# Design Overview for *Banking program*

Name: Thomas Couser

Student ID: 692529

# Summary of Program

This program will create and manage bank accounts with a variety of features. Accounts can be created, transactions can be performed, balance check, transaction history, save account information to a txt file, and exit the program. Below is a link to a short youtube video demonstrating its functionality.

<https://www.youtube.com/watch?v=Qo5JUrffZSs&t=41s>

# Required Data Types

Describe each of the custom data types (data classes and enumerated types) you will create using the following table (one per type). Repeat the tables if you need to define more.

Table 1: Bank. The object that stores the bank accounts.

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| accounts | dictionary | A dictionary containing all accounts with a key of the account number and the value being the account |
| transactions | dictionary | A dictionary containing all transactions with the key being the account number and the value being a list of transactions |

Table 2: BankAccount. Each bank account is an object with the following properties.

|  |  |  |
| --- | --- | --- |
| Property | Type | Notes |
| name | string |  |
| account\_number | int |  |
| balance | float |  |

Table 3: MenuOptions. The main menu and its options.

|  |  |  |
| --- | --- | --- |
| Member name | Member value | Notes |
| CREATE\_ACCOUNT | “Create account” |  |
| PERFORM\_TRANSACTION | “Perform transaction” |  |
| CHECK\_BALANCE | “Check balance” |  |
| TRANSACTION\_HISTORY | “Transaction history” |  |
| VIEW\_ALL\_ACCOUNTS | “View all accounts” |  |
| SAVE | “Save” |  |
| EXIT | “Exit” |  |

…

# Overview of Program Structure

List the key functions you are going to need to create for this program and where they will reside (most may be in the module containing main(), which is fine). For each function provide its name, return type (if known) and a brief description of what it will do.

|  |  |  |  |
| --- | --- | --- | --- |
| Module or class | Name | Return type | Description |
| banking\_functions | get\_integer | int | Takes a prompt as a parameter, then returns a validated integer from the user |
| banking\_functions | get\_float | float | Takes a prompt as a parameter, then returns a validated float from the user |
| banking\_functions | get\_enum\_choice | str | Takes an enumerated class as a parameter, then returns the string value |
| Bank | create\_account | BankAccount | Prompts user for an account name and a starting balance, gets a generated 8 digit account number, and returns the account object |
| Bank | get\_account | BankAccount | Takes an account number and returns the account if it exists. |
| Bank | transaction | str | Prompts the user for an account number, presents a menu with the options to deposit, withdraw or cancel, then gets a float and changes the account balance accordingly. |
| Bank | get\_transaction\_history | list | Takes an account number as a parameter and returns a list of the transaction history of that account. |
| Bank | check\_balance | str | Prompts the user for an account number then displays the balance of the associated account |
| Bank | view\_all\_accounts | none | Prints the name of each account currently in the system. |
| Bank | show\_transactions | none | Displays each transaction for a certain account number. |
| BankAccount | deposit | none | Takes a float as input then adds it to its balance. |
| BankAccount | withdraw | none | Takes a float as input then removes it from its balance. |
| BankAccount | get\_balance | float | Returns the current balance of a specific BankAccount object. |
| main | save | none | Takes a path as parameter and saves each account name and number to a txt file in that location. |
| main | generate\_account\_number | int | Returns a random 8 digit integer |

*Don’t spend too long on this at this stage. Focus on the main things you think you are likely to need and you can build on this as your program develops. (Remember to create a structure chart showing how the functions are related too.)*