BANKING SYSTEMS   
DONE BY: HAMSINI BHARAT (RA2111009010076)  
 RAGHAVAN NARSIMHAN (RA2111009010077)

**ABSTRACT:**

**Banking system mini project in C** is a console application without graphics which focuses on providing automated banking services to its customers. The main objective of this project is to develop a banking system for banks to facilitate a faster access to all the banking procedures. It includes designing an application, which could store bank data and provide an interface for retrieving customer related details with better accuracy. In this banking system, one can create a new customer account, access transactions like depositing money or withdrawing money, check account details ,etc., The system will also check the user’s existence in the database and provide the set of services with respect to the role of the user. This banking system project includes the concepts of structures, pointers and functions to store the data of customers. The following flow charts illustrates the mechanism of this project,

1. **main():**

create\_new\_account()

Option

Display.options()

Read option

num\_acc=0

char option

While(1)

***TRUE***

System (“clr”)

(Print statements)

***ELSE* 1**

cash-deposit()

**2**

**3**

cash\_withdrawal()

**4**

account\_information()

**5**

End

**6**

System (“clr”)

1. **Display.options():**

Print statements

1. **Create\_new\_account():**

Print bank\_name, bank\_branch,

Acc\_holder\_name, acc\_number

Write strcpy

statements

Read bank\_name, bank\_branch, acc\_holder\_name, acc\_number

Available\_bal=0

1. **Account\_information():**

Strlen(account[acc\_number-1].bank\_name>0

num\_acc=0

***FALSE***

***TRUE***

num\_acc++

Print bank\_name, branch\_name, acc\_holder\_name, acc\_number, acc\_holder\_address, available\_balance

1. **Cash\_deposit():**

account[acc\_no-1].available\_balance= account[acc\_no-1].acc\_number

Print “The new balance is”, available\_balance

acc\_no=account[acc\_no-1].acc\_number

Print “The current available balance is”, available\_balance

Print “Enter amount of money to deposit”, add\_money

Print “Enter account number”

Read acc\_no

int acc\_no

float add\_money

fl

***TRUE***

acc\_no++

1. **Cash\_withdrawal():**

acc\_no++

strcpy(account[acc\_number-1].bank\_name,bank\_name)

strcpy(account[acc\_number-1].bank\_branch,bank\_branch)

strcpy(account[acc\_number-1].acc\_holder\_name, acc\_holder\_name)

account[acc\_number-1].acc\_number=acc\_number

strcpy(account[acc\_number-1].acc\_holder\_address, acc\_holder\_address)

account[acc\_number-1].available\_balance=available\_balance

acc\_no=account[acc\_no-1].acc\_number

Print “Enter the account number”

Read acc\_no

Print”The current available balance is”

Print “Enter amount money to withdraw’

Read withdraw\_money

int acc\_no

float withdraw\_money

***FALSE***

***TRUE***

**Banking system   
Code:**

#include <stdio.h>

#include <conio.h>

#include <string.h>

#include <stdlib.h>

// Structure declaration

struct acc\_type

{

char bank\_name[20];

char bank\_branch[20];

char acc\_holder\_name[30];

int acc\_number;

char acc\_holder\_address[100];

float available\_balance;

};

struct acc\_type account[20];

/\*

printf("The above structure can be declared using

typedef like below");

typedef struct acc\_type

{

char bank\_name[20];

char bank\_branch[20];

char acc\_holder\_name[30];

int acc\_number;

char acc\_holder\_address[100];

float available\_balance;

}Acc\_detail;

Acc\_detail account[20];

\*/

int num\_acc;

void Create\_new\_account();

void Cash\_Deposit();

void Cash\_withdrawl();

void Account\_information();

void Log\_out();

void display\_options();

/\* main program \*/

int main()

{

char option;

char f2f[50] = "http://Bank2save.com/";

num\_acc=0;

while(1)

{

printf("\n\*\*\*\*\* Welcome to Bank Application \*\*\*\*\*\n");

printf("\nThis demo program is brought you by %s",f2f);

display\_options();

printf("Please enter any options (1/2/3/4/5/6) ");

printf("to continue : ");

option = getch();

printf("%c \n", option);

switch(option)

{

case '1': Create\_new\_account();

break;

case '2': Cash\_Deposit();

break;

case '3': Cash\_withdrawl();

break;

case '4': Account\_information();

break;

case '5': return 0;

case '6': system("cls");

break;

default : system("cls");

printf("Please enter one of the options");

printf("(1/2/3/4/5/6) to continue \n ");

break;

