CIS 22C - Data Structures
Team Project

Background

As a team of three to six programmers, you will assist in the implementation of a data processing system. The application purpose and data will be of your own choice.

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 in the second to last week of the quarter. While the final result is one integrated program, **each team member's work is to be maintained as separate source file(s)**.

Your grade for the project will be a factor of the team score weighted by multiple peer evaluations.

Basic Requirements

The application data must contain a **unique key attribute** and at least **three** non-key attributes (fields). There should be at least 25 data records that are read in from a file.

The data structure for the project consists of:

- A hashed table of at least 25 data records read from a file. As the records are read, they are placed in dynamic memory and the memory location (address) is then inserted into a hashed table. Collisions will be resolved using open addressing, linked lists or buckets.
- A binary search tree is to provide sequential processing and also contains the locations of the data in dynamic memory. The key used in the BST does not have to be the unique key of the data, it can be a secondary key.

Processing is to be menu driven with the following options:

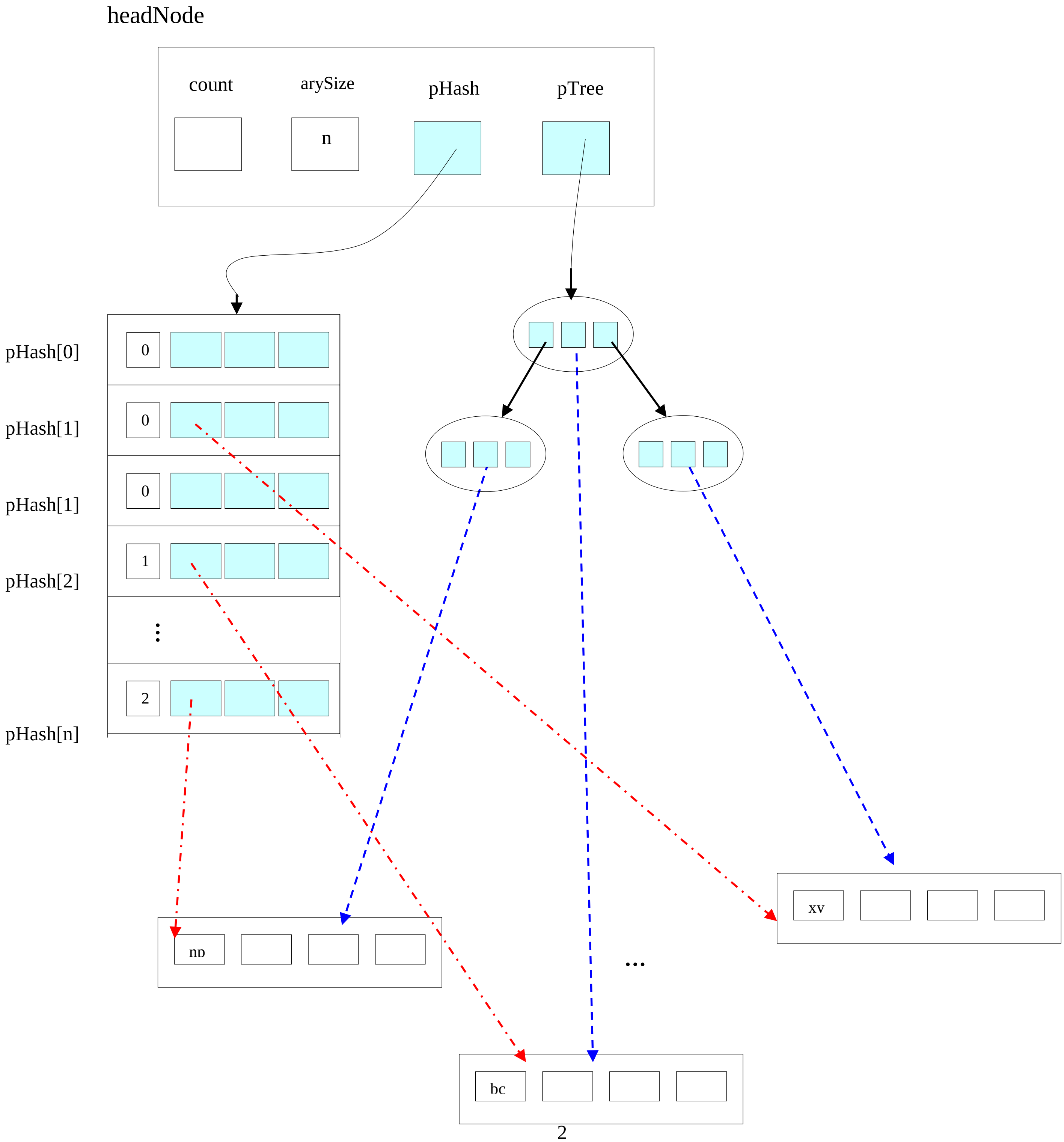
(1). Add new data
(2). Delete data
(3). Find and display one data record using the primary key
(4). List data in hash table sequence
(5). List data in key sequence (sorted)
(6). Print indented tree
(7). Efficiency
(8). <Team choice menu option>
(9). Quit

At the end of the program, the file is to be automatically written to the same input file. This is in addition to the menu write file option.

Data Structure Diagram

The following proposes a basic data structures for the project. You will need to change or add new structures to this diagram depending on the variation of the project you have. Project variations will be determined after the teams have been formed

Except for the head node, all structures need to be dynamically allocated. The hashed table size will depend on the number of data records and is not a constant. Each data record is dynamically allocated and only its address is stored in the BST and the hashed table.



Team Member Assignments

The following are the minimum units of the project – each team member must take on one complete unit. If team size is smaller, members may have to take on more than one unit.

Unit 1: Team Coordinator: Code team header file, main, and menu function. Coordinate integration testing and project presentation. Submit weekly reports.

