



Chettinad Vidyashram

(Affiliated to Central Board of Secondary Education, New Delhi)
(Chettinad House, R.A.Puram, Chennai – 600 028)

COMPUTER SCIENCE

Certified to be the Bonafide Record of work done by

_____ of Std XII Sec F2
in the Computer Science Lab of the CHETTINAD VIDYASHRAM,
CHENNAI, during the year 2016 – 2017.

Date:

Teacher-in-charge

REGISTER NO. _____

**Submitted for All India Senior Secondary Practical Examination in
Computer Science held on _____ at
Chettinad Vidyashram, Chennai – 600 028.**

Principal

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

I would like to express my sincere thanks to Meena Aunty,
Principal Mrs. S.Amudhalakshmi
for their encouragement and support to work on
this Project . I am grateful to my computer science
teacher R.Uma Mageswari and to the computer science
department for the constant guidance and support to
complete the project.

INDEX

1. OVERVIEW OF C++ AND OOP CONCEPTS.....	
2. PROJECT DESCRIPTION.....	
3. FUNCTIONS USED.....	
4. FILES USED.....	
5. SOURCE CODE.....	
6. SAMPLE OUTPUTS.....	
7. CONCLUSION.....	
8. BIBLIOGRAPHY.....	

OVERVIEW OF C++ AND OOP CONCEPTS

The C++ programming language was developed at AT&T Bell Laboratories in the early 1980s by Bjarne Stroustrup. Lacking for simulations and decided to extend the language by adding features from his favourite language, Simula 67. Simula 67 was one of the earliest object-oriented languages. The name C++ was coined by Rick Mascitti where “++” is the C increment operator. Ever since the birth, C++ evolved to cope with problems encountered by users, and through discussions at AT&T. The maturation of the C++ language was attested by the two events: formation of American National Standard Institute (ANSI) and the publication of the annotated C++ reference manual by Ellis. Object oriented technology is regarded as the ultimate paradigm for the modeling of information, be that data or logic. The C++ has by now shown to fulfill this goal. The object oriented programming paradigm is superset of object based programming. It offers all the features of object base programming and overcomes its limitation by implementing inheritance so that the real world relations among objects can be represented programmatically. OOP programming objects represents an entity that can store data and has its interface through functions.

The basic concepts of OOP are as follows :

1) Data abstraction:

Abstraction refers to the act of representing the essential features without including the background details or explanations. The way of allowing only selected access of components without including the background details to objects and members of other classes is referred as Data abstraction.

2) Encapsulation:

The wrapping up of data and operations /functions (that operate on the data) into a single unit (called class) is known as Encapsulation. It is also a way to implement data abstractions. Encapsulation hides the details of implementation of an object.

3) Modularity:

Modularity is the property of system that has been decomposed into a set of cohesive and loosely coupled modulus. The act of partitioning the programme into individual components is called modularity. It reduces the complexity of the programme and creates number of well defined, documented boundaries within the programme.

4) Inheritance:

Inheritance is the capability of one class of things to derive capabilities or properties of other classes. Inheritance is transitive in nature.

5) Polymorphism:

Polymorphism refers to ability for a data or message to be processed in more than one form. There are two types: function overloading and operator overloading. It is the property by which same message can be sent to object several different classes and each object can respond in a different way depending on its class.

FUNCTIONS USED

`void Productshome();`

Menu list to modify/delete/create or view stock of the products.

`void Salespage();`

Enables the manager to view the sales for the day.

`void Passwordpage();`

To accept and check the password before login.

`void Salesprint();`

To calculate and print the sales for the day.

`void pcreate();`

To create a new product.

`void pcreateauth(char c[]);`

`void pdelete();`

To delete a product.

`void pmodify();`

To modify the details of a product.

`void pstockupdate();`

To update the stock every time a transaction takes place.

