**PROJECT REPORT**

**On**

**CUSTOMER BILLING SYSTEM**

**C-PROGRAMMING LAB**

**Sub-Code:MCA-103**

**Lal Bahadur Shastri Institute of Management**

**Affiliated to Guru Gobind Singh Indraprastha University (GGSIPU)**

**MCA BATCH(2018-21)**

**SUPERVISOR: SUBMITTED BY:**

Dr. Lalit Kishore Arora Naveen Garg

Associate Professor MCA 1st Sem.

02414504418

**PROJECT CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **CONTENT** | **PAGE NO.** |
| 1. | Abstract |  |
| 2. | Acknowledgement |  |
| 3. | Introduction |  |
| 4. | Objective |  |
| 5. | System Modules |  |
| 6. | Modules Description  1.Add Customer  a)input()  b)writefile()  2.Search Customer Account  a)search()  b)output()  3.Exit |  |
| 7. | Source code and Output |  |
| 8. | Description about File  \*Data Record File |  |
| 9. | Conclusion |  |

**Abstract**

Customer Billing System is an application designed to generate the bill to the customer which can be used in any departmental store, shops, cafes etc. You can use this application to keep the records such as name, address, mobile number, paid amount, due amount, payment data etc. of your regular customer. Moreover, if you have a new customer, you can add .

**Tools to be used :**

1. GCC Compiler

2. VIM Tutor

3.GNU/GDB debugger

4.C/C++ interpreter

The source code of Customer Billing System Project in C has been written in C programming language without of the use of any graphics.

**Deliverables :**

1.You can hold any number of accounts and account can be added to the program at any time.

2.The programming of simple calculations such as calculation of due amount, balance etc. have been embed in the code of project.

3.The due amount to be paid is shown as negative balance.

4.If you have nothing to do with the program, you can directly exit from the main menu.

**Acknowledgement**

Any attempt at any level cannot be satisfactorily completed without the support and guidance of learned people. We would like to express our immense gratitude to **Dr. Lalit Kishore Arora** on computer programming for their constant support and motivation that has encouraged us to come up with this project.

We are also thankful to all our friends of MCA who have rendered their whole hearted supported directly or indirectly at all times for the successful completion of this project.

**Naveen Garg**

**(STUDENT)**

**Introduction**

Customer Billing System Project is a simple console application designed to demonstrate the practical use of C programming language and its features as wells as to generate an application which can be used in any departmental store, shops, cafes etc. for billing to the customer.

You can use this application to keep the records such as name, address, mobile number, paid amount, due amount, payment date etc. of your regular costumer. Moreover, if you have a new customer, you can add and edit the account at any time.

The source code of Customer Billing System Project in C has been written in C programming language without of the use of any graphics.

The project can be used in many aspects, firstly the application file generated can be used. Secondly the source code of Customer Billing System project in C can be used to learn C programming and its different features such as use of user defined functions, structures etc. Submitting this C project as your college/school C project without any modifications is completely discouraged.

**OBJECTIVE**

The main objectives of this project are listed as follows:

* Proper management of customer billing system.
* Add and maintain customer detail.
* Can be useful for both manager and cashier.
* Maximum providence of search options.
* User friendly messages in the way for easy application of program.

**SYSTEM MODULES**

We will use 3 modules in this project. These are as follows:

**Module 1: To add account**

It can hold any number of accounts and account can be added to the program at any time.

The programming of simple calculations such as calculation of due amount, balance etc. have been embed in the code of project.

**Module 2: To search account**

The Customer Billing System project in C gives you the facility of searching the account by two ways, either by name of the customer or by the number of customer.

The due amount to be paid is shown as negative balance.

**Module 3:To exit**

If you have nothing to do with the program, you can directly exit from the main menu.

