Week 4-Task 1

PROJECT
SIMPLE BANKING SYSTEM

PREPARED BY

Sana Fatima

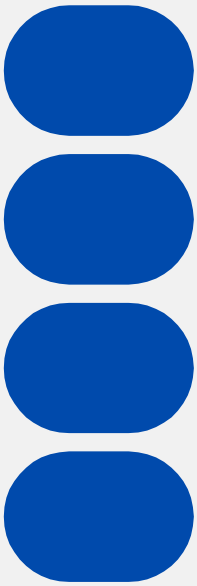
sana.faati@gmail.com

PREPARED TO

A&D Tech

ad.techinnov25@gmail.com

TABLE OF CONTENTS



Problem Statement

Implementing the Program

1. 1.Create a project and name it as Simple Banking System.
2. 2.Define a class BankAccount.
3. 3.Implement public methods for:

- Initializing a new account (constructor with initial balance and account holder details).

- Depositing money (deposit method that increases the balance).
- Withdrawing money (withdraw method that checks if sufficient funds are available and then decreases the balance).
- Checking the current balance (getBalance method that returns the current balance).
- Displaying account details (displayAccountInfo method to show all account details)

SOLUTION OVER VIEW

The solution overview for this program is:

The goal is to implement a simple banking system in C++ that allows users to:

1. Initialize a new account.
2. Deposit money.
3. Withdraw money.
4. Check the current balance.
5. Display account details.

The system consists of a BankAccount class and a main driver program to interact with the user. The BankAccount class encapsulates the account details and provides methods to manipulate and retrieve account information. The main program uses a menu-driven interface to allow the user to perform different banking operations.

BankAccount Class Implementation

The BankAccount class includes:

- Constructor: Initializes the account with the account holder's name, account number, and initial balance.
- Deposit Method: Increases the balance by a specified amount.
- Withdraw Method: Decreases the balance by a specified amount if there are sufficient funds.
- Get Balance Method: Returns the current balance.
- Display Account Info Method: Displays all account details.

CHALLENGES AND RESOLUTIONS

Input Validation:

- **Challenge:** Ensuring all user inputs are valid integers or doubles. Handling incorrect inputs gracefully without crashing the program.
- **Resolution:** Implement input validation using while loops and cin to check for valid inputs. Clear and ignore invalid inputs to prevent the program from crashing.

Maintaining State:

- **Challenge:** Correctly updating and maintaining the account balance after deposits and withdrawals. Ensuring that operations are only performed on valid accounts.
- **Resolution:** Use member functions to update the balance and check the account number to ensure operations are performed on valid accounts.

User Experience:

- **Challenge:** Creating a user-friendly interface for the menu-driven program. Providing clear error messages and instructions to the user.
- **Resolution:** Design a clear and simple menu interface. Provide detailed error messages and prompt users to re-enter data when invalid input is detected.

INSTRUCTIONS

Instructions for Implementation

1. **Define the BankAccount Class:**
    - Include private member variables for the account holder's name, account number, and balance.
    - Implement the public methods as described.
2. **Main Program:**
    - Create a menu-driven interface to interact with the user.
    - Use a do-while loop to repeatedly show the menu and perform the selected operation.
    - Implement input validation to ensure the user enters valid data.