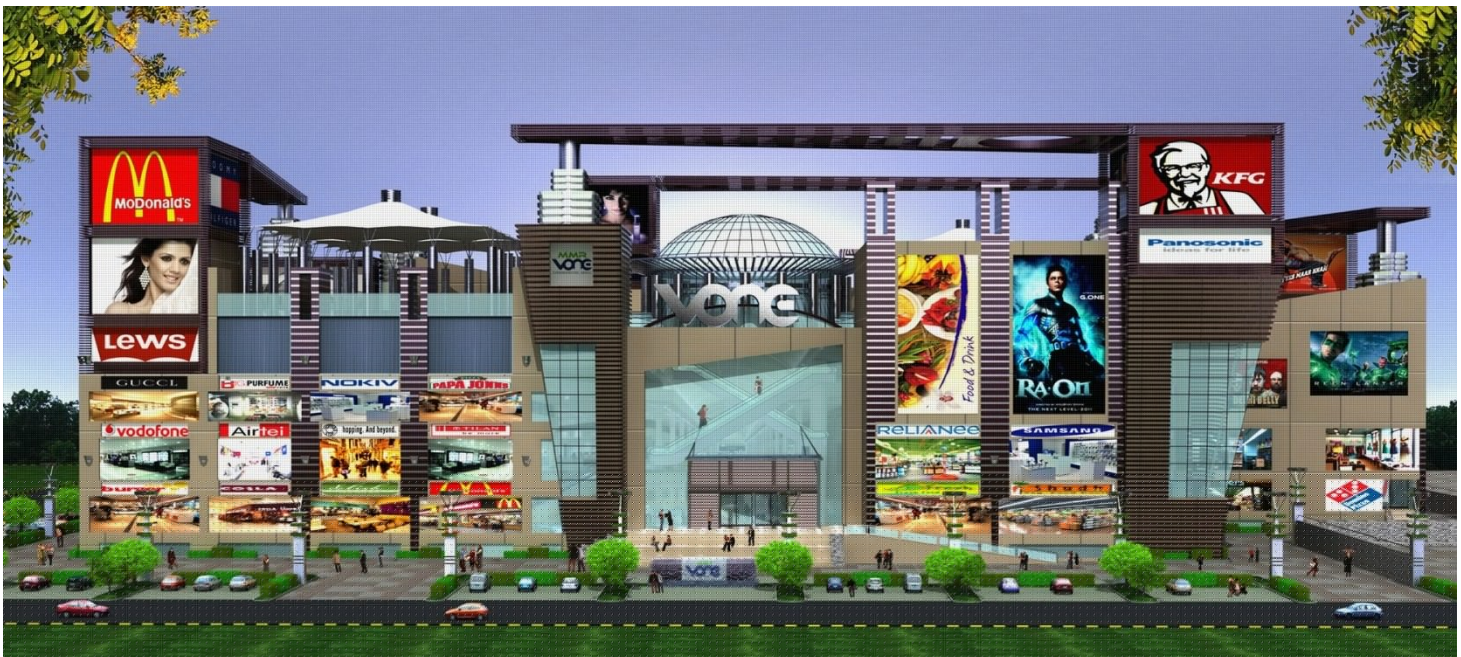


OXFORD PUBLIC SCHOOL

# Computer Science Project

Super Market Billing System



Made By- Tribhuwan Kumar

Class- XII 'F'

Board Roll no--

# Computer Science Project On

## Super Market Billing System

# Certificate

Certified to be the bonafide work done by Mr. Tribhuwan Kumar of class XII- F in the COMPUTER SCIENCE during the year 2014-2015 at “ Oxford Public School”.

This Project is absolutely genuine and does not indulge plagiarism of any kind.

The reference taken in making this project has been declared at the end of this project.

**Internal Signature**

**External Signature**

# Acknowledgement

I would like to express my sincere gratitude to my computer teacher Mrs. Rolley for her vital support, guidance and encouragement without which this project would not have completed.

I would also like to thank my parents and those who have supported me during the making of this project.

# INDEX

- ☺Introduction
- ☺System Requirement
- ☺About C++
- ☺History Of C++
- ☺Element Of C++
  - ☺ Objects
  - ☺ Encapsulation
  - ☺ Inheritance
  - ☺ Polymorphism
- ☺Objective of the proposed system
- ☺Advantage of the proposed system
- ☺Function Used in the program
- ☺Working
- ☺Source Code
- ☺Bibliography

# Introduction



The project is on Supermarket Billing. Supermarket is the place where customers come to purchase their daily using products and pay for that. So there is a need to calculate how many products are sold and to generate the bill for the customer.

