**Bank Account System Algorithm Design**

**Benjamin Rezentes**

**Bmr0208 –** [**Benjaminrezentes@my.unt.edu**](mailto:Benjaminrezentes@my.unt.edu)

**CSCE1040 – Section 310**

**1. Include Necessary Header Files**

* Include standard libraries like <iostream> <string> <vector> etc…
* Include the “Account.h” and “Project.h” header files for the Account.cpp class definition file and Project.cpp file.

**2. Declare the Account Class (Account.h)**

* **Private Member Variables:**

private:

     int\* accNumber;

*string* name;

      float balance;

      static int nextAccNumber;

* **Public Member Functions:**

public:

      Account();

      Account(const *string*& *userName*, float *initialBalance*);

      Account(const *Account*& *other*);

      ~Account();

      void setName(*string* *name*);

      void setBalance(float *amount*);

      void deposit(float *amount*);

      void withdraw(float *amount*);

      int getAccNumber() const;

*string* getName() const;

      float getBalance() const;

      int getNextAccNumber();

**3. Define the Account Class Implementation (Account.cpp)**

* Implement the **constructor**, **destructor**, **copy constructor**, and **copy assignment operator**.
* Implement **accessor** and **mutator** functions for the member variables.
* Define the **static variable initialization** and overloaded  **operator** for formatted printing of account details.

*Account*::Account(const *string*& *userName*, float *initialBalance*) {

   // Dynamically creating memory for accNumber

   accNumber = **new** int;

   // Setting pointer to nextAccNumber value

   \*accNumber = nextAccNumber;

   name = *userName*;

   balance = *initialBalance*;

   // Incrementing nextAccNumber so the next account has a different number

   nextAccNumber++;

}

**4. Declare and Implement the Project Functions (Project2.h & Project2.cpp)**

* **Include necessary libraries** and Account.h header.
* **Functions necessary for adding accounts, changing the balance of those accounts, and finally displaying those accounts.**

**5. Main Program Logic (Project2.cpp)**

* **Print program information** (your name, EUID, email, course, etc.).
* **Loop until user exits** (option 5):
  + Inside the loop:
    1. **Display the Menu**:

void displayMenu() {

   cout << "----------M E N U----------" << endl;

   cout << "|  1. Open Account        |" << endl;

   cout << "|  2. Deposit             |" << endl;

   cout << "|  3. Withdraw            |" << endl;

   cout << "|  4. Print Accounts      |" << endl;

   cout << "|  5. End Transaction     |" << endl;

   cout << "---------------------------" << endl;

   cout << "-------> ";

}

* + 1. **Prompt for user input** for menu selection.
    2. **Switch statement** to perform the appropriate action:
       - **Option 1**:
         * Call addNewAcc() to create a new account.
         * Prompt for customer name and initial deposit.
         * Generate the account number using the static integer.
         * Display account details with formatted balance.

* + - * **Option 2**:
        + Call depositToAcc() to deposit money.
        + Prompt for account number and deposit amount.
        + Update balance and display the updated details.
      * **Option 3**:
        + Call withdrawFromAcc() for withdrawal.
        + Prompt for account number and withdrawal amount.
        + Ensure sufficient funds are available before processing the withdrawal.
      * **Option 4**:
        + Call displayAccounts() to display a list of all accounts.
        + If no accounts exist, display a message indicating so.
        + Otherwise, print each account's number, name, and balance.
      * **Option 5**:
        + End the program with a message and exit.