Your work on file manipulation for the upcoming automated advisor assign functionality was received with high praise from University administration. It has been decided that you are not to be assigned to the next phase of the project…building JAVA objects from the last phase validated file input.

Specifications:

- The validated file used from the last phase does need a couple of modifications before it is used in this project. Your partner is taking care of this and all you need is a test file that is being supplied to you. The two items being modified are:
    - The full name field is being removed. It has been decided that the process may be sped up by building it into the object instantiation process by adding a get method named **`getFullName()`** that contains the code to build the full name upon demand within the method. Not all schools/departments are using the full name field and that is another reason it is being removed.
    - Schools have requested that the field containing the number of assignees (the last field in the file) be optional and create a default setting/number if it is left out. The default number is contained in the specifications found later in this document.
    - Let's look at the supplied test file in.dat.
- You must create two public JAVA classes.
    - The first class, Advisors, is to be used to instantiate Advisors objects that are to be used by other processes.
    - The second class, CreateAdvisors, is used to instantiate an array of Advisors objects that are to be used by other processes that remain to be programmed
        - This class also contains a test display method written by your partner (the instructor) that displays the objects data in order to determine that all logic/code is performing correctly. After enough testing, this display method is to going to be commented out, but left it the code in case it is necessary.

<u>Advisors class:</u>

- Should be coded first…nothing works without it
    - Tip: It can be coded partially and added to…more later
- Should contain no code that displays anything to the screen or elsewhere
- Should contain no exception processing…we are doing it slightly different than the text
- Contains instance variables that will contain each of the fields in the input file and no more.
    - Let's talk…how many does that mean?
- Instance variables must be the appropriate data type.
    - Again…let's talk.
- Should contain a single class variable to count advisor objects as they are created
- Should contain 3 constructors
- Each constructor ONLY initializes the instance variables
- Exception…may be used to add to the count of advisors if you wish
- Constructors MUST initialize the instance variable by calling a properly coded set method…1 set method sets 1 variable. No set method can set more than 1 variable.
- First constructor:
    - Zero parameter constructor
    - Calls set methods to initialize all instance variables to an error value
        - String error value: **Unknown** (including asterisks)
        - Numeric error value: 999
    - Yes…it still adds to the total advisor count
- Second constructor:
    - Contains all the parameters needed to initialize only the String instance variables
    - This constructor is called if the number of advisees is left out of the row of the file
    - Calls the proper set method to set the number of advisees to the default value of 35
    - Calls the appropriate set methods to initialize all other instance variables to the values passed to this constructor
- Third constructor:
    - Contains the same functionality as the second constructor except the integer parameter (number of advisees) is added and the value passed to it is used to initialize the number of advisees instead of the default 35
- Set methods:
    - A single set method is coded that are invoked ONLY by constructors to set the instance variables AFTER the set method has verified the value passed to it
    - First Name validation:
        - Cannot be null
        - Cannot be an empty String
        - Must be 3 characters in length or longer
        - If invalid, set the variable to ****UNKNOWN****
            - Include the asterisks
    - Last Name validation
        - Cannot be null
        - Cannot be an empty String

        - Must be 5 characters in length or longer
        - If invalid, set the variable to ****UNKNOWN****
            - Include the asterisks
    - Advisor ID
        - Cannot be null
        - Cannot be an empty String
        - Must be exactly 10 characters in length
        - Must contain ONLY digits…no other characters are allowed
        - If invalid, set the variable to ****UNKNOWN****
            - Include the asterisks
    - Major
        - Cannot be null
        - Cannot be an empty String
        - Must be between 4 and 7 characters inclusive
        - If invalid, set the variable to ****UNKNOWN****
            - Include the asterisks
    - Assignees (the lone integer)
        - If contained in row in the file (row has five fields)
            - Must be between 20 and 200 inclusive
            - If the value retrieved from the file is less than 20 or greater than 200, set the instance variable to 999
        - If an assignees value is NOT in the file (row has only 4 fields)
            - Set the instance variable to the default of 35 as indicated earlier
    - How many do you need?
        - Points are removed for an incorrect number of set methods
    - All instance variables must be private
    - All set and get methods must be public
    - Constants must be public
- Get methods
    - A get method must be written to return:
        - The value contained in each instance variable
        - The full name
            - Built inside the get method in the format lastName, firstName
        - The total count of advisor objects
        - One value returned = a single get method
            - No toString-like methods
    - Suggestion…look at the display method provided to you for get method names and use those
        - Otherwise the display method will not function
    - How many do you need?
        - Points are removed for an incorrect number of get methods
- All get methods must begin with get
- All set methods must begin with set
- Constants (using the keyword final) must be created and used whenever appropriate

