# BISY1003/ISY1003/ISY104 Foundations of Programming

**Assignment 2 Guide**

For the Assignment 2, you are required to create a **modular program** that performs a **number of transactions**. Tyr to include all the programming constructs which you have learned so far like variables, methods, lists, loops, decision statements, file IO, exception handling etc.

Your program should be **menu driven** which provided user with a **list of choices** to perform various transactions. It should allow the user to continue using the system until he/she selects the exit option.

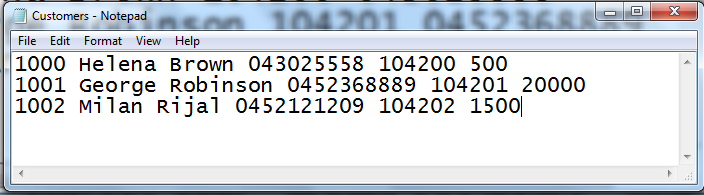
Below is a sample guide on Banking System transactions which will help you to understand the requirements of the assessment.

**Sample Banking System**

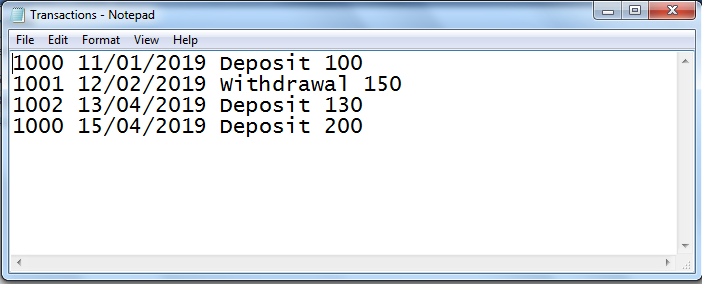
The objectives of this system is to maintain customers’ detail and their transactions. There are two kinds of transactions this system handles 1) Deposit and 2) withdrawal.

Here you are required to store **customers and transactions details permanently** so you can use files for persistence storage. For example we can manage two separate file a) customers.txt and b) transactions.txt.

**Customers file** will be holding data for customers like unique account number, first name, last name, contact number, unique customer number and current balance.



**Transactions file** will hold transaction details account number, date of transaction, type of transaction (deposit or withdrawal, amount of transaction) like below:



**Functionalities:**

Some sample list of functionalities you might think of:

1. Add new customer
2. Perform deposit transaction
3. Perform withdrawal
4. View current balance
5. View transaction history
6. Exit

1) **Add new customer:**

* This function allows user to create a new customer.
* You need some user input like name, phone number and the opening balance (which will be current balance).
* Try to generate the unique account number and customer number i.e. read those numbers from Customer file of the last customer and increment by 1 or as per your chosen strategy to generate unique numbers.
* Append the new customer once you have all the required details in existing Customer file.

2) **Perform deposit/withdrawal transaction:**

* For this functionality, you need the account number from the user and perform the validation to check whether the account exist or not.
* If account exist, ask for type of the transaction and amount. Take consideration of all necessary validations like the sufficient fund availability for withdrawal. Append the transaction details in Transactions file and update the current balance for that account in Customer file.
* If account not exist, ask for valid account number.

3) **View current balance:**

* Using this functionality customer can view their current balance.

4) **View transaction history:**

* Customer can view their transactions using this functionality.
* You can implement the better versions of this functionality by including various parameters like
  + display transactions between particular date range only,
  + only deposit or withdrawal transactions
  + Transactions between amount range etc.

Think about the validation for input data and handle all the possible run time exceptions to deliver robust system.

Create user friendly system which is easy to operate with appropriate self-descriptive messages.