

Encapsulate concrete details

Encapsulate what might change

What do the current requirements suggest?

- A Bank is made up of Accounts (at most 100)
- Accounts can be either a Savings or Checking Account
- Commands come from a Text file
- Accounts are initialized from a Text file
- Each Account type has a different fee structure
- Each Account has the same basic operations
 - (GainInterest, Withdraw, Deposit, Monthly Update, Display)

What would I expect to change in the future?

- Accounts source (not just text file)
- Commands source (not just text file)
- New command types (Transfer?)
- Add new types of accounts (Loan? CD? IRA?)
- get rid of static array of 100 account pointers
- long long as money is limiting (no currency)
- Keep a transaction history per account
- Add generation of year end tax documents (total interest earned per account)
- New interest calculation strategies

What does the UML tell us?

What is encapsulated in each subsection?

How well are future changes isolated?

What if you add a new command to Command.txt

What if you add a new source of Commands?

Let driver just construct objects and put them together. Put loops and ifs in objects as much as possible.

```
int main ()
{
    Bank cTheBank;
    FileAcctReader cTheFileAcctReader("Accounts_Small.txt");
    FileCommandReader cTheFileCommandReader (&cTheBank, "Commands_Small.txt");

    cTheBank.addAccounts (cTheFileAcctReader);

    cTheBank.runCommands (cTheFileCommandReader);

    return EXIT_SUCCESS;
}
```