`void pview();`

To view all the products.

void employeeopts();

Menu list of the options available for the employee.

void Billaccept();

To accept all the items in the bill along with the quantity.

void Billprint();

To print the bill.

void pdisplay();

To display a

void Managerhome();

void listcreate();

void employeehome();

Home page for the employee.

void creditspage();

Contains details of our project along with credits

void Billamount();

To calculate the total amount billed.

void bempn();

To accept the name of the employee during login.

void default(int,int);

Contains the code for the box in each page along with shop's name.

FILES USED

PRODUCTSLIST.dat

BILL.dat

SOURCE CODE

```
#include<dos.h>
#include<fstream.h>
#include<conio.h>
#include<process.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<ctype.h>

struct product
{
    char pcode[20],pname[30];
    int pstock;
    float pcost;
};

void Productshome();
void Salespage();
void Passwordpage();
```

```

void Salesprint();
void pcreate();
void pcreateauth(char c[]);
void pdelete();
void pmodify();
void pstockupdate();
void pview();
void employeeopts();
void Billaccept();
void Billprint();
void pdisplay();
void showpro();
void Managerhome();
void listcreate();
void employeehome();
void creditspage();
void Billamount();
void Passwordpage();
void bempn();
void default(int,int);

```

```

class LINES
{
    public :
        void LINE_HOR(int, int, int, char) ;
        void LINE_VER(int, int, int, char) ;

```

```

        void BOX(int,int,int,int,char) ;

};

void default(int a=71,int b=21)
{
    clrscr();
    LINES L;
    L.BOX(10,5,a,b,218);
    L.BOX(11,6,a-1,b-1,219);
    gotoxy(33,5);
    // insert graphics
    cout<<" NAMMA CANTEEN ";

}

void pview()
{
    fstream x;
    x.open("PRODUCTLIST.dat",ios::in|ios::binary);
    product p;
    clrscr();
    cout<<"\n\n\tITEMCODE\tITEM NAME\tSTOCK\tCOST\n";
    while(x.read((char*)&p,sizeof(p)))
    {
        cout<<"\t\t"<<p.pcode<<"\t\t"<<p.pname<<"\t\t"<<p.pstock<<"\t"<<
        p.pcost<<endl;
    }
}

```

```

        getch();
    }

void pdisplay()
{
    fstream x;
    x.open("PRODUCTLIST.DAT",ios::in|ios::binary);
    product p;
    cout<<"ITEM CODE\tProduct name\tCOST";

    while(x.read((char*)&p,sizeof(p)))
    {
        cout<<p.pcode<<"\t"<<p.pname<<"\t"<<p.pcost<<endl;
    }
    x.close();
}

```

```

//*****
// THIS FUNCTION DRAWS THE HORRIZONTAL LINE
//*****

```

```

void LINES :: LINE_HOR(int column1, int column2, int row, char c)
{

```

```

        for ( column1; column1<=column2; column1++ )
        {
            gotoxy(column1,row) ;
            cout <<c ;
        }
    }

//*****

// THIS FUNCTION DRAWS THE VERTICAL LINE

//*****

void LINES :: LINE_VER(int row1, int row2, int column, char c)
{
    for ( row1; row1<=row2; row1++ )
    {
        gotoxy(column,row1) ;
        cout <<c ;
    }
}

//*****

// THIS FUNCTION DRAWS THE BOX

//*****

```

```

void LINES :: BOX(int column1, int row1, int column2, int row2, char c)
{
    char ch=218 ;
    char c1, c2, c3, c4 ;
    char l1=196, l2=179 ;
    if (c == ch)
    {
        c1=218 ;
        c2=191 ;
        c3=192 ;
        c4=217 ;
        l1 = 196 ;
        l2 = 179 ;
    }
    else
    {
        c1=c ;
        c2=c ;
        c3=c ;
        c4=c ;
        l1 = c ;
        l2 = c ;
    }
    gotoxy(column1,row1) ;
    cout <<c1 ;
    gotoxy(column2,row1) ;

