Programming Assignments #4 and 5
CS 202 Programming Systems

For This Program

With both programs #4 and #5 you will be implementing your solutions using Java. Your goal must be to develop an object-oriented solution but this time implement it in Java. Make sure that your OO Design is not centered around your data structures – your data structures support the design but shouldn't be the primary emphasis of your design. You may use Eclipse or IntelliJ to develop your software. Your Java programs must follow these rules:

- **No public or friendly fields (data members) *** NONE!!!*****
- No friendly methods (member functions)
- Yes, you SHOULD use the string class!
- Limit your use of static methods – these should be restricted to just utility functions and main
- Use an inheritance hierarchy using "extends"; there must be a minimum of 5 classes with 3 of them in a hierarchy. *These should not be isolated to just your data structures.*
- The application that USES the hierarchy must be in a class on its own. Or, it could be part of a hierarchy, so think about that.
- Create at least one abstract base class
- Implement at least one constructor with arguments
- Implement at least two functions using function overloading between classes and experiment with the way function overloading works in Java. ***Write about this.
- Implement dynamic binding and experiment with how it works in Java. Prove to yourself that the functions are being overridden versus overloaded. ***Write about this
- Try out the super keyword in invoking a base class' constructor. *This is what we use instead of an initialization list. ***Write about this in your*

For each of the above that you experiment with, write up information about it in your efficiency write-up

You are required to turn in a paper on how this solution is object oriented (your design). There is ONLY ONE design writeup for the combined programs 4-5. But, with EACH program, you are required to write 400 words about the efficiency and IDE (rather than the debugger).

Data Structures

In these last two programs, you must implement two data structures:

1. Program 4: A circular linked list of arrays, with base class references as the data performing upcasting
2. Program 5: Tree
    1. If you have already implemented a 2-3 tree in program #3, then it is time to implement a Binary Search tree (BST). With your implementation, make sure to fully support insert, display, retrieve, retrieve all related items, remove an individual item, and remove all; the algorithms must be implemented recursively.
    2. If you have not yet implemented a balanced tree in program #3, then implement a 2-3 tree, implementing insert, display, retrieve, retrieve all related items (no remove individual items).
    3. Each node needs to have a head pointer to a linear linked list for all matching items. This means there can be duplicate data – but will only appear in one node's data structure.

The required data structures specified in the assignment must be your own implementation: as in BST (or balanced tree) and linear linked lists. Once you meet the basic requirements of the assignment, you are allowed to use libraries for any subsequent data structures.

Program Requirements

It is Rose Festival time! This is always a great time of the year. It is also great to have the Carnival Mid-way right near us at Waterfront Park. But, have you ever thought about what happens behind the scenes to organize such an event? There are rides, games, food vendors, music venues, vendors selling balloons or other fun gadgets. There are photo booths and of course porta-potties.

To start, with program #4, your job is to create a program that will help the Rose Festival organizers figure out where to put everyone. They are wanting a circular loop for people to walk through and experience the various activities. But, they need to be careful not to have two different corn-dog vendors next to each other or music venues too close. So, when new participants apply Carnival Mid-way, your software will help the organizer determine if there is a need and where they should be placed (i.e., which booth number!). For example, when the second music venue applies, our program will make sure that they are placed at the opposite end of the park. Or, once the seventh corn dog vendor applies, they really need to be turned down!

To simplify this, you are allowed to pick just three different types of types of vendors (e.g., carnival activities which are rides and games, vendors which are food or trinkets, and talent which could be music or comedy skits). Use OO to push up the common elements and derive those that are different. In your data structure, use dynamic binding with base class references referring to derived class objects!

Then, in Program #5, your job will be to have a way for people wanting to go to the Carnival Mid-way to look up what is available by name (which is the BST or 2-3 tree). So, if I am dying for a corn dog, I want to know where the vendors are that will let me eat one (I must be hungry right now!).