**CLASSES**

Bank

* Processes all transactions: Withdraw, Display, Open, Delete, Deposit.
* Receives requests from the User/Driver and returns information to the User.
* Contains Queue of transactions as a private member.
* Contains BSTree of accounts as a private member.

BSTree

* Stores accounts as an ordered tree, consisting of Nodes of Account objects.
* Account Nodes can be inserted and deleted.

Account

* Contains these data members: name, account type, ID, balance, and transaction history.

**HEADER CODE**

class Bank

{

public:

Bank();

~Bank();

Queue getTransactions();

bool setTransactions(file input);

bool processTransactions();

bool Deposit();

bool Withdraw();

bool Open();

bool Delete();

vector<string> DisplayHistory();

vector<string> DisplayBank();

private:

Queue transactions;

BSTree accountDatabase;

};

class BSTree

{

public:

BSTree();

~BSTree();

bool Insert(Account \*);

bool Retrieve(const int &, Account \* &) const;

void Display() const;

void Empty();

bool isEmpty() const;

private:

struct Node

{

Account \*pAcct;

Node \*right;

Node \*left;

};

Node \*root;

};

class Account

{

public:

Account();

~Account();

private:

vector<string> history;

string accountName;

string accountType;

int accountId;

int accountBalance;

};