**Modules Description**

**1.Add Customer**

**Input()**

The function void input() is used to add the new customer account i.e. with the help of this functions the parameters such as name, address, phone number, amount paid etc. are asked and entered.

void input()

{

FILE \*fp=fopen("data","rb");

if(fp != NULL)

{

fseek (fp,0,SEEK\_END);

tl=ftell(fp);

if(tl > 0)

{

sl=sizeof(customer);

ts=tl/sl;

fseek(fp,(ts-1)\*sl,SEEK\_SET);

fread(&customer,sizeof(customer),1,fp);

printf("\ncustomer no:%d\n",++customer.number);

}

}

else

{

printf("\ncustomer no:%d\n", 1);

customer.number = 1;

}

fclose(fp);

printf("\n Account number:");

scanf("%d",&customer.acct\_no);

printf("\n Name:");

scanf("%s",customer.name);

printf("\n Mobile No.:");

scanf("%f",&customer.mobile\_no);

printf("\n Address:");

scanf("%s",customer.street);

printf("\n City:");

scanf("%s",customer.city);

printf("\n Previous balance:");

scanf("%f",&customer.oldbalance);

printf(" Current payment:");

scanf("%f",&customer.payment);

printf("\n Payment Month(mm):");

scanf("%d",&customer.lastpayment.month);

printf(" Day:");

scanf("%d",&customer.lastpayment.day);

printf(" Year:");

scanf("%d",&customer.lastpayment.year);

while(getchar() != '\n');

return;

}

**writefile()**

Another function void writefile() has been utilized to create a file on hard disc of computer for storing the information and data of a customer.

void writefile()

{

FILE \*fp;

fp=fopen("data","ab");

fwrite(&customer,sizeof(customer),1,fp);

fclose(fp);

return;

}

**2.Search Customer Account**

**search() and output()**

The function void search() has been used to look for previously stored accounts either by name or by number of the customer. The fourth and the last user defined function used in this Customer Billing System Project in C is void output() which has been defined to show the result as console output.

In Customer Billing System, structure has very beautifully used to group the data type in single unit. The date variables (day, month and year) have been grouped in the structures named date where as other variables such as name, number, street, paid amount etc. are grouped under another structure named account.

void search()

{

char ch,ch1;

char nam[100];

int n,i,m=1;

FILE \*fp;

fp=fopen("data","rb");

do{

printf("\nEnter your choice:");

scanf("%c",&ch);

}while(ch!='1' && ch!='2');

switch(ch)

{

case '1':

:

:

:

:

:

fclose(fp);

}

return;

}

void output()

{

printf("\n\n Customer no :%d\n",customer.number);

printf(" Name :%s\n",customer.name);

printf(" Mobile no :%.f\n",customer.mobile\_no);

printf(" Account number :%d\n",customer.acct\_no);

printf(" Street :%s\n",customer.street);

printf(" City :%s\n",customer.city);

printf(" Old balance :%.2f\n",customer.oldbalance);

printf(" Current payment:%.2f\n",customer.payment);

printf(" New balance :%.2f\n",customer.newbalance);

printf(" Payment date :%d/%d/%d\n\n",customer.lastpayment.month,customer.lastpayment.day,customer.lastpayment.year);

printf(" Account status :");

switch(customer.acct\_type)

{

case 'C':

printf("CURRENT\n\n");

break;

case 'O':

printf("OVERDUE\n\n");

break;

case 'D':

printf("DELINQUENT\n\n");

break;

default:

printf("ERROR\\n\n");

}

return;

}

**SOURCE CODE & OUTPUTs(with screen snapshot)**

***SOURCE CODE***

/\* Customer Billing System.

\*/

#include<stdio.h>

#include<stdlib.h>

void input();

void writefile();

void search();

void output();

struct date

{

int month;

int day;

int year;

};

struct account

{

int number;

char name[100];

int acct\_no;

float mobile\_no;

char street[100];

char city[100];

char acct\_type;

float oldbalance;

float newbalance;

float payment;

struct date lastpayment;

}customer;

int tl,sl,ts;

void main()

{

int i,n;

char ch;

printf("\n\n CUSTOMER BILLING SYSTEM:\n\n");

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

printf("\n 1: To Add Account on list\n");

printf(" 2: To Search Customer Account\n");

printf(" 3: Exit\n");