}

}

return 0;

}

/\*Function to display available options in this application\*/

void display\_options()

{

printf("\n1. Create new account \n");

printf("2. Cash Deposit \n");

printf("3. Cash withdrawl \n");

printf("4. Account information \n");

printf("5. Log out \n");

printf("6. Clear the screen and display available ");

printf("options \n\n");

}

/\* Function to create new account \*/

void Create\_new\_account()

{

char bank\_name[20];

char bank\_branch[20];

char acc\_holder\_name[30];

int acc\_number;

char acc\_holder\_address[100];

float available\_balance = 0;

fflush(stdin);

printf("\nEnter the bank name : ");

scanf("%s", &bank\_name);

printf("\nEnter the bank branch : ");

scanf("%s", &bank\_branch);

printf("\nEnter the account holder name : ");

scanf("%s", &acc\_holder\_name);

printf("\nEnter the account number(1 to 10): ");

scanf("%d", &acc\_number);

printf("\nEnter the account holder address : ");

scanf("%s", &acc\_holder\_address);

strcpy(account[acc\_number-1].bank\_name,bank\_name);

strcpy(account[acc\_number-1].bank\_branch,bank\_branch);

strcpy(account[acc\_number-1].acc\_holder\_name,

acc\_holder\_name);

account[acc\_number-1].acc\_number=acc\_number;

strcpy(account[acc\_number-1].acc\_holder\_address,

acc\_holder\_address);

account[acc\_number-1].available\_balance=available\_balance;

printf("\nAccount has been created successfully \n\n");

printf("Bank name : %s \n" ,

account[acc\_number-1].bank\_name);

printf("Bank branch : %s \n" ,

account[acc\_number-1].bank\_branch);

printf("Account holder name : %s \n" ,

account[acc\_number-1].acc\_holder\_name);

printf("Account number : %d \n" ,

account[acc\_number-1].acc\_number);

printf("Account holder address : %s \n" ,

account[acc\_number-1].acc\_holder\_address);

printf("Available balance : %f \n" ,

account[acc\_number-1].available\_balance);

//num\_acc++;

}

// Displaying account informations

void Account\_information()

{

register int num\_acc = 0;

//if (!strcmp(customer,account[count].name))

while(strlen(account[num\_acc].bank\_name)>0)

{

printf("\nBank name : %s \n" ,

account[num\_acc].bank\_name);

printf("Bank branch : %s \n" ,

account[num\_acc].bank\_branch);

printf("Account holder name : %s \n" ,

account[num\_acc].acc\_holder\_name);

printf("Account number : %d \n" ,

account[num\_acc].acc\_number);

printf("Account holder address : %s \n" ,

account[num\_acc].acc\_holder\_address);

printf("Available balance : %f \n\n" ,

account[num\_acc].available\_balance);

num\_acc++;

}

}

// Function to deposit amount in an account

void Cash\_Deposit()

{

auto int acc\_no;

float add\_money;

printf("Enter account number you want to deposit money:");

scanf("%d",&acc\_no);

printf("\nThe current balance for account %d is %f \n",

acc\_no, account[acc\_no-1].available\_balance);

printf("\nEnter money you want to deposit : ");

scanf("%f",&add\_money);

while (acc\_no=account[acc\_no-1].acc\_number)

{

account[acc\_no-1].available\_balance=

account[acc\_no-1].available\_balance+add\_money;

printf("\nThe New balance for account %d is %f \n",

acc\_no, account[acc\_no-1].available\_balance);

break;

}acc\_no++;

}

// Function to withdraw amount from an account

void Cash\_withdrawl()

{

auto int acc\_no;

float withdraw\_money;

printf("Enter account number you want to withdraw money:");

scanf("%d",&acc\_no);

printf("\nThe current balance for account %d is %f \n",

acc\_no, account[acc\_no-1].available\_balance);

printf("\nEnter money you want to withdraw from account ");

scanf("%f",&withdraw\_money);

while (acc\_no=account[acc\_no-1].acc\_number)

