**Object Oriented Programming**

**Assignment # 01**

**Note:**

**Submission Deadline:** Due Date: Friday 23rd February 2024 (3pm)

- First think about a problem statement and then write/draw your logic on paper.
- After designing the logic on paper, code the problem statement on any editor (VS Code, Gedit, etc).
- Copied tasks will be awarded **zero** marks without any investigation.
- Comments you code properly.
- Assignment After Due DateAssignment will not be Accepted.
- Assignment should be submitted in a zip file named as 23P-9307_M_Kaif.
- The zip file should contain all .cpp files.
- Plagiarism of any shape or form will not be tolerated. In case of plagiarism, the particular question will be marked zero and 50% marks from total obtained marks will be deducted.

## Problem: 1 | Bank Management System

**Note:** Store Arrays Dynamically.

You're developing a bank account management system for a financial institution. The system should allow customers to deposit/withdraw funds, and view their account balance. Design a C++ program to implement this functionality using arrays, loops, and functions.

Tasks:

**1**.Define arrays to Dynamically store account information such as account number, account holder name, balance, and account type (savings, checking ).

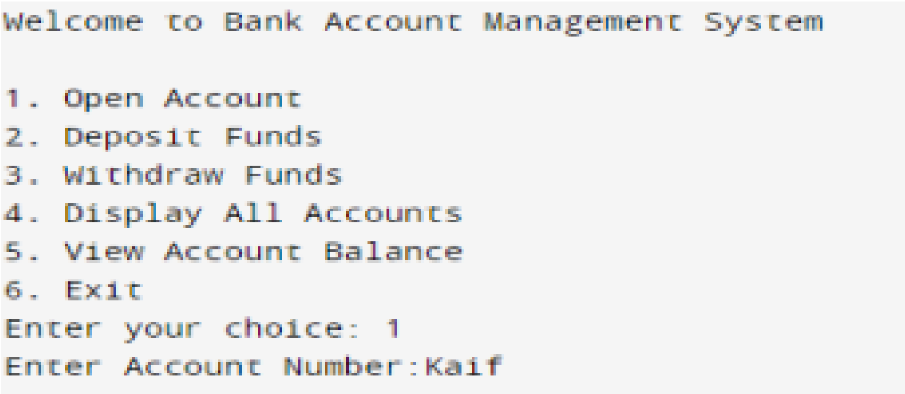
**2**.Implement functions to deposit funds, withdraw funds,display all accounts and view account balances. **3**.Overload functions to handle cases where customers deposit or withdraw different amounts of money. **4**.Utilize loops to search for specific account records and perform transactions.

**5**.Ensure that the program handles scenarios where customers attempt to withdraw more funds than available in their account.

**Hint:**

*Create Separate arrays for A/c no,A/c Holder Name and balance etc.*

**Sample Output**:



```
Welcome to Bank Account Management System

1. Open Account
2. Deposit Funds
3. Withdraw Funds
4. Display All Accounts
5. View Account Balance
6. Exit
Enter your choice: 1
Enter Account Number:Kaif
```

## Problem: 2

Create a menu-driven program which will have three of the following functionalities,

1. Add new number
2. Delete a number
3. Print the array
4. Print sum of array
5. Exit Program

Initially, create a dynamic array of size 5 and give random values from 0 - 10 (using rand function) in the array.
- The function add_new_number should take one number from user and insert it into array **by increasing size of array**. The size of the array should be increased by using a resize function which has been briefly explained below,
*Create a function called "ResizeArray" to be used whenever a new number is to be added in your array. The user should not be aware that the size of the array is changing. Rather, he/she should simply be allowed to keep adding new numbers until he/she is done, and ResizeArray should be called (transparently to the user) whenever a user likes to enter new number. Each call to ResizeArray should increase the size of the existing array by one.*
Similarly, delete a number should take input from user, search it in the array and delete it from it (Remember, by deletion, it means that you must decrease the size of array by one as well). If the number does not exist in the array, simply display an error message.

**Problem: 4 | Reverse Array**

Write a function that reverses the elements of an array so that the last element becomes the first, the second from the last becomes the second , and so forth . The function is to reverse the elements in place—that is, without using another array. ( It is permissible to use a variable to hold an element temporarily.) Then write a test driver to test your function . Test it twice , once with an even number of elements in the array and once with an odd number of elements in the array.