  - Let's talk about it
- I would expect between 120 and 150 lines of code not including comments etc.
  - But a lot of it can be "copy-paste-change" if you wish to do so
  - Probably 3 to 4 hours coding time…maybe more
- Questions about the Advisors class?

- **CreateAdvisors class:**
  - Is the executable class
    - Contains the main method to drive the application
  - All code may be placed inside of the main method if you wish, but an extra credit option is available for writing it using method…see the last page of this document
  - This class involves using file I/O
    - All file I/O is capable of throwing exceptions
    - Exceptions must be handled EXACTLY as they were in Project 02. See Project 02 specs
  - Must prompt the user for the absolute path and file name
    - Use whatever path you wish…no exact drives etc. are used for this project
  - Must read each row of the file and use the data contained in each row to instantiate an Advisors object
  - Each object is added to an Advisors array of objects until the end of the file is reached
  - When the end of the file is reached, close the input file immediately
  - A file will contain a maximum of 100 rows
    - Note…you may load the entire file into a String array and build the object array from the String array OR forget about the String array and build the object array as you read each line. You choose…either works fine
  - Remember that the file given you has already been validated from the previous process for the correct number of fields AND the integer values can be parsed into integers
    - Do not validate these again!
  - Each Advisors object must be instantiated by calling the proper constructor of the Advisors class
    - If it contains 4 fields, you call the four-parameter constructor
    - If it contains 5 fields, you call the five-parameter constructor
    - The default constructor is only there for safety reasons and it should never be intentionally called
      - Hint: I will call it during grading to ensure it behaves as expected
  - After the object array is created completely, you should call the supplied display method **`testDisplayAdvisorsObjects()`** and it will display a test screen for you
    - Just copy and paste it into your class and call it from main passing to it the object array you have constructed
    - Note: For the display method to work, you must add the following two import statements to your class
      - import javax.swing.*;
      - import java.awt.Font;

        - The display method may NOT be modified in any way…what you are getting is what it must be tested with
        - See in-class demonstration
    - I would expect 100 to 150 lines of code
        - This includes the display method (26 lines) already coded for you
        - Depends a lot on whether you use a String array or not

    - Tips:
        - Do not forget that you must still parse the integer field (when present) into an integer value even though you do not need to data verify it
        - Code in "chunks" and validate as you go
            - You could code only 1 value in each class (for example Last Name) and see that it is behaving as desired before going on
                - Let's talk about it
        - Start early…this one will probably take more time than any of your projects so far
            - If you have questions, ask!
            - Most questions can be answered via Canvas
        - Copy/Paste/Change can be used a lot in this project
            - You could use code from Project 02 in the executable class
            - All set and get methods have similar structure…do they not?
            - What about constructors?
            - Just don't copy/past from web sites or someone else's code😊

General Rules and Specifications:

1. All current syllabus and shop rules policies apply to both classes you are coding and are strictly enforced
2. Any modification of the supplied display method is not allowed
3. Use properly coded constants whenever possible
    a. Instructors decision if all are in place
    b. Questions about constants?
4. Due Date: November 25 by 11:59 PM
    a. No late or incomplete projects accepted per syllabus and shop rules
5. Project value: 100 points

Extra Credit

- You may earn up to 10 extra credit points by coding the executable class using methods instead of placing all code within main AND passing all values/references
    - Each method must perform a single task…not two or more
    - No variables etc. are allowed outside of a method
    - main must ONLY call other methods to do the necessary work
- Caution! The extra credit option probably WILL take more time. Don't lose 100 points in an attempt to gain 10! Poor economy😊