```

```

    cout <<c2 ;
    gotoxy(column1,row2) ;
    cout <<c3 ;
    gotoxy(column2,row2) ;
    cout <<c4 ;
    column1++ ;
    column2-- ;
    LINE_HOR(column1,column2,row1,l1) ;
    LINE_HOR(column1,column2,row2,l1) ;
    column1-- ;
    column2++ ;
    row1++ ;
    row2-- ;
    LINE_VER(row1,row2,column1,l2) ;
    LINE_VER(row1,row2,column2,l2) ;
}

void showpro(product p)
{
    cout<<p.pcode<<"\t"<<p.pname<<"\t"<<"\t"<<p.pcost<<endl;
}

int fault=0;

void main()
{

```



```

    default();
    gotoxy(15,11);
    cout<<"JUST EAT IT";
    gotoxy(15,13);
    cout<<"Not your everywhere Amma Canteen";
    gotoxy(15,14);
    cout<<"Normal prices apply :p ";
    gotoxy(15,20);
    cout<<" Press any key to continue (0 to exit)";
    char ch;
    ch=getch();
    if(ch=='0')
    exit(0);
    // insert graphics
    fault=0;
    Passwordpage();

}

```

```

char empname[20];
long float sales;

void Managerhome()
{
    default(77,22);

```

```

gotoxy(13,8);
cout<<"HELLO BOSS! ";
gotoxy(13,10);
cout<<"WHAT DO YOU WANT TO ACCESS - ";
gotoxy(13,11);
cout<<"1.PRODUCT";
gotoxy(13,12);
cout<<"Allows you to access various features of our extensive database";
gotoxy(13,13);
cout<<"2.SALES";
gotoxy(13,14);
cout<<"Lets you view today's sales"<<endl;
gotoxy(13,15);
cout<<"3.EXIT "<<endl;
int ch;
gotoxy(13,18);
cout<<"ENTER CHOICE HERE : ";
cin>>ch;
switch(ch)
{
    case 1:
        Productshome();
        break;
    case 2:
        Salespage();
        break;

```

```

        case 3:
            clrscr();
            cout<<"\n\n\t\t\tLOG OUT?";
            cout<<"\n\n\t\t\tYES\tNO\n";
            cout<<"\n\t\t";
            char ch[5];
            gets(ch);
            if(strcmpi(ch,"yes")==0)
                main();
            else
                Managerhome();
            break;
        default:
            gotoxy(13,19);
            cout<<"PLEASE ENTER VALID CHOICES BOSS...";
            sleep (5);
            Managerhome();
            break;
    }
}

void Salespage()
{
    default();
    gotoxy(15,12);
    Salesprint();
}

```

```
    Managerhome();  
}
```

```
struct billitem  
{  
    char pcode[20],pname[30];  
    float pcost;  
    int qty;  
};
```

```
void listcreate()  
{  
    fstream x;  
    x.open("PRODUCTLIST.dat",ios::out|ios::binary);  
    x.close();  
}
```

```
void pcreateauth(char c[])  
{  
    product pw;  
    cout<<"\n\nEnter product details\n";  
    strcpy(pw.pcode,c);  
    cout<<"\nProduct name : ";
```

```

    gets(pw.pname);
    cout<<"\nProduct stock : ";
    cin>>pw.pstock;
    cout<<"\nProduct cost : ";
    cin>>pw.pcost;
    fstream x;
    x.open("PRODUCTLIST.dat",ios::app|ios::binary);
    x.write((char*)&pw,sizeof(pw));
    x.close();
    cout<<"\nItem successfully created\n";
}

void pcreate()
{
    product pw,pr;
    int d=0;
    char ch='y';
    while(ch=='y' || ch=='Y')
    {
        clrscr();
        fstream x;
        x.open("PRODUCTLIST.dat",ios::in|ios::binary);
        cout<<"\t\tCREATE MODE\n\n\n"<<endl;
        d=0;
        x.seekg(0,ios::beg);
        cout<<"Enter to be added product code : ";
    }
}