**“Super Market billing system”** aims at developing into software that can be used at places like Shopping malls, Super Markets to easily maneuver the daily tasks of taking the order, calculating the bill etc.

The main advantage of this project is that it converts all the manual work which is time consuming and error prone to fully automated system which helps in



eliminating all the paper work, saves time, improves customer services. It also speeds up various processes such as addition of new items to the menu, deletion of items from the menu, modification of details of items and calculation of bills thus providing convenience to the workers as well as customers.

In the development of the project , selection of an appropriate programming language and a platform is of primary importance. The major part of the credit goes to the software environment chosen by the developer. Selection of a language from the ocean of languages is very difficult , a developer has to consider various features and functionalities that a particular language can provide.

In this project c++ language is used to maintain all the data. It provides many

features like file handling ,data can be easily maintained and many features that are required while doing a project.

# System Requirement

## ➤ **HARDWARE SPECIFICATION**

- Memory : 1 MB
- Microprocessor:1.2 Ghz
- Hard Disk: 40 GB
- Printer: Laser Printer

## ➤ **SOFTWARE SPECIFICATION**

- ✚ Platform: C++ With Graphic
- ✚ Front End: C++
- ✚ Window XP
- ✚ M.S word



# About C++

**C++** (pronounced *cee plus plus*) is a general purpose [programming language](#). It has [imperative](#), [object-oriented](#) and [generic](#) programming features, while also providing the facilities for low level memory manipulation.

It is designed with a bias for [systems programming](#) (e.g. embedded systems, operating system kernels), with performance, efficiency and flexibility of use as its design requirements. C++ has also been found useful in many other contexts, including [desktop applications](#), servers (e.g. [e-commerce](#), [web search](#), [SQL](#)), performance critical applications (e.g. [telephone switches](#), [space probes](#)) and entertainment software, such as [video games](#).

It is a [compiled](#) language, with implementations of it available on many platforms. Various organizations provide them, including the [FSF](#), [LLVM](#), [Microsoft](#) and [Intel](#).

C++ is standardized by the [International Organization for Standardization](#) (ISO), which the latest (and current) having being ratified and published by [ISO](#) in September 2011 as [ISO/IEC 14882:2011](#) (informally known as [C++11](#)). The C++ programming language was initially standardized in 1998 as [ISO/IEC 14882:1998](#), which was then amended by the C++03, [ISO/IEC 14882:2003](#), standard. The current standard ([C++11](#)) supersedes these, with [new features](#) and an enlarged [standard library](#).

Before standardization (1989 onwards), C++ was developed by [Bjarne Stroustrup](#) at [Bell Labs](#), starting in 1979, who wanted an efficient flexible language (like [C](#)) that also provided high level features for program organization.

Many other programming languages have been influenced by C++, including [C#](#), [Java](#), and newer versions of [C](#) (after 1998).

# History of C++

[Bjarne Stroustrup](#), a Danish and British trained computer scientist, began his work on C++'s predecessor "C with Classes" in 1979. The motivation for creating a new language originated from Stroustrup's experience in programming for his Ph.D. thesis. Stroustrup found that [Simula](#) had features that were very helpful for large software development, but the language was too slow for practical use, while [BCPL](#) was fast but too low-level to be suitable for large software development. When Stroustrup started working in [AT&T Bell Labs](#), he had the problem of analyzing the [UNIX kernel](#) with respect to [distributed computing](#). Remembering his Ph.D. experience, Stroustrup set out to enhance the [C](#) language with [Simula](#)-like features. C was chosen because it was general-purpose, fast, portable and widely used. As well as C and Simula's influences, other languages also influenced C++, including, [ALGOL 68](#), [Ada](#), [CLU](#) and [ML](#).

Initially, the class, derived class, [strong typing](#), [inlining](#), and [default argument](#) features were added to C via Stroustrup's "C with Classes" to C compiler, Cpre.

In 1983, it was renamed from *C with Classes* to C++ (++ being the [increment operator](#) in C). New features were added including [virtual functions](#), function name and [operator overloading](#), references, constants, type-safe free-store memory allocation (new/delete), improved type checking, and BCPL style single-line comments with two forward slashes (//), as well as the development of a proper compiler for C++, [Cfront](#).

