819]
25. [oqtat, 9916371546]
26. [hyeocapyr, 9916372460]
27. [txkittwy, 9916373545]
28. [ynffhr, 9916380017]
29. [fazdesm, 9916382401]
30. [rvs, 9916383718]
Enter something to continue/enter s to stop: s
```

```
1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 3
Sorting is done!
```

```
1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
▶▶ Enter a number[1-6]: 5

**** Printing the list ****
1. [aaegsaym, 9925687075]
2. [aakolhnbe, 9917279454]
3. [aamtrkzxc, 9917655744]
4. [aazzq, 9924061450]
5. [abeyd, 9922046924]
6. [abnby, 9920828997]
7. [adkyczo, 9916813569]
8. [adln, 9916690590]
9. [adviinx, 9931024049]
10. [advwpvvb, 9925575685]
▶▶ Enter something to continue/enter s to stop: c
11. [ael, 9922483852]
12. [aetaat, 9918362060]
13. [affitgtvr, 9920984855]
14. [afjegflpc, 9930259838]
15. [aflxqwgdnt, 9928078034]
16. [afur, 9917058325]
17. [aglqyeyy, 9921813239]
18. [ahawyx, 9923864361]
19. [ahvpb, 9924952982]
20. [aif, 9928370692]
▶▶ Enter something to continue/enter s to stop: s

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
▶▶ Enter a number[1-6]: 4
```

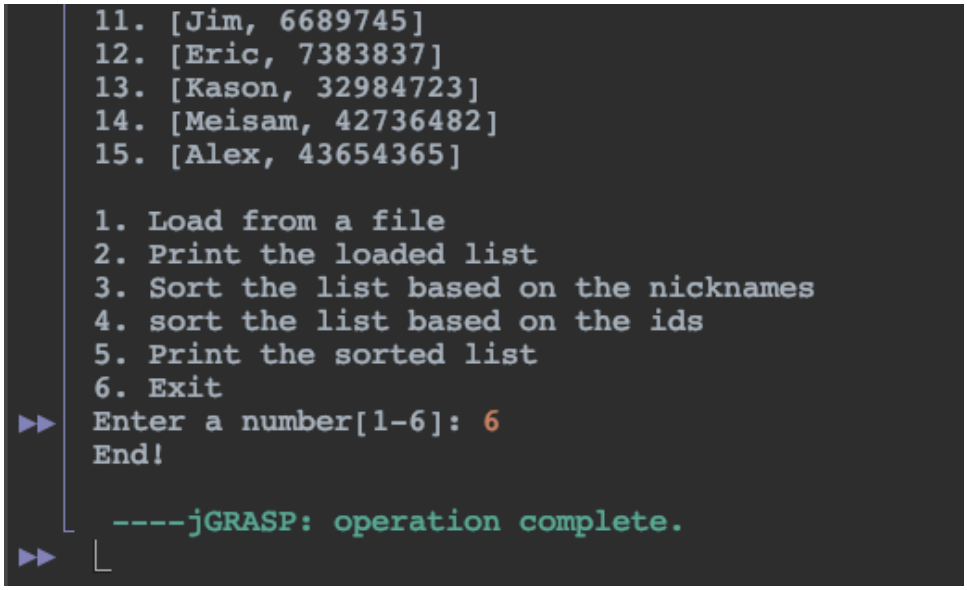
```
Sorting is done!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
```

```
Enter a number[1-6]: 5

**** Printing the list ****
1. [hygmfaizz, 9916134364]
2. [ehluopilk, 9916152142]
3. [khkv, 9916165273]
4. [atxkmtob, 9916167866]
5. [izimhg, 9916169430]
6. [jjdmsg, 9916178693]
7. [elfjb, 9916180232]
8. [rkinbcvx, 9916194910]
9. [gtjd, 9916224758]
10. [tjpsgjqgw, 9916234418]
Enter something to continue/enter s to stop: c
11. [dxidry, 9916266182]
12. [bhnwnszimr, 9916271265]
13. [gbpcsds, 9916275669]
14. [nudwzo, 9916280038]
15. [fnknfyrnn, 9916300170]
16. [ucniyrv, 9916302067]
17. [mwapy, 9916316819]
18. [jlsb, 9916320501]
19. [gmwfsbql, 9916332767]
20. [sltdoid, 9916332876]
Enter something to continue/enter s to stop: s

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 2

**** Printing the list ****
1. [hygmfaizz, 9916134364]
2. [ehluopilk, 9916152142]
3. [khkv, 9916165273]
4. [atxkmtob, 9916167866]
5. [izimhg, 9916169430]
```

```
6. [jjdmsg, 9916178693]
7. [elfjb, 9916180232]
8. [rkinbcvx, 9916194910]
9. [gtjd, 9916224758]
10. [tjpsgjqgw, 9916234418]
Enter something to continue/enter s to stop: s
```

```
1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 1

Enter the name of the file: list.txt
Loading from the file is done!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 5
Nothing is sorted yet!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 3
Sorting is done!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 5

**** Printing the list ****
1. [Alen, 888888]
```

```
2. [Alex, 43654365]
3. [Eric, 7383837]
4. [Ethan, 237423]
5. [Fatima, 654782]
6. [Jeniffer, 987451]
7. [Jim, 6689745]
8. [Jordan, 378954]
9. [Jordan, 378954]
10. [Kason, 32984723]
Enter something to continue/enter s to stop: s

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 4
Sorting is done!

1. Load from a file
2. Print the loaded list
3. Sort the list based on the nicknames
4. sort the list based on the ids
5. Print the sorted list
6. Exit
Enter a number[1-6]: 5

**** Printing the list ****
1. [Ethan, 237423]
2. [Jordan, 378954]
3. [Jordan, 378954]
4. [Mike, 473674]
5. [Fatima, 654782]
6. [Alen, 888888]
7. [Jeniffer, 987451]
8. [Steven, 1231231]
9. [victor, 1283719]
10. [Patty, 2736282]
Enter something to continue/enter s to stop: x
```



Part6: Submit on Canvas

If your program worked as shown above, now it's time to submit your work on Canvas. The Code Plugin will test the result of your code, along with sort1 and sort2 methods.

Rubric

Criteria	Full Credit	Partial Credit	No Credit
Successful submission via Code	A fully successful submission to CODE that passes all tests will earn full credit. **60 Points**	Complete each test. **12 Points**	If your submission is not accepted by code, you will receive no credit. **0 Points**
Correct Style	Zero style errors, will earn full credit. **10 Points**		Any style errors present, you will receive no credit. **0 Points**
Proper programming practices	If your code follows all acceptable practices, you will earn full credit **30 Points**	Unacceptable code gets partial credit.	Significant issues related to programming practices will earn no credit. **0 Points**