Cameron Shore
502B
Teacher: Prof. Clark Olson
TA: Carl Mofjeld

Program 4 Design Document

This program will be used to automate inventory (buying, selling, displaying) for a store with 3 kinds of collectibles. This **Collectible** class is a pure virtual class whose base class is the **Comparable** class from program 1 this is a requirement since all classes will be stored in a **SearchTree** class BST. Collectible will have all methods from comparable class for its derived classes to override. 3 classes will be derived class of collectibles to represent the 3 kinds of collectibles that the store will contain. Derived classes: **Coin**, **Comic**, and **Sports card** each has specific member variables for each. Each of these classes will have a buy, sell, and display method to represent the 3 possible transactions associated with those classes as well as override all the methods from the Comparable class. **Customer** class will also be derived from the comparable class this will be a person associated with a specific number; it will have a data member to hold their transaction history, and overrides for all comparable methods. Finally, there will be a **manager** class (ShopManager) to handle data flow between all the classes. Main will consist of 3 stages first it will read an initialization file that contains the stores initial collectibles. Next it will read a customer initialization file to store customer information. Finally, it will read a transaction file which will be used to perform 5 different actions a multitude of times. Prototype main:

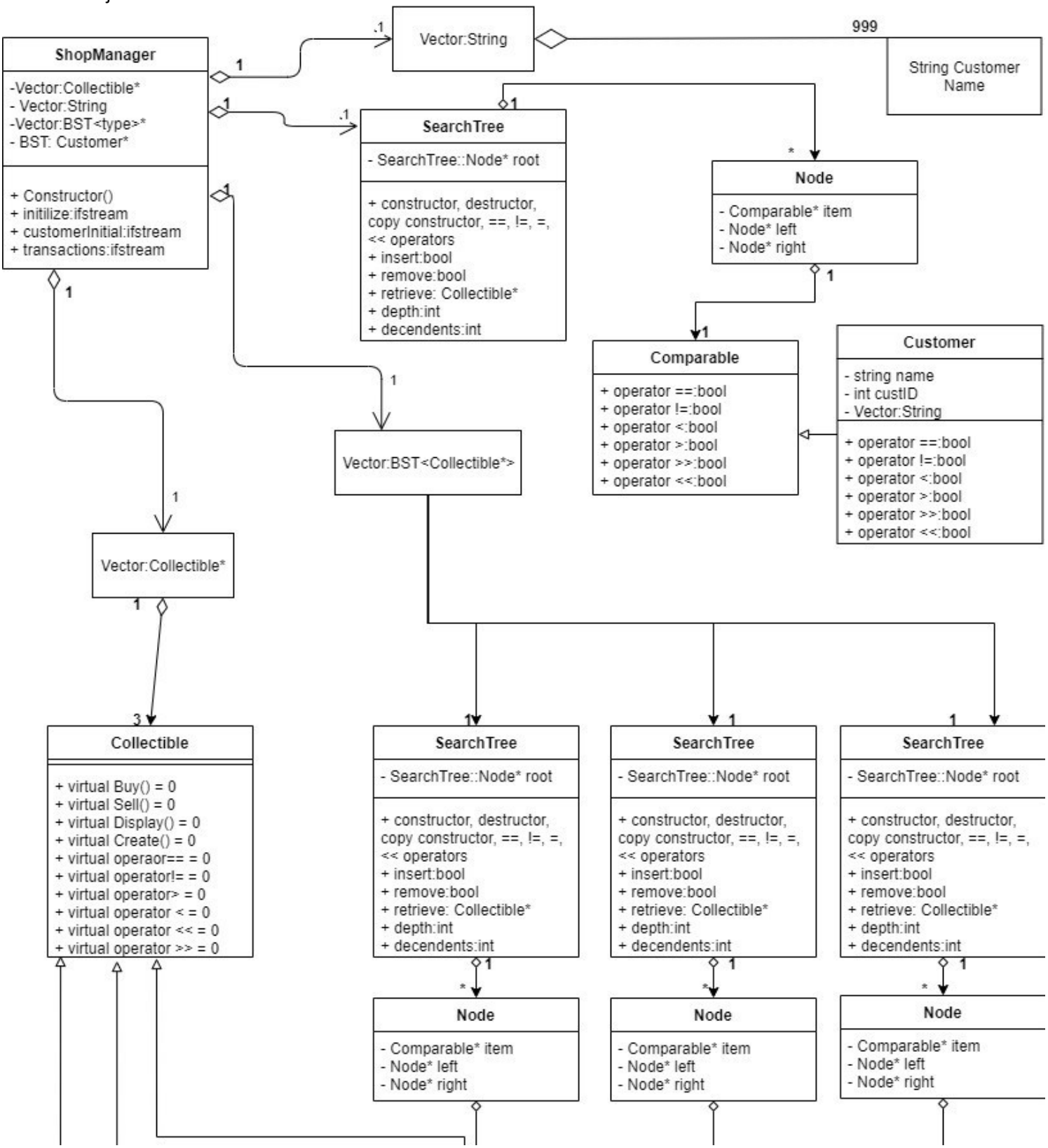
```
int main() {
  fstream infile("initial.txt");
  fstream infile2("customers.txt");
  fstream infile3("transactions.txt");
  // ShopManager mngr; // create manager class
  if (!infile) {
    cerr << "initialization file could not be opened." << endl;
    return 1;
  }
  // mngr.initialize(infile);
  if (!infile2) {
    cerr << "Customer file could not be opened." << endl;
    return 1;
  }
  // mngr.customerInitial(infile2);
  if (!infile3) {
    cerr << "Transaction file could not be opened." << endl;
    return 1;
  }
  // mngr.transactions(infile3);
  return 0;
}
```

After manager class is created, which consist of 1 vector to store dummy objects of each collectible class in order to instantiate classes (through hashing) no matter what derived class they are, 1 Vector will store customers names based upon their ID number, the customer objects themselves will be stored in a SearchTree the last vector will hold inventory of the store, which are SearchTree's of each type of collectible. All of these vectors will have hash functions specific to each one.

