Using the **HouseException.java** and the**House.java** from A01-House you will implement a driver class called **RealEstate.java** to manipulate an Array of House objects. This time you will be creating the driver class.

**For this assignment you will learn the following:**

* Using the Arrays class methods to manipulate arrays
* using enhanced for loops (if so desired)

**You will practice**

* Creating driver classes
* Reading information from the user
* Looping and conditionals
* Using exceptions try/catch
* Using the Java coding standards

**When your code begins execution the following menu should be displayed**

1. add a house  
2. remove a house  
3. print houses that cost less than a given price  
4. print all the houses  
0. end this program

**Your driver class should:**

* be named should be RealEstate.java
* work with your House.java and HouseException.java (fix them if you did not get full credit otherwise you will get points deducted \*again\*)
* properly validate user input
* not crash (always use try/catch).

**Implementing the looping menu in the driver class.** Looping menu total 55 points

* When requesting user input, you must print a user-friendly message requesting it. Your code must be user-friendly. (10 points)
* Your code should continue to loop until the user selects 0. If the program terminates at any other point for any other reason you will lose points.  *(15 points)*.
* If the user enters an out of range menu option number, you should print a carefully crafted error message explaining the problem and print the menu again. *(15 points)*.
* If the user enters anything other than a number, you should catch the exception, print a carefully crafted error message explaining the problem and print the menu again. *(15 points)*.

**Implementing the menu options**

**If the user selects 1, (add a house)***( total 60 points)*.

* Your code should read the house information from the user using the Scanner class.
* You should validate everything *using* the House and HouseException and in case of invalid input print the appropriate message and go back to print the menu again.
* If the user information is valid then you should make sure that there is no other house with the same MLS. In other words, you should make sure there are no duplicate houses/MLSs. If the MLS already exists you should print a message saying so and go back to the main menu, and continue to loop.
  + For the purpose of this assignments a duplicate house is a house with the same mls as another regardless of the other instance variables.

**If the user selects 2, (remove a house)***(60 points)*.

* You should read the MLS number
* if the input is not valid, print an error message and loop back into the menu
* You should search the array looking for a house matching the mls entered
* If a match is found you should remove the house. You must print an appropriate message and loop back into the menu
  + when removing a house from the array you should be careful about leaving holes in the middle. They may result in an exception if they are not carefully handled.
* it there is no match you should print an appropriate message for the user and loop back into the menu.

**If the user selects 3 (print houses that cost less than a given price)***(40 points)*.

* read the price from the user
* You should validate it and in case of invalid input print the appropriate message and go back to print the menu again, or the main menu.
* If the price is valid then you should should search the array and print all the houses with a price smaller than the price entered by the user and print them to the screen

**If the user selects 4 (print all the houses)***(20 points)*.

* you may use the enhanced for loop.
* Make sure that the output is properly formatted; add a title to your printout!

**For both print methods (options 3 and 4):** You should never print any null positions. If you do this you will get points deducted.

**If the user selects 0***(5 points)*.

* you will print a "thank you" message and end program execution

**Other considerations**

* The array should not contain duplicate mls houses at any time.
* Feel free to use any of the Arrays class methods we learned this week in class.
* You should use the proper exceptions, like InputMismatchException or HouseException, etc as appropriate
* You should always use try/catch and your code should not crash

**You MAY get extra credit if:**

* If you use several static methods within your driver, for example one method per menu option and maybe one method for the menu itself.
* Great documentation / comments
* Overall program usability (user friendliness)
* *Extra credit is only earned if overall your code is at least 90% correct.*

When you submit your assignment, you should submit the driver class as well as your House.java and HouseException.java (again) *(40 points for the resubmission if correct)*. If previously, your code allowed for the creation of invalid objects you must review and fix your code or you will get deductions in this assignment as well. ***Make sure you read your feedback!***

[**Following the Java coding standards**](http://www2.hawaii.edu/~tp_200/ics111/material/codingStandards.html)and put your name at the top of your code. *(20 points)*

***If your code is incomplete or it does not compile you will get at least 60% deduction.***

Questions? Send an email ASAP

Need help? Send an email ASAP.