{

account[acc\_no-1].available\_balance=

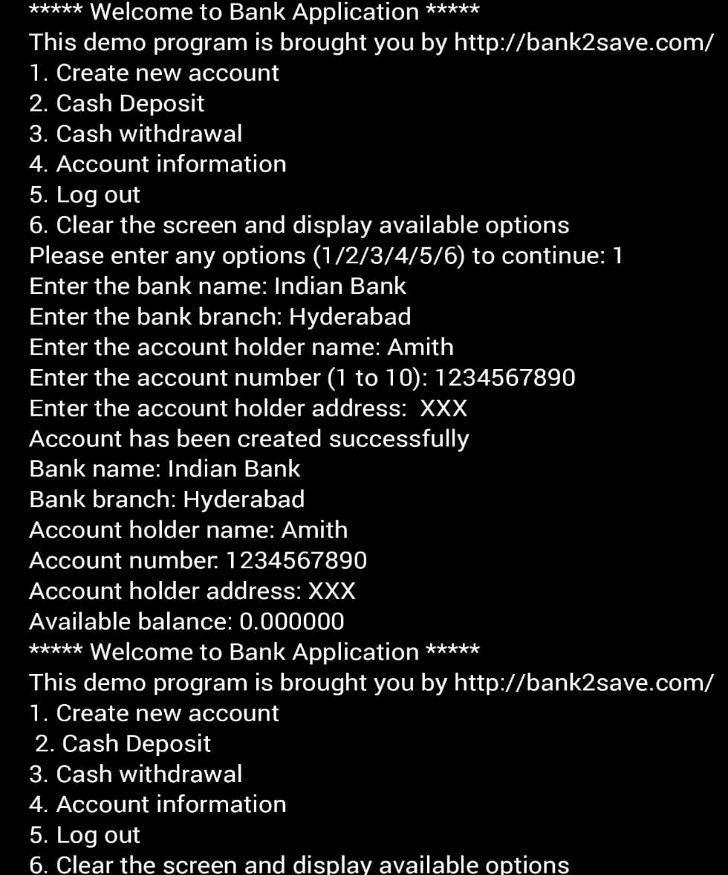
account[acc\_no-1].available\_balance-withdraw\_money;

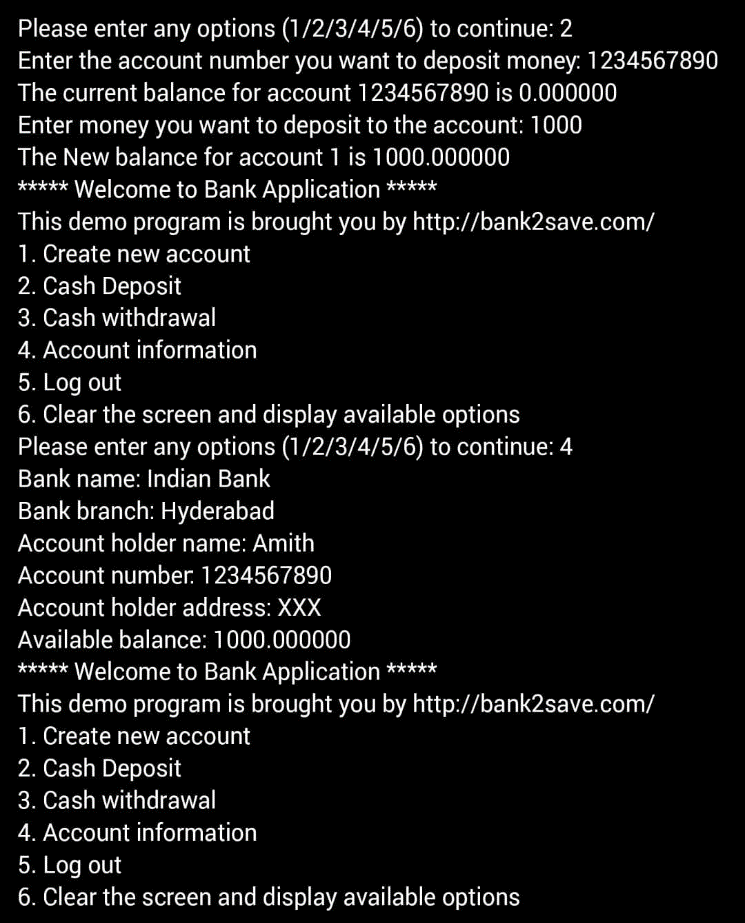
printf("\nThe New balance for account %d is %f \n",