```

```

        gets(pw.pcode);
        while (x.read((char*)&pr,sizeof(pr)))
        {
            if(strcmpi(pw.pcode,pr.pcode)==0)
                d=1;
        }
        x.close();
        if (d==0)
            pcreateauth(pw.pcode);
        else
            cout<<"\nItem code already in use...\n";
        sleep (2);
        cout<<"Do you want to add another product?(y/n) ";
        cin>>ch;
        sleep (2);
    }
    Productshome();

}

void pdelete()
{
    product p1,p2;
    int f=0;
    fstream x,y;
    char ch;

```

```

while(ch=='y'||ch=='Y')
{
    cout<<"\t\tDELETE MODE\n\n";
    x.open("PRODUCTLIST.dat",ios::in|ios::binary);
    y.open("temp.dat",ios::out|ios::binary);
    cout<<"Enter product code : ";
    gets(p2.pcode);
    while(x.read((char*)&p1,sizeof(p1)))
    {
        if(strcmpi(p1.pcode,p2.pcode)==0)
        {
            f=1;
            continue;
        }
        y.write((char*)&p1,sizeof(p1));
    }
    if(f==0)
        cout<<"Requested record doesnt exist... ";
    sleep (3);
    cout<<"Do you want to delete another product(y\n)";
    cin>>ch;
    remove("PRODUCTLIST.dat");
    rename("temp.dat","PRODUCTLIST.dat");
    x.close();
    y.close();
}

Productshome();

```

```
}
```

```
void pstockupdate()
{
    product pw,pr;
    int f=0;
    char ch='y';
    fstream x,y;
    while(ch=='y' || ch=='Y')
    {
        cout<<"\t\tSTOCK UPDATE\n\n";
        x.open("PRODUCTLIST.dat",ios::in|ios::binary);
        y.open("temp.dat",ios::out|ios::binary);
        cout<<"Enter pcode : ";
        cin>>pw.pcode;
        while(x.read((char*)&pr,sizeof(pr)))
        {
            if(strcmpi(pr.pcode,pw.pcode)==0)
            {
                cout<<"Enter updated stock : ";
                cin>>pr.pstock;
                y.write((char*)&pr,sizeof(pr));
                continue;
            }
        }
    }
}
```



```

        }
        y.write((char*)&pr,sizeof(pr));
    }
    if(f==0)
        cout<<"Requested product doesnt exist...";
    cout<<"Do u want to update the stock of another product? y/n \n";
    cin>>ch;
    remove("PRODUCTLIST.dat");
    rename("temp.dat","PRODUCTLIST.dat");
    x.close();
    y.close();
}
Productshome();
}

```

```
void Productshome()
```

```

{
    defaul(75,30);
    gotoxy(15,10);
    cout<<"PRODUCT OPTIONS";
    gotoxy(15,13);
    cout<<"1.CREATE PRODUCT";
    gotoxy(15,14);
    cout<<"Allows you to create a product on our database";
}

```

```

gotoxy(15,15);
cout<<"2.MODIFY PRODUCT";
gotoxy(15,16);
cout<<"Allows you to change product details";
gotoxy(15,17);
cout<<"3.DELETE PRODUCT";
gotoxy(15,18);
cout<<"Allows you to erase a product from our database";
gotoxy(15,19);
cout<<"4.UPDATE STOCK";
gotoxy(15,20);
cout<<"Allows you to update an existing product's stock";
gotoxy(15,21);
cout<<"5.VIEW PRODUCT";
gotoxy(15,22);
cout<<"Allows you to view our extensive database";
gotoxy(15,23);
cout<<"6.EXIT";
gotoxy(15,24);
cout<<"Enter your choice : ";
int ch;
cin>>ch;
if (ch==1)
{
    clrscr();
    cout<<"Directing you to CREATE MODE .";

