Main (It is a driver)

# include “Banker”

Banker Jolley = new Jolley();

Jolley.ReadScript();

Jolley.ConductTransaction();

Jolley.ShowAllBalance();

Banker (Does the three main actions)

# include Transaction / Account / BSTree

(Methods)

1. Reading from the file and put these transactions into queue
   1. Make if statement for each first letter
2. Take each queue and conduct transaction process
   1. Make if statement for each first letter
3. Print out final balance for all accounts

(Variables)

1. BSTree object
2. Transaction objects stored in vector

BSTree (It focuses on managing accounts)

# include “Account”

1. Several methods to retrieve, add and display accounts

Account (Biggest class, does all account calculation and holds each fund object )

* Deals with O, D, W, T and H actions
  + All the methods must be read from Transaction objects
  + For Withdraw and transfer, there is a method for Linked funds
    - Check whether it’s linked fund or not, then make suitable deduction method.
  + All the calculated account balance is surely stored in an each object if Fund Class.
* When an account created, 10 objects of Fund class will also be created

Fund (It mainly records each transaction to each fund)

* It holds the **amount of balance,** and stores transaction objects in fund’s object’s vector.
* Main methods are to show the fund’s balance. And some accessors.

Transaction (This class’s object will be stored in a queue in Main class)

* It just records the transactions with Constructor, accessors and <<operator.
* It only holds variables to display the transaction information



1. Transaction completion

**TEST: Make a test case to make sure all the transaction objects are stored in the queue in Main class**

1. Modify to be able to use BSTree from Transaction object
2. (ALL) Create BSTree class
   1. Account object only holds ID
3. (Only Frame) Create big picture of Account class
   1. Complete Account.h
   2. Write down easy methods in Account.cpp
4. (ALL) Create Fund class
5. (ALL) Complete Account class

**TEST: Make a test case for Account class**

1. Complete Banker class