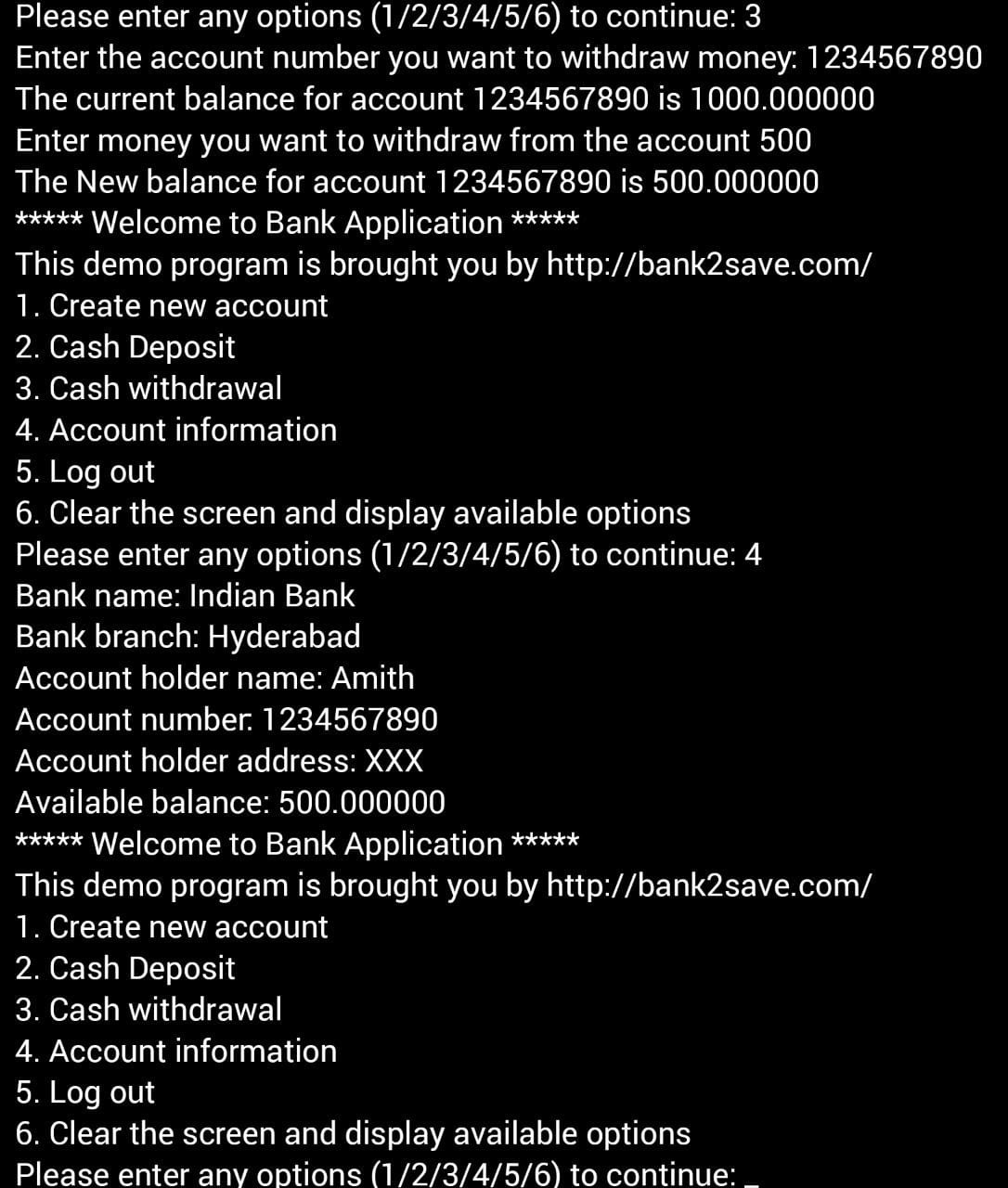
acc\_no, account[acc\_no-1].available\_balance);

break;

}acc\_no++;

**OUTPUT:**





**USES OF CODING IN BANKING**

The days of bankers yelling into telephones to facilitate deals are over, as these professionals pick up skills in programming to build and monitor algorithms that watch the banks' positions, create price quotes for clients, match buyers and sellers, and alert them to risks. However, this growing automation across the industry has not replaced traders, but changed their roles from pure banker to technologist, the reports said. Coding in banking has made customer more convenient and accessible. It has made sure that there is at least 95 % customer satisfaction. The use of technology in banking has created more jobs which has increased the employment rate .

Java is the most widely used programming language in major financial institutions.