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

do{

// while(getchar() != '\n');

printf("\nSelect what do you want to do?\n ");

scanf("%c",&ch);

while(getchar() != '\n');

}while(ch<='0' || ch>'3');

switch(ch)

{

case '1':

printf("\nHow many customer accounts you have to add?\n");

scanf("%d",&n);

for(i=0;i<n;i++)

{

input();

if(customer.payment>0)

customer.acct\_type=(customer.payment<0.1\*customer.oldbalance)? 'O': 'D';

else

customer.acct\_type=(customer.oldbalance>0)?'D' : 'C';

customer.newbalance=customer.payment - customer.oldbalance;

writefile();

}

main();

case '2':

printf("Search by what?\n");

printf("\n1 --- Search by Customer\_Number\n");

printf("2 --- Search by Customer\_Name\n");

search();

scanf("%c",&ch);

main();

case '3':

printf("\nA PROJECT BY NAVEEN GARG");

exit(1);

}

}

void input()

{

FILE \*fp=fopen("data","rb");

if(fp != NULL)

{

fseek (fp,0,SEEK\_END);

tl=ftell(fp);

if(tl > 0)

{

sl=sizeof(customer);

ts=tl/sl;

fseek(fp,(ts-1)\*sl,SEEK\_SET);

fread(&customer,sizeof(customer),1,fp);

printf("\ncustomer no:%d\n",++customer.number);

}

}

else

{

printf("\ncustomer no:%d\n", 1);

customer.number = 1;

}

fclose(fp);

printf("\n Account number:");

scanf("%d",&customer.acct\_no);

printf("\n Name:");

scanf("%s",customer.name);

printf("\n Mobile No.:");

scanf("%f",&customer.mobile\_no);

printf("\n Address:");

scanf("%s",customer.street);

printf("\n City:");

scanf("%s",customer.city);

printf("\n Previous balance:");

scanf("%f",&customer.oldbalance);

printf(" Current payment:");

scanf("%f",&customer.payment);

printf("\n Payment Month(mm):");

scanf("%d",&customer.lastpayment.month);

printf(" Day:");

scanf("%d",&customer.lastpayment.day);

printf(" Year:");

scanf("%d",&customer.lastpayment.year);

while(getchar() != '\n');

return;

}

void writefile()

{

FILE \*fp;

fp=fopen("data","ab");

fwrite(&customer,sizeof(customer),1,fp);

fclose(fp);

return;

}

void search()

{

char ch,ch1;

char nam[100];

int n,i,m=1;

FILE \*fp;

fp=fopen("data","rb");

do{

printf("\nEnter your choice:");

scanf("%c",&ch);

}while(ch!='1' && ch!='2');

switch(ch)

{

case '1':

fseek(fp,0,SEEK\_END);

tl=ftell(fp);

sl=sizeof(customer);

ts=tl/sl;

do{

printf("\nEnter Customer\_Number:");

scanf("%d",&n);

if(n<=0 || n>ts)

printf("\nEntered Wrong Customer\_Number\n");

else{

fseek(fp,(n-1)\*sl,SEEK\_SET);

fread(&customer,sl,1,fp);

output();

}

while(getchar() != '\n');

printf("\n\nAgain Search another account?(y/n):");

scanf("%c",&ch);

}while(ch=='y');

fclose(fp);

break;

case '2':

fseek(fp,0,SEEK\_END);

tl=ftell(fp);

sl=sizeof(customer);

ts=tl/sl;

fseek(fp,(ts-1)\*sl,SEEK\_SET);

fread(&customer,sizeof(customer),1,fp);

n=customer.number;

do{

printf("\nEnter the Name for Searching Account\_Delail:");

scanf("%s",nam);

fseek(fp,0,SEEK\_SET);

for(i=1;i<=n;i++)

{

fread(&customer,sizeof(customer),1,fp);

if(strcmp(customer.name,nam)==0)

{

output();

m=0;

break;

}

}

if(m!=0)

printf("\n\nDoesn't exist\n");

while(getchar() != '\n');

printf("\nAgain Search for another\_account?(y/n):");

scanf("%c",&ch);