Unit 2: BST: Code all BST functions, including the print indented tree function

Unit 3: Hashed Table: Code all hashed table functions, including collision resolution and print hashed table statistics

Unit 4: Standard IO: Code all standard, i.e. screen and keyboard IO functions, including basic user input checking

Unit 5: File IO: Code all file IO functions

Unit 6: Data Record**:** Code functions for the data record

Team of 3			**Team of 4**				**Team of 6**						
1	2	3	1	2	3	4	1	2	3	4	5	6	**Assignment**
X			X				X						**Unit 1:** Team Coordinator
	X			X				X					**Unit 2:** BST
		X			X				X				**Unit 3:** Hashed Table
X						X						X	**Unit 4:** Standard IO
		X				X					X		**Unit 5:** File IO
	X			X						X			**Unit 6:** Data Record
X	X	X	X	X	X	X	X	X	X	X	X	X	**Test Plan:** Options and data so that anyone could use it to demonstrate the project. **Presentation Outline:** Activity, duration, etc.
		X	X		X		X		X		X	X	**Project Documentation:** soft copy of all files

In addition to writing and debugging his / her own code, each member of the team will:

A. Participate in designing a solution to the problem, including flow chart, data diagram, UML diagram

B. Participate in writing the weekly reports

C. Provide relevant test cases for the final presentation

D. 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)

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

F. Create a part of the final documentation (introduction, data diagram, flow chart, etc.)

G. Come prepared to *all* the team meetings

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

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

J. Prepare for and participate during the team presentation

K. Actively attend all of the presentation sessions

L. Write the peer evaluations for own team members

M. Work with the team coordinators and other members to formulate questions to be asked of other teams and to evaluate their projects

Work submission

The team coordinator will submit two zip files to the Project Submission Link on Catalyst. These zip files will correspond to folders as explained below to collate project deliverables. In between submissions to the dropbox, you are welcome to use any source code control solution to manage code changes and synchronizations. Inside the Team_Name folder, create 2 separate folders as described below.

First folder, named **program_docs** (this will be your entire VS project folder tree):

1. Source Files and Header File(s). Make sure your name is at the top of each file. Name your file descriptively, for example bst.h is a lot more descriptive than george.h. Each class that is not a template class should have both a source file and a header file, if there are non inline functions.
2. Input Data File
3. Executable version of the project – remember to use Visual Studio as your executable should work on a Windows machine

Second folder, named **presentation**:

1. Demonstration Test Plan - should contain enough detail (options and data) so that anyone could use it to demonstrate the project. The test plan must demonstrate collision resolution.

2. Presentation file, could be PowerPoint, Word, PDF, any format as long as I can run it on the Window system. Include in the presentation:
    a. Team name and team member
    b. Introduction – a short summary describing the project application
    c. Data Diagram
    d. Flow Chart
    e. A short description of each team member's assignment
    f. Hash function (the actual code), and pseudocode and examples with typical data, showing collision resolution

Presentation

The presentation consists of two parts:

1. Project description (Sections 2.a – f listed above).
2. Complete demonstration of all options of the menu. The demonstration must include collision resolution and deletion of the root (it should have two children).

Weekly Reports

Four weekly reports are required. Each team turns in 1 report per week, and the due dates will be on Catalyst. The team coordinator will designate one responsible team member each week for this task and these reports will be submitted via Catalyst.

Intra-team Peer Evaluation

Each team member will rate the contribution of the other team members using the Peer Evaluation Form. These evaluation forms are to be turned in individually after all the project presentations. These will be submitted via the corresponding link on Catalyst. Any member who does not complete a peer evaluation will receive a 5 point deduction from his/her score for the project. Each team member’s score has to be a different score, accurately reflecting their contribution evaluation. Any member who gives the same score to everyone else on the team will receive a 5 point deduction from his/her score for the project. These evaluations will be used to determine a tentative score for each individual on the project.

Inter-team Peer Evaluation

All teams and their members are required to be present for the project presentations. Attendance will be taken at the start of the class and tardiness as well as group chatter will be penalized to allow for project presentations to begin and end on time. The team members will sit in their own groups and each team will be required to pose one or two queries to the presenting team in order of their team number, starting with the team that comes after the presenting team. Each team’s query has to be different from those of other teams, so the team members can huddle to come up with a common query. The audience teams will be given up to 2 minutes after the presentation has ended to fine-tune their query. The queries will be directed to the team member of the presenting team whose area of work is being questioned.

Each audience team will be required to evaluate the presentation using the presentation evaluation form, which is the same as be used by the instructor. Similarly, each presenting team will be required to evaluate the query from the audience teams on the same form. At the end of the evaluation form, a team score for that team will be given on a scale of 0-10. The team coordinator will be responsible for uploading all the evaluations to the project link on Catalyst after all the project presentations are complete. These evaluations will be part of the team grade for the project.

Project Score

A team score will be calculated as shown below.

1. Completeness (20pts). The project contains all of the required units and functionality along with the required documentation. The code is written efficiently.
2. Accuracy (15pts). The system demonstration runs without errors. And the system loads and executes without errors in a customer (instructor) test.
3. Documentation (10pts). Documentation of the code is complete and readable.
4. Presentation (5pts). The team is prepared with the presentation and the presentation file is complete.
5. Other team evaluations (10pts). This will be an average of all scores received in the inter-team evaluations.

The entire team will receive one team score. Then the individual score will be calculated by multiplying the team score with the average of the peer evaluation score. 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
18 – 20	100%
15 – 17	90%
12 – 14	80%
9–11	70%
6–8	60%
3–5	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 flow charts
4. Do not change function prototypes without everybody's approval
5. Attend all meetings and 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 start of the project, 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.