Name: Ronald Macmaster

UT EID: rpm953

PA2: Fun with Bank Accounting!

1) Analysis

Problem Statement:

Implement the BankAccount class developed in lecture, and write a driver for the class.

The driver will create customers which own up to four bank accounts and process transactions across those accounts.

Finally, the driver should compile and return a bank account summary.

Input:

A sequence of bank account transactions input through various dialogue boxes

The dialogue boxes will prompt the user through a GUI implemented with the JOptionPane class.

Example transaction: **1 D 5000.00 C**

Output:

A final bank account summary will be compiled for each customer and each of their accounts

Transaction actions will be logged on the screen. Invalid inputs will be logged and discarded.

The account summaries will be output to the screen at the very end.

Questions?

What classes other than a BankAccount class should I create?

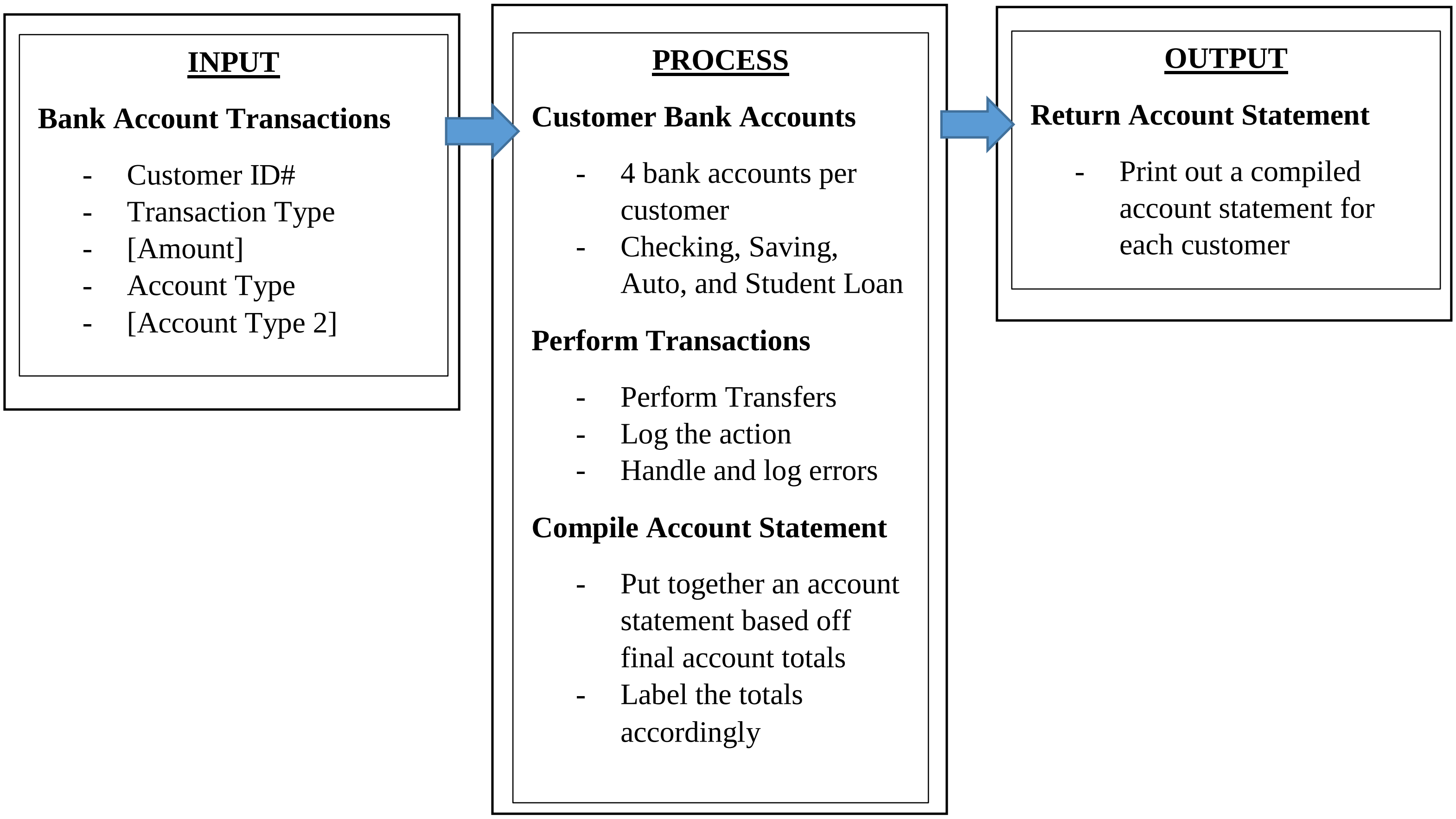
(Customer & BankAccountManager)

How much will I need to know about Swing GUIs before advancing the development process?

