CS3310/Kaminski

Name (print)____

Asgn # ______1_________ Due ___Thurs Sept 18_______

Dr. Kaminski - you have my word that:

1. I wrote this entire application myself in accordance with the guidelines in the course policies & syllabus and university policies of what's considered acceptable student academic conduct. Any code (longer than a couple lines) used in this assignment which was created by someone else (even if changed slightly) or was written jointly with someone else is both clearly attributed (in a comment in the code) and described on the BOTTOM OF THIS COVER PAGE.
2. The program code attached here **did** actually produce the data file which is attached here.
3. There was no editing of the data file after the attached program produced it (except perhaps the font, font size and/or page-orientation for printing).

Signature ____

Final score for the asgn is (0.8 * outputPoints) + (0.2 * programmingPoints) which comes out to: ___________

(you can calculate this)

---------------------------------- for GRADER (below) -------------------------------------

________ points for output on a 0-100 scale, including

- Log.txt is correct, all present and follows all specs in terms of formatting
- Log file contains just what's requested in the DemoSpecs and NOT what's been written to the file during testing (because TestDriver deletes the file, that should be taken care of)
- follows DemoSpecs for what TestDriver does, in the correct order, using correct fileNameSuffix's for A1RawData and A1TransData files
- PrettyPrintUtility called (repeatedly) at appropriate times (see DemoSpecs's pseudocode for TestDriver)
- transaction itself is echoed before the response is provided (in the Log file)
- N is printed by PrettyPrintUtility (since it's the 1st thing in the Backup file)
- Status messages appear in the correct order in the correct places (when the event actually happens)
- SN and AN transaction requests show countries' DATA and NOT just the nameIndex data
- Data shown in the Log file for SN/SI/AN/AI is nicely formatted (as what's shown in the specs)

________ points for program code on a 1-100 scale including

- follows requirements specs & demo specs
- 8 (or more) physical files (4 program files, 4 (or more) object-classes)
- File and class names for the above follow what's in the specs
- uses OOP and modularizes appropriately
- uses descriptive naming (following what's in the specs, approximately)
- status messages are GENERATED at appropriate times, just next to where the actual reported event happens
  - file OPENED messages in the constructor which opens the file
  - file CLOSED messages in the FinishUp method which closes the file
  - Setup & UserApp starting & stopping messages at the top & bottom of those chunks of code
- **NO LINEAR SEARCH or SORTING of the dataTable – else -50 points**
  - **Uses DirectAddress in Insert and SelectById**
- ALL data file handling (RawData, UI) is done inside its CLASS, and NOT in Setup or UserApp code (e.g., opening file, closing file, reading file, detecting EOF)
- UI class handles ALL TransData file handling AND, it does Log file handling for UserApp (although Setup might just deal with Log file directly (opening, writing to, closing))
- appropriate commenting – see demo specs

PRESENTATION - no specific points deducted this time, but in future asgn's . . . points will be deducted for

- not putting the hand-in packet in the correct order
- no comment-line-of-stars between methods
- poor indent/align (which SHOULD visually show the logic of the code)
- no wrap-around (you should use a smaller font and/or landscape to eliminate the wrap-around irritation)
- fixed-width font for printing of Log so it's easier to grade