Data Structures

Assignment 5

(1) (30 pts) Implement a hash table with linear probing for collision resolution. Create a list of 20 words and use the standard mod hash function. Build it with an m of 40 and 80. Write this all as a class and include a find function that will return true or false if a word is in the data structure or not.

(2) (35 pts) Work with inheritance and polymorphism by setting up a base class with at least one virtual function and several derived classes and running a main() function that calls the virtual function to exhibit polymorphism.

That's really it. This can be open-ended. Decide what your classes represent and what you want to code up.

Some ideas include:

(1) Derive a variety of game-playing agents from a Player base class. The way these derived players make moves can be different—from human to computer to random players, etc. Have them play a tic-tac-toe variant, a simple card game, etc.

(2) A predator-pray simulation where you have an Organism base class and multiple different species as derived class and their update each time step is polymorphic. Track their locations on a 2D grid and give them simple update rules. An example is Ants and Bugs: an Ant moves one step randomly and will breed if it survives for three time steps in a row, whereas a Bug moves to an adjacent cell containing an Ant and eats it (moves randomly if no adjacent Ant) and will breed if it survives for eight time steps and will starve if it hasn't eaten an Ant for three time steps. Populate a 20 x 20 grid with 100 Ants and 5 Bugs and print output to the console in simple ASCII, with the user hitting enter to see what happens each time step.

(3) Continue work on our Bookstore simulation. Create a text file of info for, Books, Pens, and at least one other Item and have the Bookstore read from the files into the inventory. Have the Bookstore able to print_inventory into a file that includes all the to_string() info from all the Items in the inventory. You wouldn't have to add the Customer yet that comes to buy Items as that will be the next step we do in class.

(3) (20 pts) You have collected a file of movie ratings where each movie is rated from 1 (bad) to 5 (excellent). The first line of the file is a number that identifies how many ratings are in the file.
Each rating then consists of two lines: the name of the movie followed by the numeric rating from 1 to 5. Use a C++ map to store the data.

Here is a sample file with four unique movies and ten ratings:

```
10
Barbie
4
```

```
Barbie
5
Oppenheimer
2
Barbie
4
Oppenheimer
1
Mutant Mayhem
3
Equalizer 3
5
Oppenheimer
3
Barbie
2
Mutant Mayhem
5
```

Write a program that reads in a file in this format (or you may initialize a vector of strings with the data and hardcode it), stores the data in a map, and then calculates the average rating for each movie and outputs the average along with the number of reviews for each.

(4) (15 pts) Write a C++ program that performs a depth-first search on an arbitrary graph. Represent the graph in adjacency matrix form. You may use a two-dimensional array or a vector of vectors to represent this matrix. Your output should be the order of the nodes that are visited in the graph.