```

```

        sleep (1);
        cout<<".";
        sleep (1);
        cout<<".";
        sleep(1);
        cout<<".";
        pcreate();
    }
    else if (ch==2)
    {
        clrscr();
        cout<<"Directing you to MODIFY MODE .";
        sleep (1);
        cout<<".";
        sleep (1);
        cout<<".";
        sleep(1);
        cout<<".";
        pmodify();
    }
    else if (ch==3)
    {
        clrscr();
        cout<<"Directing you to DELETE MODE .";
        sleep (1);
        cout<<".";

```

```

        sleep (1);
        cout<<".";
        sleep(1);
        cout<<".";
        pdelete();
    }
    else if (ch==4)
    {
        clrscr();
        cout<<"Directing you to STOCK UPDATE .";
        sleep (1);
        cout<<".";
        sleep (1);
        cout<<".";
        sleep(1);
        cout<<".";
        pstockupdate();
    }
    else if(ch==5)
    {
        clrscr();
        cout<<"Directing you to view the product .";
        sleep(1);
        cout<<".";
        sleep(1);
        cout<<".";
    }

```

```

        sleep(1);
        cout<<".";
        pview();
        Productshome();
    }
else if (ch==6)
{
    clrscr();
    cout<<"Directing you to HOMEPAGE .";
    sleep (1);
    cout<<".";
    sleep (1);
    cout<<".";
    sleep(1);
    cout<<".";
    Managerhome();
}
else if (ch==7)
{
    clrscr();
    cout<<"NERRUPU DA";
    sleep (5);
    Productshome();
}
else
{

```

```

        clrscr();
        cout<<"YOU HAD ONE JOB....";
        getch();
        Productshome();
    }

}

```

```

void employeehome()
{
    clrscr();
    cout<<"\n\tENTER NAME : ";
    //insert graphics
    bempn();
    clrscr();
    cout<<"\n\n\n\t\t\tWELCOME "<<empname;
    clrscr();
    cout<<"Directing you to EMPLOYEE HOME .";
    sleep (1);
    cout<<".";
    sleep (1);
    cout<<".";
    sleep(1);
    cout<<".";
    clrscr();
}

```

```

        employeopts();
    }

void employeopts()
{
    defaul();
    gotoxy(25,10);
    cout<<"MENU";
    gotoxy(15,13);
    cout<<"1.BILLING";
    gotoxy(15,14);
    cout<<"2.EXIT";
    gotoxy(15,17);
    cout<<"ENTER CHOICE : ";
    int ch;
    cin>>ch;
    switch(ch)
    {
        case 1 :
            Billaccept();
            break;
        case 2 :
            main();
            break;
        default :
            gotoxy(15,18);

```

```

        cout<<"INVALID CHOICE";
        getch();
        employeeopts();
    }
}

```

```

void Billaccept()
{
    fstream x,y;
    int f;
    billitem b;
    char ch='y',c='y',code[20];
    product p;
    y.open("BILL.dat",ios::out|ios::binary);
    while(ch=='y'||ch=='Y')
    {
        default();
        gotoxy(15,12);
        cout<<"Enter product code : ";
        cin>>code;
        f=0;
        x.open("PRODUCTLIST.dat",ios::in|ios::out|ios::binary);
        while(x.read((char*)&p,sizeof(p)))
        {
            if(strcmpi(p.pcode,code)==0)

```



```

{
    gotoxy(15,13);
    showpro(p);
    gotoxy(15,15);
    cout<<"Is this the correct product? y/n";
    cin>>c;
    if(c=='n'||c=='N')
        goto l1;
    else
    {
        f=1;
        gotoxy(15,16);
        cout<<"Enter quantity : ";
        cin>>b.qty;
        int size=sizeof(p);
        p.pstock-=b.qty;
        if (p.pstock<=10)
        {
            gotoxy(15,17);
            cout<<"WARNING : STOCKS RUNNING  
LOW... \n"<<endl;
        }
        x.seekg((-1)*size,ios::cur);
        x.write((char*)&p,sizeof(p));
        strcpy(b.pcode,p.pcode);
        strcpy(b.pname,p.pname);
    }
}