2) Design

Architecture Models:

1) **System IPO Model**



INPUT

Bank Account Transactions

- Customer ID#
- Transaction Type
- [Amount]
- Account Type
- [Account Type 2]

PROCESS

Customer Bank Accounts

- 4 bank accounts per customer
- Checking, Saving, Auto, and Student Loan

Perform Transactions

- Perform Transfers
- Log the action
- Handle and log errors

Compile Account Statement

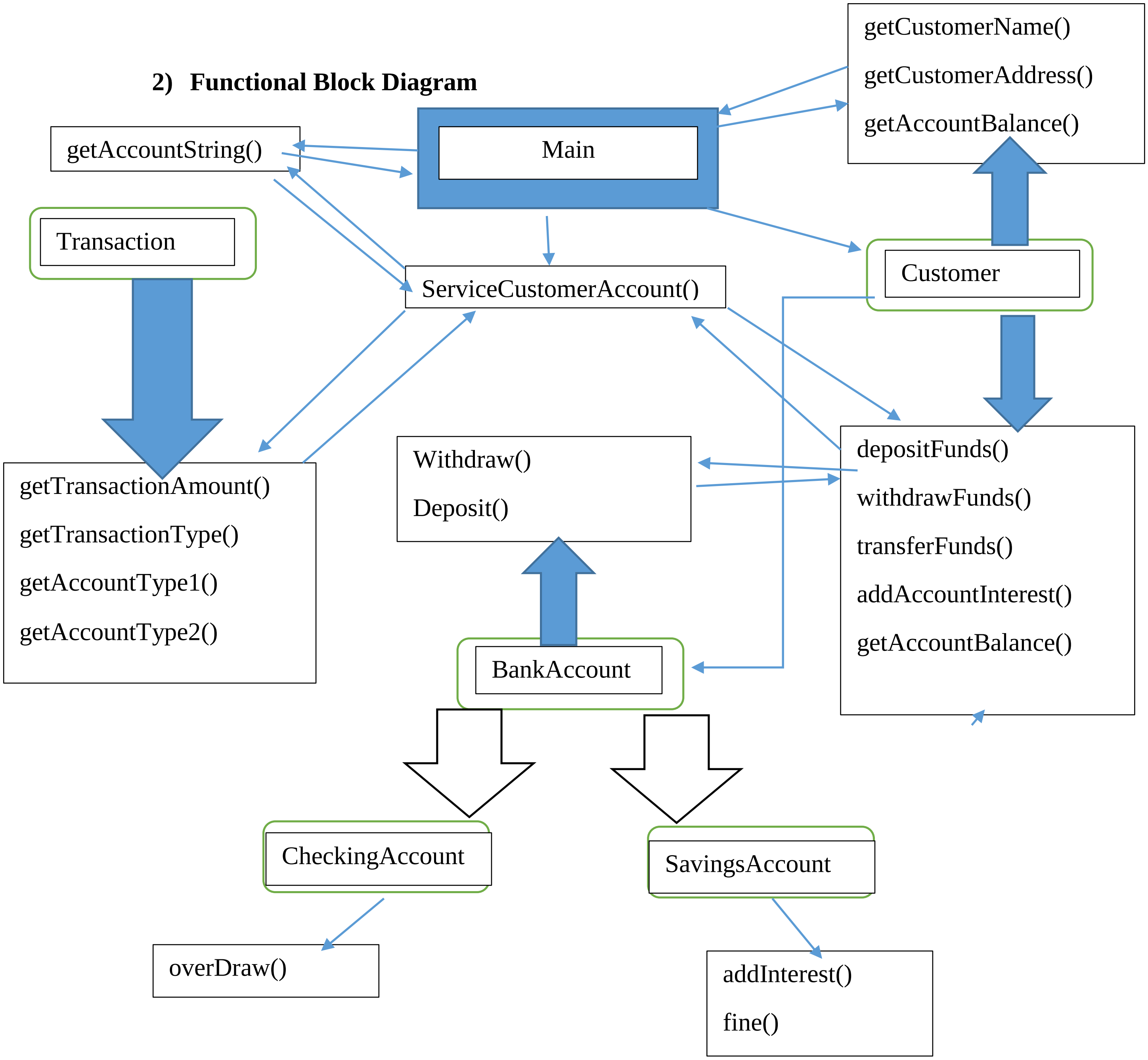
- Put together an account statement based off final account totals
- Label the totals accordingly

OUTPUT

Return Account Statement

- Print out a compiled account statement for each customer

2) Functional Block Diagram



Algorithms

Driver Algorithm: (BankAccountManager)

1) Create bank account list / database
2) Prompt the bank teller for a transaction
3) Process and log the transaction
4) Perform the bank account transaction
5) Ask to continue. (Y/N)? Yes: repeat from 2)
6) Compile and display a bank account statement for each customer

Transaction Processing:

```
Parse Customer ID and Action Code

Withdrawals, Deposits, and Transfers Requests:

    Parse Amount and Account Type #1

Interest and Balance Requests:

    Parse Accouont Type 1

Transfer Payments:

    Parse Account Type #2
```

Account Transaction Algorithms:

Withdraw:

```
If amount > balance:
    Overdraft for Checking account
    Insufficient Funds otherwise
Decrement account balance by withdrawal amount
```

Transfer:

```
If amount > source balance:
    Overdraft for Checking account
    Insufficient Funds otherwise
Withdraw from funds source account
Deposit funds into destination account
```

Deposit:

```
Increment account balance by deposit amount
```

Interest:

```
If the account is a Savings Account:
    Increment account interest to Account Balance
Invalid Account Type Otherwise
```

Get Balance:

```
Return the relevant account balance
```