}while(ch =='y');

fclose(fp);

}

return;

}

void output()

{

printf("\n\n Customer no :%d\n",customer.number);

printf(" Name :%s\n",customer.name);

printf(" Mobile no :%.f\n",customer.mobile\_no);

printf(" Account number :%d\n",customer.acct\_no);

printf(" Street :%s\n",customer.street);

printf(" City :%s\n",customer.city);

printf(" Old balance :%.2f\n",customer.oldbalance);

printf(" Current payment:%.2f\n",customer.payment);

printf(" New balance :%.2f\n",customer.newbalance);

printf(" Payment date :%d/%d/%d\n\n",customer.lastpayment.month,customer.lastpayment.day,customer.lastpayment.year);

printf(" Account status :");

switch(customer.acct\_type)

{

case 'C':

printf("CURRENT\n\n");

break;

case 'O':

printf("OVERDUE\n\n");

break;

case 'D':

printf("DELINQUENT\n\n");

break;

default:

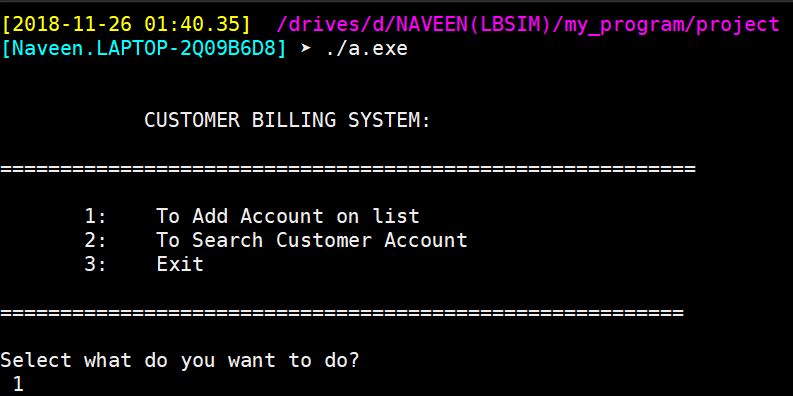
printf("ERROR\\n\n");

}

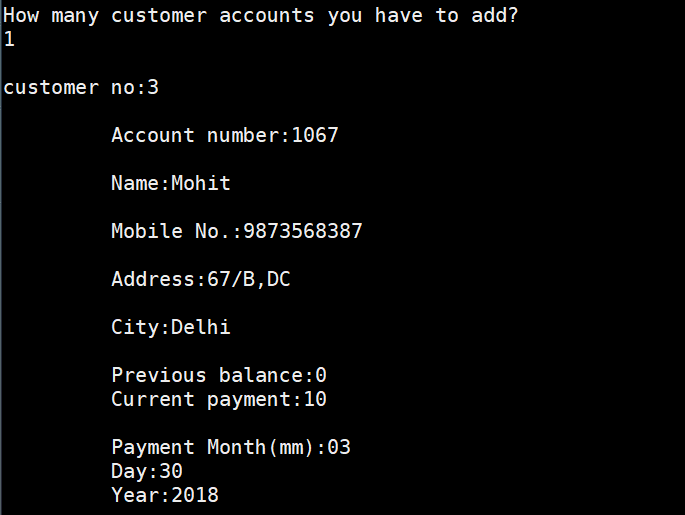
return;

