



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

FACULTY OF COMPUTER SCIENCE

UNIVERSITI TEKNOLOGI MALAYSIA

**DATA STRUCTURE & ALGORITHM - SECJ 2013 Section 02**

---

**ASSIGNMENT 1 REPORT**

---

**TITLE : BANKING TRANSACTION SYSTEM**

**PREPARED BY : DACCrew**

NAME	MATRIC NO
CHAI YU TONG	A22EC0145
NUR FARAH ADIBAH BINTI IDRIS	A22EC0245
WONG QIAO YING	A22EC0118

**PREPARED FOR:**

**DR. LIZAWATI BINTI MI YUSUF**

## **Table of Contents**

<b>Table of Contents</b>	<b>2</b>
<b>Part 1: Introduction</b>	<b>3 - 4</b>
1.1 Objective of Project	<b>3</b>
1.2 Synopsis of Project	<b>4</b>
<b>Part 2: System Design</b>	<b>5 - 8</b>
2.1 System Class Design	<b>5</b>
2.2 System Algorithm Design	<b>5 - 6</b>
2.3 Data Structure Operation	<b>6 - 7</b>
2.3.1 Sorting Process	<b>6</b>
2.3.2 Searching Process	<b>7</b>

# PART 1 : INTRODUCTION

## 1.1 Objectives

- To simulate the CIMB Banking Transaction System
- To apply the data structure concept in the system
- To gain hands-on experience in applying the data structure concept

## 1.2 Synopsis

We will develop a banking transaction system focusing on sorting and searching operations using fundamental data structure concepts. This system is designed to perform basic transaction operations, including depositing money into user accounts, enabling withdrawal, and facilitating money transfers between accounts. The sorting operations encompass arranging data by name, Identification Card number, account number and balance in ascending order. Additionally, this system supports search operations based on account holder's names and account numbers, allowing users to locate their accounts using unique identifiers.

### The existing class and its attributes:

#### 1. Bank

<code>accName</code>	the name of the account holder <string>
<code>accNum</code>	the account number <string>
<code>ic</code>	the identity card number of the account holder <string>
<code>accBalance</code>	the quantity of the amount of money in the account <double>

### The existing data structure:

#### 1. Array

The array, in this case, stores a list of Accounts with their holder's name, account number, IC number, and Balance. The searching operations iterate through this array to find specific records based on criteria such as name, account number, and IC number.

`Bank[100]` stores a list of user name, account number, IC and balance

## 2. Sorting

The sorting function is defined to sort an array of 'Bank' objects on different criteria. These functions use various partitioning techniques, such as quickSort.

<code>partitionName</code>	partitions the array based on the name attributes during sorting
<code>sortByName</code>	sorts the array 'Bank' objects based on the account name
<code>partitionIc</code>	partitions the array based on the IC attributes during sorting
<code>sortByIc</code>	sorts the array 'Bank' objects based on the IC
<code>partitionAccNum</code>	partitions the array based on the account number attributes during sorting
<code>sortByAccNum</code>	sorts the array 'Bank' objects based on the account number
<code>partitionBal</code>	partitions the array based on the balance attributes during sorting
<code>sortByBal</code>	sorts the array 'Bank' objects based on the account balance

## 3. Searching:

The searching functions are defined to search for records in the array based on name, account number and ic. These functions will iterate through the array and display matching records if found.

<code>searchByName</code>	searches for records in the array based on the account name
<code>searchByAccNum</code>	searches for records in the array based on the account number
<code>searchByIc</code>	searches for records in the array based on the IC

## PART 2 : SYSTEM DESIGN

### 2.1 System Class Design (Class Diagram)

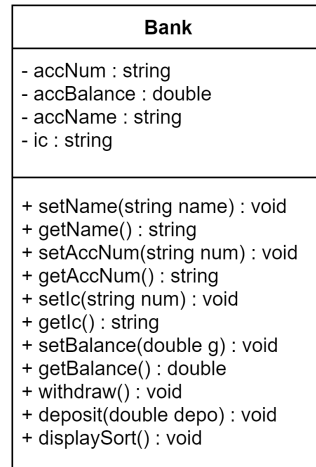


Figure 1: Class Diagram for Bank Entity

### 2.2 System Algorithm Design (Pseudo Code)

1. Start.
2. Display menu.
3. User input choice
4. If choice == 1
  - 4.1. Display Sort by name
    - 4.1.1. Else if choice == 2
      - 4.1.1.1. Display Sort by Account Number
    - 4.1.2. Else if choice == 3
      - 4.1.2.1. Display Sort by IC Number
  - 4.2. Else if choice == 4
    - 4.2.1. Display Sort by Account Balance
  - 4.3. Else if choice == 5
    - 4.3.1. Display Search by Name
  - 4.4. Else if choice == 6
    - 4.4.1. Display Search by Account Number
  - 4.5. Else if choice == 7
    - 4.5.1. Display Search by IC Number

- 4.6. Else
  - 4.6.1. Exit
- 5. End

## 2.3 Data Structure Operation

### 2.3.1 Sorting Process

As for the sorting process, we will use a quicksort method in the banking system. Quicksort is a divide-and-conquer algorithm that involves choosing a pivot and then partitioning the array around the pivot. Quicksort involves moving elements less than the pivot value on the left, while the elements bigger than the pivot will be on the right. Thus, the pivot is now in the correctly sorted position.

The `sortByName()` function sorts the array of `Bank` objects according to the `accName` attribute using the quicksort algorithm in ascending order by partitioning the arrays using the `partitionName()` function. Meanwhile, the `sortByNum()` function sorts the array of elements according to the `accNum` attribute using the quicksort algorithm in ascending order by partitioning the collection of arrays using the `partitionaccNum()` function. Moreover, `sortByIc()` function sorts the arrays according to the `ic` attribute using the quicksort algorithm in ascending order by partitioning the collection of bank arrays using the `partitionIc()` function. The `sortByBal()` function sorts the array of elements according to the `accBalance` attribute using the quicksort algorithm in ascending order by partitioning the collection of arrays using the `partitionBal()` function. All the sort functions recursively call itself until all elements are sorted.

### 2.3.2 Searching Process

The system is applying the sequential search technique to search for the data. Data can be searched through the attribute. If the data exists, the whole piece of details will be returned as the output after the search process has been performed. On the other hand, if no such data exists in the array, the system will print out the error message.

In this process, users can search for their intended data by Name, Account Number, and IC Number. Therefore, there exist three search functions for each attribute. The functions included are `searchByName()`, `searchByAccNum()`, and `searchByIc()`. After the user chooses the method of searching, the system will request for an input. Then, the system iterates through the `Bank` array object starting from index 0 and makes a comparison with user input. The system will display the complete information about the account if a match is found. The searching process will continue until the last object in the array.