**SUPERMARKET BILLING SYSTEM**

SOURCE CODE:

#include<iostream>

#include<conio.h>

#include<string.h>

using namespace std;

class item

{

protected:

string pitems[30], npitems[30];

string pid[30], npid[30];

double pprice[30], npprice[30];

int pn, npn;

public:

void get\_i()

{

int i;

cout << "Enter no of perishable items\n";

cin >> pn;

cout << "Enter the items,id and price:\n";

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

{

cin >> pitems[i] >> pid[i] >> pprice[i];

}

cout << "\nEnter no of non perishable items\n";

cin >> npn;

cout << "Enter the items,id and price:\n";

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

{

cin >> npitems[i] >> npid[i] >> npprice[i];

}

sort\_i();

}

void sort\_i()

{

int i, j;

double a, b;

string pit, npit, pidd, npidd;

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

{

j = i - 1;

while (pprice[i] < pprice[j] && j >= 0)

{

a = pprice[i];

pprice[i] = pprice[j];

pprice[j] = a;

pit = pitems[i];

pitems[i] = pitems[j];

pitems[j] = pit;

pidd = pid[i];

pid[i] = pid[j];

pid[j] = pidd;

i--;

j--;

}

}

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

{

j = i - 1;

while (npprice[i] < npprice[j] && j >= 0)

{

a = npprice[i];

npprice[i] = npprice[j];

npprice[j] = a;

npit = npitems[i];

npitems[i] = npitems[j];

npitems[j] = npit;

npidd = npid[i];

npid[i] = npid[j];

npid[j] = npidd;

i--;

j--;

}

}

}

void delete\_i()

{

int n, i, j;

char x[30];

cout << "Enter 1 to delete a perishable item\n";

cout << "Enter 2 to delete a non-perishabe item\n";

cin >> n;

cout << "Enter the item to be deleted: ";

cin >> x;

switch (n)

{

case 1:

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

if (pitems[i] == x)

break;

if (i < pn)

{

pn = pn - 1;

for (j = i; j < pn; j++)

{

pitems[j] = pitems[j + 1];

pid[j] = pid[j + 1];

pprice[j] = pprice[j + 1];

}

}

cout << "\nItem " << x << " successfully deleted.\n";

break;

case 2:

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

if (npitems[i] == x)

break;

if (i < npn)

{

npn = npn - 1;

for (j = i; j < npn; j++)

{

npitems[j] = npitems[j + 1];

npid[j] = npid[j + 1];

npprice[j] = npprice[j + 1];

}

}

cout << "\nItem " << x << " successfully deleted.\n";

break;

}

}

void show\_i()

{

int i;

cout.width(25);

cout << "TYPE OF ITEM";

cout.width(15);

cout << "ITEMS";

cout.width(15);

cout << "ITEM\_ID";

cout.width(15);

cout << "PRICE" << endl;

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

{

cout.width(25);

cout << "Perishable item";

cout.width(15);

cout << pitems[i];

cout.width(15);

cout << pid[i];

cout.width(15);

cout << pprice[i] << endl;

}

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

{

cout.width(25);

cout << "Non-Perishable item";

cout.width(15);

cout << npitems[i];

cout.width(15);

cout << npid[i];

cout.width(15);

cout << npprice[i] << endl;

}

}

};

class perishable :virtual public item

{

protected:

double total\_pi = 0, dis\_pi = 0;

public:

void pi\_total()

{

int i;

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

total\_pi += pprice[i];

}

void pi\_discount()

{

pi\_total();

dis\_pi = 0.05 \* total\_pi;

}

};

class non\_perishable :virtual public item

{

protected:

double total\_npi = 0;

public:

void npi\_total()

{

int i;

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

total\_npi += npprice[i];

}

};

class customer

{

protected:

char name[30];

int contact\_no;

int membership\_no, m;

public:

void get\_c()

{

cout << "\nEnter name: ";

cin >> name;

cout << "Enter contact no: ";

cin >> contact\_no;

cout << endl;

}

void membership()

{

get\_c();

int choice;

cout << "\n\nEnter 1 if customer is a member\n";

cout << "Enter 2 if customer is not a member\n";

cin >> choice;

switch (choice)

{

case 1:

cout << "Enter membership no: ";

cin >> membership\_no;

m = 1;

break;

case 2:

membership\_no = 0;

m = 0;

break;

default:

cout << "Enter again\n";

}

}

void show\_c()

{

membership();

cout << "\n\n\n-------------------------------------------BILL------------------------------------------\n\n";

cout << "\nCustomer details:\n";

cout << "Name: " << name << endl;

cout << "Contact No.: " << contact\_no << endl;

cout << "Membership No.: " << membership\_no << endl << endl;

}

};

class bill : public customer, public perishable, public non\_perishable

{

protected:

double m\_discount, total\_bill;

public:

void i\_get()

{

item::get\_i();

}

void i\_delete()

{

item::delete\_i();

}

void total()

{

double dis, tdis;

pi\_discount();

npi\_total();

dis = total\_pi - dis\_pi;

total\_bill = dis + total\_npi;

customer::show\_c();

item::show\_i();

if (m == 1)

{

cout << "\nExtra 5% discount is given to memeber customer\n";

tdis = total\_bill \* 0.05;

m\_discount = total\_bill - tdis;

}

else

{

m\_discount = total\_bill;

tdis = 0;

}

cout << "\nTotal before discount= " << total\_pi + total\_npi << endl;

cout << "\nDiscounted amount= " << dis\_pi + tdis << endl;

cout << "\n\n\nFinal bill:" << m\_discount << endl << endl;

}

};

int main()

{

bill b;

int choice;

cout << "\n\n\t\t\t\t@@@@@-----WELCOME TO SV SUPERMARKET BILLING-----@@@@@\n\n";

cout << "\ MENU\n";

cout << " 1: To enter data\n";

cout << " 2: To delete data\n";

cout << " 3: To display bill\n";

cout << " 4: To exit\n";

do

{

cout << "\nEnter your choice\n";

cin >> choice;

switch (choice)

{

case 1:

cout << "\n\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n";

b.i\_get();

cout << "\n\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n";

break;

case 2:

cout << "\n\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n";

b.i\_delete();

cout << "\n\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n";

break;

case 3:

cout << "\n\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n";

b.total();

cout << "\n\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n";

break;

default:

cout << "\n\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n";

cout << "\nEnter again\n";

cout << "\n\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*-\*\n";

break;

}

cout << endl;

} while (choice != 4);

\_getch();

return 0;

}