}

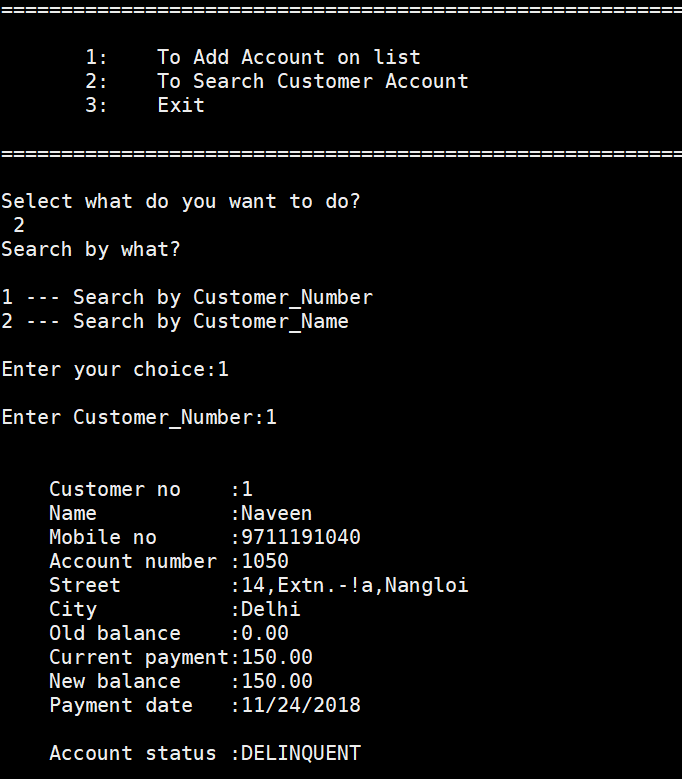
***OUTPUTS:-***



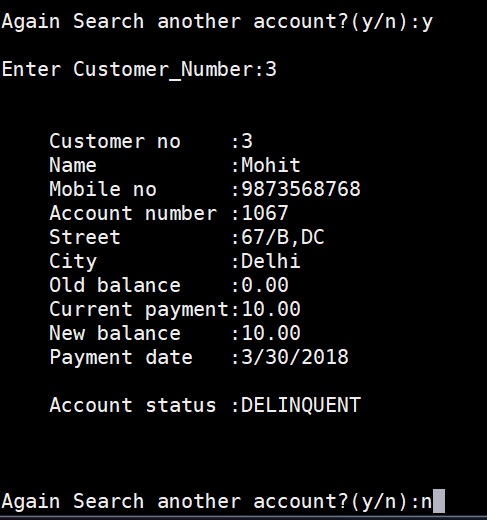
***Add the new customer:-***



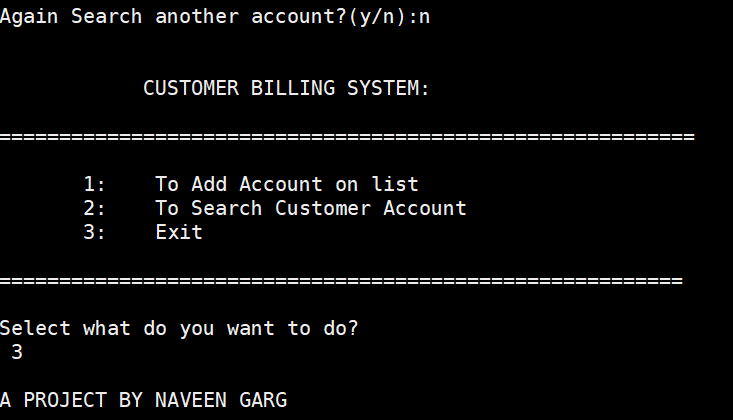
***Search the Customer Account***



***Again search another account***



***Exit the code***



**DESCRIPTION ABOUT FILE**

**Data Record file**

A binary field is an item of stored data. A field could be a name, a date, an address, a payment, etc. When a field is defined it is given a name (identifier) and a type. This is use to file pointer and mode and then open the Binary file .

Example:

FILE \*fp;

fp=fopen("data","ab");

fwrite(&customer,sizeof(customer),1,fp);

fclose(fp);

A record is the collection of fields that relate to a single entity. For example, we could have a customer record that includes fields for the customer’s name, address, mobile number, etc.

Within a file, all records usually have the same structure. That is, every record in the file contains the same fields.

**Conclusion**

Our project is on Customer Billing System. We have successfully completed it. We take this opportunity to express our sense of indebtedness and gratitude to all those people who helped us in completing this project.

We are immensely grateful to our esteemed faculty guide Assistant Prof. Dr. Lalit Kishore Arora for their supervision and guidance without which this work would not have been possible. This project has contributed a lot to my knowledge that has proved to be a value addition for me.