```

```

        b.pcost=p.pcost;
        y.write((char*)&b,sizeof(b));
    }

}

}

if(f==0)
{
    cout<<"Product does not exist\n";
}

11:
cout<<"\nDo you want to continue? y/n\n";
cin>>ch;
x.close();
}
y.close();
Billprint();
}

```

```

void creditspage()
{

```

```

    defaul(71,33);
    //graphics here
    gotoxy(15,11);

```

```
gotoxy(25,11);
cout<<"CREDITS";
gotoxy(15,12);
gotoxy(25,12);
cout<<"-----";
gotoxy(15,13);
gotoxy(25,13);
cout<<"-----";
gotoxy(15,14);
cout<<"Code developed and designed by ";
gotoxy(15,15);
cout<<"RS Balajee";
gotoxy(15,16);
cout<<"Code implemented by ";
gotoxy(15,17);
cout<<"Anirudh S";
gotoxy(15,18);
cout<<"Graphics and debugging by";
gotoxy(15,19);
cout<<"Joel Raja Singh";
gotoxy(15,20);
cout<<"Project guided by";
gotoxy(15,21);
cout<<"MRS UMA";
gotoxy(15,23);
cout<<"CHETTINAD VIDYASHRAM ";
```

```

gotoxy(15,24);
cout<<"YEAR 2016-2017 ";
gotoxy(15,25);
cout<<"CLASS XII F2 ";
gotoxy(15,27);
cout<<"THANK YOU ";
gotoxy(15,28);
getch();
clrscr();
cout<<"Go to system login ? y/n\n";
char ch;
cin>>ch;
if((ch=='y')||(ch=='Y'))
    Passwordpage();
else
{
    clrscr();
    cout<<"Exiting the program ";
    sleep(1);
    cout<<".";
    sleep(1);
    cout<<".";
    sleep(1);
    cout<<".";
    sleep(1);
    delay(10);
}

```

```
        exit(0);
    }
}
```

```
void mainhome()
{
    cout<<"\t\tXYZ Store";
    getch();
    Passwordpage();
}
```

```
void Passwordpage()
{
    defaul();
    char pass[20];
    int i=0;
    gotoxy(27,11);
    cout<<"ENTER PASSWORD : ";
    while(1)
    {
        pass[i]=getch();
        if(pass[i]==13)
            break;
        cout<<"*";
        i++;
    }
}
```

```

}
pass[i]='\0';
if(strcmp(pass,"CREDITS")==0)
creditspage();
else if(strcmp(pass,"FULLMARKS")==0)
Managerhome();
else if(strcmp(pass,"XYZ@123")==0)
employeehome();
else
{
    gotoxy(32,15);
    cout<<"Incorrect password";
    fault++;
    if(fault>=3)
    {
        gotoxy(27,16);
        cout<<"You have typed wrong password 3 times ";
        int t=30;
        gotoxy(27,17);
        cout<<"Try again after "<<t<<" seconds";
        //cout<<"or press 0 to exit";
        gotoxy(27,18);
        while(t>=0)
        {
            sleep(1);
            t--;

```

```

        gotoxy(43,17);
        cout<<t<<" ";
        gotoxy(27,18);
    }
    fault=0;
    Passwordpage();
}
else
{
    gotoxy(27,17);
    cout<<"Press any key to try again";
    gotoxy(35,18);
    cout<<"0 to exit\n";
    char ch;
    ch= getch();
    if(ch=='0')
        main();
    else
        Passwordpage();
}
}

void Billamount()
{
    fstream x;

