

DEPARTMENT OF BASIC SCIENCE AND HUMANITIES

INSTITUTE OF ENGINEERING AND MANAGEMENT, KOLKATA

# “BANK MANAGEMENT SYSTEM”

**Submitted by:-**

**Name of the Student: Soumyadip Saha**

**Enrolment Number: 12022002004059**

**Registration Number: 221040110819**

**Section: J**

**Class Roll Number: 51**

**Stream: IT**

**Subject: Programming for problem solving**

**Subject Code: ESC103(pr.)**

Under the supervision of: **Prof.Swarnendu Ghosh**

**Academic Year: 2022-26**

(PROJECT REPORT SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE SECOND SEMESTER)



## CERTIFICATE OF RECOMMENDATION

We hereby recommend that the project prepared under our supervision by **Sambhram Banerjee**, entitled **“”** be accepted in fulfillment of the requirements for the degree of fulfillment of the second semester.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Head of the Department Project Supervisor IEM, Kolkata Basic Science and Humanities

1. ***Introduction:*** A bank management system is a software application that helps manage various banking operations such as customer account management, transaction processing, loan management, and other financial operations. In C language, you can build a bank management system by creating functions that perform different tasks .
2. ***Variable Description:***

Here are the variable descriptions for the bank management system example written in C language:

* **MAX\_ACCOUNTS**: A constant that specifies the maximum number of accounts that can be added to the system. In this example, it is set to 1000.
* **struct Account**: A structure that defines the account details. Each account has an account number, name, and balance.
* **accounts**: An array of type **struct Account** that stores all the account details.
* **num\_accounts**: An integer variable that keeps track of the number of accounts currently stored in the system.
* **add\_account()**: A function that takes in an account number, name, and balance and adds a new account to the **accounts** array.
* **display\_account()**: A function that takes in an account number and displays the corresponding account details.
* **deposit()**: A function that takes in an account number and an amount and updates the balance of the corresponding account by adding the amount.
* **withdraw()**: A function that takes in an account number and an amount and updates the balance of the corresponding account by subtracting the amount, if the account has enough balance.
* **choice**: An integer variable that holds the user's choice from the menu options.
* **account\_number**: An integer variable that holds the account number for various operations.
* **name**: A character array that holds the name of the account holder.
* **balance**: A floating-point variable that holds the balance of the account.
* **amount**: A floating-point variable that holds the amount to be deposited or withdrawn.

These variables and functions together make up the bank management system and help perform various banking operations.

Top of Form

Bottom of Form

1. ***Function Description:***

Here are the descriptions for the functions used in the bank management system example written in C language:

**void add\_account(int account\_number, char \*name, float balance),**

**void display\_account(int account\_number), void deposit(int account\_number, float amount), void withdraw(int account\_number, float amount), int main()etc .**

1. ***Programs :***

#include<stdio.h>

struct BankAccount {

int accountNumber;

float balance;

};

void createAccount(struct BankAccount \*account) {

printf("Enter account number: ");

scanf("%d", &(account->accountNumber));

printf("Enter initial balance: ");

scanf("%f", &(account->balance));

printf("Account created successfully!\n");

}

void deposit(struct BankAccount \*account) {

float amount;

printf("Enter the amount to deposit: ");

scanf("%f", &amount);

account->balance += amount;

printf("Deposit successful!\n");

}

void withdraw(struct BankAccount al successful!\n");

} else {

printf("Insufficient balance!\n");

}

}

void displayBalance(struct BankAccount account) {

printf("Account Number: %d\n", account.accountNumber);

printf("Balance: %.2f\n", account.balance);

}

int main() {

struct BankAccount account;

int choice;

printf("Bank Management System\n");

printf("======================\n");

while (1) {

printf("\n1. Create Account\n");

printf("2. Deposit\n");

printf("3. Withdraw\n");

printf("4. Check Balance\n");

printf("5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

createAccount(&account);

break;

case 2:

deposit(&account);

break;

case 3:

withdraw(&account);

break;

case 4:

displayBalance(account);

break;

case 5:

printf("Thank you for using the Bank Management System. Goodbye!\n");

return 0;

default:

printf ("Invalid choice. Please try again.\n");

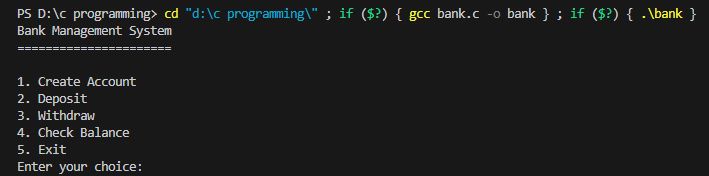
}

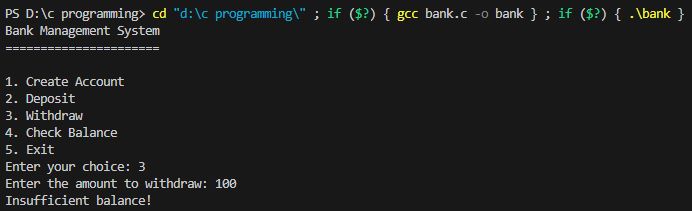
}

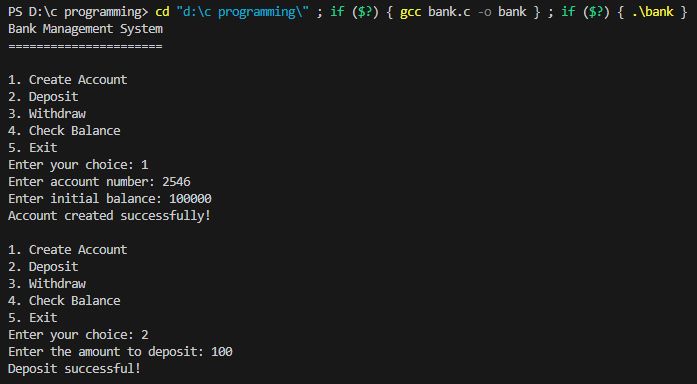
    return 0;

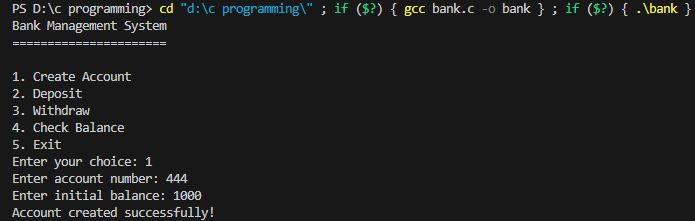
}

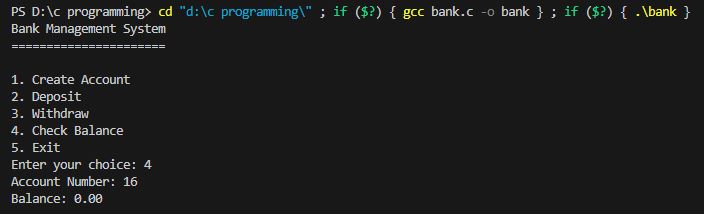
1. ***Outputs:***

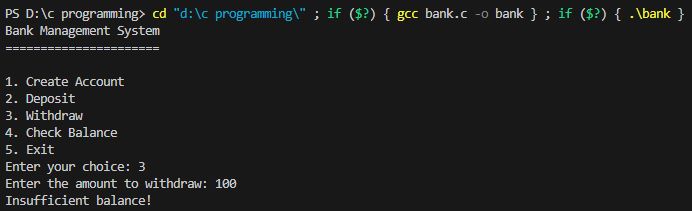
******

****

****





****

**THANK YOU!!**