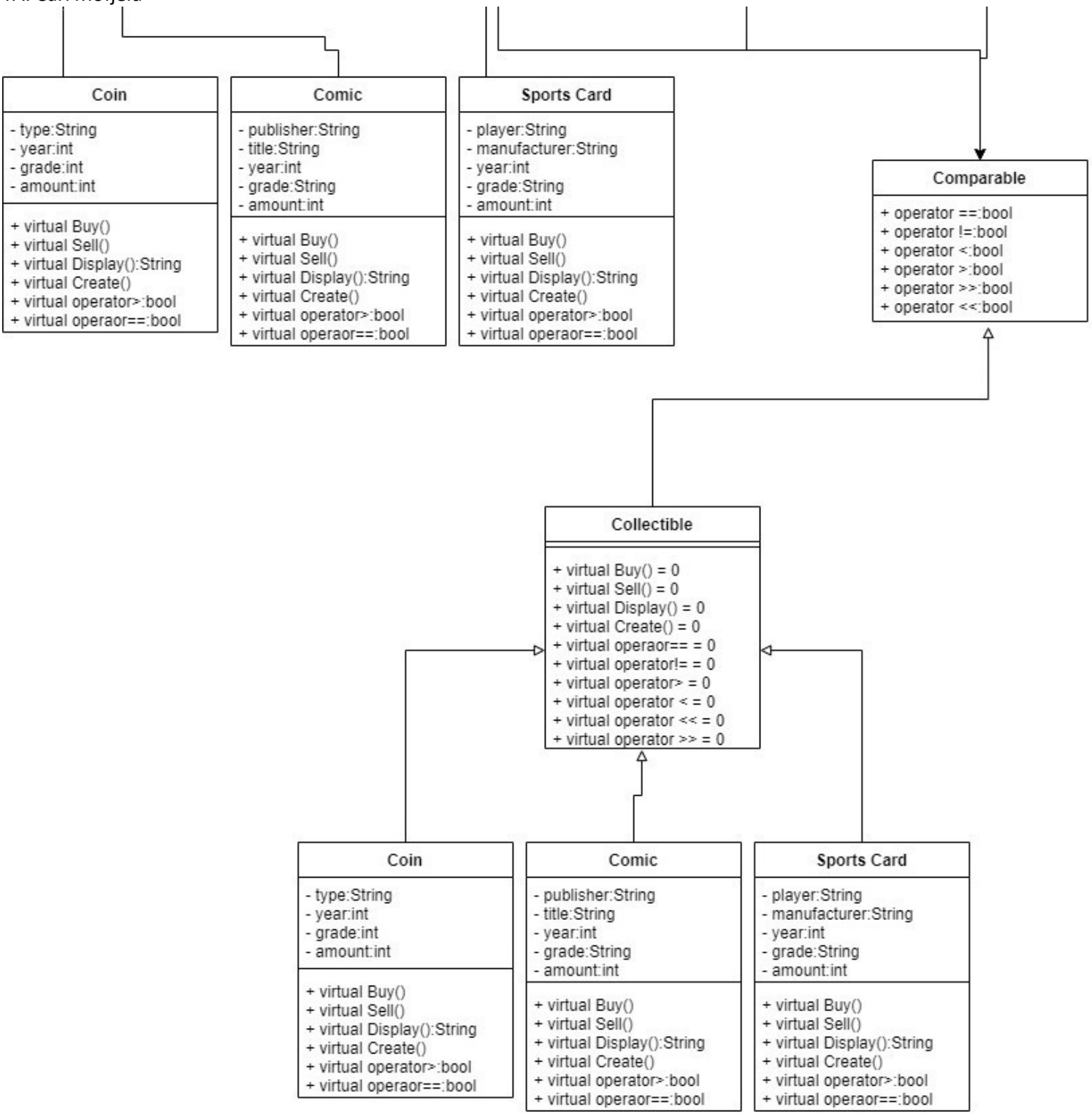
Each file is verified before being passed to the Shop manager which will read the files until the eof is reached. The initialization file will hash to the appropriate dummy object, create an instance of it, then read the data into that object which will then be stored in the appropriate SearchTree via a hash method that will point to the right SearchTree. customerInitial method will store customer names in a hash file based off of their ID, then store the customer objects in SearchTree. The transaction file will call methods on location in the inventory of the given item (through hashing) to the vector containing dummy objects and call specific methods for them.

Class Diagram *subject to change*:

Cameron Shore
502B
Teacher: Prof. Clark Olson
TA: Carl Mofjeld



Cameron Shore
502B
Teacher: Prof. Clark Olson
TA: Carl Mofjeld



Cameron Shore
502B
Teacher: Prof. Clark Olson
TA: Carl Mofjeld

Memory Diagram: