**Code:**

#include<stdio.h>

#define N 6

struct account{

long int accno;

char name[20];

char type;

float bal;

}custmer[6]={{101,"Swapnil.C",'S',19999},

{102,"Swapnil.G",'S',12000},

{103,"Utkarsh",'S',14000},

{104,"Shreyas",'C',9000},

{105,"Shubham",'C',19000},

{106,"Vrushab",'S',6000}

}; //structure defined for account information

int accin; //index

float amount; //variable to hold ammount

void deposit() //function within function

{

printf("Enter the amount = ");

scanf("%f",&amount);

printf("\n");

custmer[accin].bal=custmer[accin].bal+amount;

balance();

} //function for deposit money

void withdraw() //function within function

{

printf("Enter the amount = ");

scanf("%f",&amount);

printf("\n");

if(amount>custmer[accin].bal)

{

printf("Amount excceds balance\n");

return;

}

custmer[accin].bal=custmer[accin].bal-amount;

balance();

} //function for withdraw money

void balance()

{

printf("Balance = %f\n",custmer[accin].bal);

}

// function to show balance

int main()

{

int accnum,i,ch,f=0;

start :

printf("Enter the Account number= ");

scanf("%d",&accnum);

for(i=0;i<N;i++)//check for existance of the account

{

if(accnum==custmer[i].accno)

{

accin=i;

f=1;

break;

}

}

if(f==0)

{

printf("\nInvalid Account number \n");

goto start;

}

printf("Account number= %d\nName : %s\nAccount type= %c\nBalance = %f\n",custmer[accin].accno,custmer[accin].name,custmer[accin].type,custmer[accin].bal);

while(1)

{

printf("\nEnter your choice\n1 : Balance Inquiry\n2 : Deposit Amount \n3 : Withdraw Amount\n4 : Exit\n");

scanf("%d",&ch);

switch(ch) //switch case for options selected wrt choice

{

case 1:balance();break;

case 2:deposit();break;

case 3:withdraw();break;

case 4:exit(0);break;

default:printf("Enter the correct choice");

}

}

return 0;

}

**Algorithm:**

Step 1: Start

Step 2: Declare and define structure “account” with variables “acc\_no”, “acc\_type”, “name[20]” and “balance”

Step 3: Declare variables “amt”, “arin”, “balance” and define array “customer []” with predefined data.

Step 4: Read account number from the user.

Step 5: Check if account number is valid. If not valid print “Invalid Account Number”

GOTO 8.

Step 6: Check if account number is valid. Print Account Number, Account Type and Name.

Step 7: Read choice from the user.  
 Case ‘1’ – call “balance” function which prints account balance.

Case ‘2’ – call “deposits” function which deposits funds to current balance.

Case ‘3’ – call “withdraw” function which withdraws funds from current balance

Case ‘default’ – GOTO 8

Step 8: STOP

**Output:**

