CIS 22C Data Structures
Team Project

Background

As a team of two to five programmers, you will assist in the implementation of a data processing system. The application data are of your own design with the requirement that each record in the system must contain a unique key (it must be a string), a secondary key (allow duplicates), and at least three non-key fields. For instance, a Book class could have the following member variables:

isbn – unique key
title – secondary key
author
publisher
year
price

Each member of the team is to code one or more functional areas of the project. When complete, the individual components will be integrated into one program, which will be demonstrated in a project presentation that will be held the last week of the quarter. While the final result is one integrated program, **each team member's work is to be maintained as separate file(s)**. If this is not possible, clearly document each function including the name of the developer.

Basic Requirements

Begin by displaying some general information about the project, the names and tasks of the developers. Processing is to be menu driven with the following options:

- Add new data,
- Delete data,
- Search (has a sub-menu)
    - Find and display one element using the primary key
    - Find and display elements using the secondary key (all matches)
- List (has a sub-menu)
    - List unsorted data // for details see Team Project-Part2 (next week)
    - List data sorted by the primary key
    - List data sorted by the secondary key
    - Special print, as an indented list // for details see Team Project-Part2
    - Hidden print option // for details see Team Project-Part2
- Write data to a file.
- Statistics. // for details see Team Project-Part2
- Quit.

At the end of the program, the file is to be automatically written to a disk file. This is in addition to the menu write file option. The file names do not have to be the same as the input file name, but the file format must be the same as the input file format so that it can be read back into the program.

Your grade for the project will be a factor of the team score weighted by a peer evaluation.
In addition to writing and debugging code, each member of the team will:

A. Give suggestions for the application data: think about original/interesting/educational data that matches the program requirements (you may not use the Book example – come up with something else). During the first team meeting, every student will bring his proposal, then the team will assign a preference number to each proposal (1 – the first choice, etc.); the final decision will be made by the instructor (no two teams should process the same data).

B. Participate in designing a solution to the problem: Data Diagrams, Structure Charts, UML Diagrams

C. Participate in writing the weekly reports.

D. Provide relevant test cases for the final presentation.

E. Individually test his/her part of the program (as much as possible). This implies writing test drivers (code that will not be included in the final project, but it will be used to test parts of the project).

F. Write documentation for his/her part of the project.

G. Create a part of the final documentation (introduction, data structure design, structure charts, program documentation, etc.)

H. Participate in all of the team meetings. Come prepared to the meetings. If a team member must be absent from a meeting, he/she should inform the other team members in advance; also he/she could keep in touch by phone or email.

I. Turn in his/her part of the project on the scheduled date.

J. Work with the team coordinator and other members in the team to integrate his/her part of the code.

K. Prepare for and participate during his/her team presentation.

L. Actively attend all of the presentation sessions. (Ask questions/provide answers).

M. Write the peer evaluation (5Points)

Weekly Reports

Five weekly reports are required. The due dates will be announced in class or written in the syllabus. One team member is to be designated to prepare these reports using the "Weekly Report" forms (provided on Canvas). The final report will be uploaded on Canvas. There will be a discussion forum for each report.

Peer Evaluation

Each member of the team is to rate the contribution of the other team members using the Peer Evaluation Form. These evaluation forms are to be turned in individually after your project presentation. Any member who does not complete a peer evaluation will receive a five (5) points deduction from his/her score for the project. Note that you are not to rate yourself. These evaluations will be used to determine a tentative score for each individual on the project. The instructor, however, is responsible for marking the final determination of each person's grade on the project and will use the teams' score only as one input in the grading process.

Project Score

A team score will be calculated as shown below.

1. Completeness (40%). The project contains all of the required units and functionality along with the required documentation. Failure to use separate files will result in a five (5) points penalty. Each programmer should have one source file (and maybe a header file too).
2. Accuracy (30%). The system demonstration runs without errors. Also, the system loads and executes without errors in a customer (instructor) test.
3. Documentation (20%). Documentation is complete and readable.
4. Presentation (10%).

A tentative individual grade will be determined: by multiplying the project score by an average rating factor and using a score factor as shown below. After the individual's tentative score, up or down adjustments may be made based on individual performance as observed in the presentations or based on the individual's program code and documentation.

Average Peer Rating	Score Factor
16 – 20	100%
12 – 15	95%
10 – 12	90%
8–9	80%
5–7	70%
3–4	50%
<3	0

Personal Score = Project Score x Score Factor

Recommendations

1. Read the team project requirements entirely
2. Do not procrastinate
3. Design first: data structures diagrams, UML diagrams and structure charts
4. Do not change function prototypes without everybody's approval
5. Attend all meetings; take attendance, take notes.
6. Every student should know the design of the entire application and reuse code written by other students, instead of writing it again
7. Do not turn in code that has compiling errors
8. During the fourth week, stop refining your code and start finalizing the project documentation and prepare for presentation.
9. When students are dropping after the middle of the quarter, it is tough to redistribute the tasks among the other team members. Therefore, if you think you might have to drop, let everyone know in advance and assist them with the transition.