```

```

x.open("BILL.dat",ios::in|ios::binary);

billitem b;

double tot=0;

while(x.read((char*)&b,sizeof(b)))

{

    tot+=b.qty*b.pcost;

}

cout<<"\n\n";

cout<<"\t\t\t"<<"TOTAL AMOUNT : "<<tot<<endl;

cout<<"\t\t\t"<<"DISCOUNT %   : ";

double disc;

cin>>disc;

double finaltot;

finaltot=tot-((disc/100)*tot);

cout<<"\t\t\t"<<"FINAL AMOUNT : "<<finaltot<<"\n\n\n";

cout<<"\t\t\t"<<"AMOUNT PAID   : ";

double cash;

cin>>cash;

cout<<"\t\t\t"<<"BALANCE       : "<<cash-finaltot;

sales+=finaltot;

cout<<"\n\n\t"<<"BILLED BY : ";

cout<<empname<<" \n ";

sleep (5);

cout<<"\n\n\t\t\tTHANK YOU DEAR CUSTOMER";

cout<<" \n\t\t\tHAVE A GOOD DAY \n";

sleep (5);

```



```

    cout<<"XYZ STORES, Ground floor, Chettinad Vidyashram, Chettinad
    House\n";
    cout<<"R A Puram, Chennai\n";
    cout<<"Phone : 94444 06942\n";
    getch();
    x.close();
    employeeopts();
}

```

```

void Billprint()
{
    fstream x;
    clrscr();
    x.open("BILL.dat",ios::in|ios::binary);
    billitem b;
    cout<<"\t\tNAMMA CANTEEN\n\n\n";
    cout<<"ITEM CODE\tNAME\tQTY\tUNIT COST\tAMOUNT\n";
    while(x.read((char*)&b,sizeof(b)))
    {
        cout<<b.pcode<<"\t"<<b.pname<<"\t"<<b.qty<<"\t"<<
        b.pcost<<"\t"<<b.qty*b.pcost<<"\n";
    }
    x.close();
    Billamount();
}

```

```

void Salesprint()
{
    //clrscr();
    cout<<"SALES TODAY";
    gotoxy(15,14);
    cout<<"INR : "<<sales;
    getch();
}

```

```

void bempn()
{
    gets(empname);
}

```

```

void pmodify()
{
    product pr,pw;
    char code[10];
    char ch='y';
    int f=0;
    int size=sizeof(pr);
    fstream x,y;
    while(ch=='y'||ch=='Y')
    {
        clrscr();

```

Computer Science Project

```

        if(f==0)
        {
            cout<<"\nRequested product not found in database....\n";
        }
        else
        {
            cout<<"\nProduct has been modified \n";
        }
        cout<<"Do you want to modify another product? (y/n)";
        cin>>ch;
        x.close();
    }
    Productshome();
}

```

PROJECT DESCRIPTION

Our Computer Science Project is an attempt to create an ideal, efficient and user friendly Store management and Billing system.

Keeping in mind the requirements and experience of newly joined staff members, the main aim of this project has been to create an easy-access management system for a Retail Store.

The project contains a manager login and an employee login :

1. The manager login offers many tasks such as creation and modification of the product database, and a daily sales window.
2. The employee login enables the user to create and print the bill, with options to re-check the items before being added to the bill and upon printing the bill, receiving cash and calculating the balance to be returned.

The Project has been modeled on the kind of database required for efficient handling a supermarket and a bill-printing service designed to provide all essential details of a product and facilitates easy transaction; with the effective usage of file concepts, classes and function concepts in C++.

CONCLUSION

Thus, we have put in our best efforts to create easy-access software for a Retail Store, with the use of file concepts.

This project, with some additions to the existing options, and more sophistication, can be applied and used in actual supermarkets, which deal with hundreds of customers each day.

We have, to the best of our ability, provided the maximum options for creation & modification of the product database, thus providing an easy platform for the store management & staff, using the basic concepts of OOP and C++. The project also aims to demonstrate the working of a typical retail store.

BIBLIOGRAPHY

1. Computer Science with C++
 - Sumita Aurora
2. Basics of file concepts and data structures
 - Kushagra.
3. File handling and inheritance concepts
 - Mythili
4. Classical class concepts
 - Jagdish
5. Databases and their scope
 - Dyan Chand
6. Support & guidance of our teacher : Mrs. Uma Mageswari