In 1985, the first edition of [The C++ Programming Language](#) was released, which became the definitive reference for the language, as there was not yet an official standard. The first commercial implementation of C++ was released in October of the same year.

In 1989 C++ 2.0 was released followed by the updated second edition of *The C++ Programming Language* in 1991. New features in 2.0 included multiple inheritance, abstract classes, static member functions, [const member functions](#), and protected members. In 1990, *The Annotated C++ Reference Manual* was published. This work became the basis for the

future standard. Late feature additions included [templates](#), [exceptions](#), [namespaces](#), new [casts](#), and a [Boolean type](#).

In 2011, [C++11](#) was released which added more features and enlarged the standard library further (compared to it in 1998), providing more facilities for C++ programmers to use, with more additions [planned for 2014 and 2017](#).

# Element of C++

## Objects

C++ introduces [object-oriented programming](#) (OOP) features to C. It offers [classes](#), which provide the four features commonly present in OOP (and some non-OOP) languages: [abstraction](#), [encapsulation](#), [inheritance](#), and [polymorphism](#). One distinguishing feature of C++ classes compared to classes in other programming languages is support for deterministic [destructors](#), which in turn provide support for the [Resource Acquisition is Initialization](#) (RAII) concept.

## Encapsulation

[Encapsulation](#) is the hiding of information to ensure that data structures and operators are used as intended and to make the usage model more obvious to the developer. C++ provides the ability to define classes and functions as its primary encapsulation mechanisms. Within a class, members can be declared as either public, protected, or private to explicitly enforce encapsulation. A public member of the class is accessible to any function. A private member is accessible only to functions that are members of that class and to functions and classes explicitly granted access permission by the class ("friends"). A protected member is accessible to members of classes that inherit from the class in addition to the class itself and any friends.

The OO principle is that all of the functions (and only the functions) that access the internal representation of a type should be encapsulated within the type definition. C++ supports this (via member functions and friend functions), but does not enforce it: the programmer can declare parts or all of the representation of a type to be public, and is allowed to make public entities that are not part of the representation of the type.

Therefore, C++ supports not just OO programming, but other weaker decomposition paradigms, like [modular programming](#).

It is generally considered good practice to make all [data](#) private or protected, and to make public only those functions that are part of a minimal interface for users of the class. This can hide the details of data implementation, allowing the designer to later fundamentally change the implementation without changing the interface in any way.

## Inheritance

[Inheritance](#) allows one data type to acquire properties of other data types. Inheritance from a [base class](#) may be declared as public, protected, or private. This access specifier determines whether unrelated and derived classes can access the inherited public and protected members of the base class. Only public inheritance corresponds to what is usually meant by "inheritance". The other two forms are much less frequently used. If the access specifier is omitted, a "class" inherits privately, while a "struct" inherits publicly. Base classes may be declared as virtual; this is called [virtual inheritance](#). Virtual inheritance ensures that only one instance of a base class exists in the inheritance graph, avoiding some of the ambiguity problems of multiple inheritance.

[Multiple inheritance](#) is a C++ feature not found in most other languages, allowing a class to be derived from more than one base class; this allows for more elaborate inheritance relationships. For example, a "Flying Cat" class can inherit from both "Cat" and "Flying Mammal". Some other languages, such as [C#](#) or [Java](#), accomplish something similar (although more limited) by allowing inheritance of multiple [interfaces](#) while restricting the number of base classes to one (interfaces, unlike classes, provide only declarations of member functions, no implementation or member data). An interface as in C# and Java can be defined in C++ as a class containing only pure virtual functions, often known as an [abstract base class](#) or "ABC". The member functions of such an abstract base class are normally explicitly defined in the derived class, not inherited implicitly. C++ virtual inheritance exhibits an ambiguity resolution feature called [dominance](#).

## Polymorphism

[Polymorphism](#) enables one common interface for many implementations, and for objects to act differently under different circumstances.

C++ supports several kinds of *static* ([compile-time](#)) and *dynamic* ([run-time](#)) [polymorphisms](#). Compile-time polymorphism does not allow for

certain run-time decisions, while run-time polymorphism typically incurs a performance penalty

# OBJECTIVES OF THE PROPOSED SYSTEM

- To reduce time for the organization
- To increase efficiency and accuracy of the system
- To reduce pressure on the labour and relieving man power from repetitive and dull job
- To make the retrieval of information faster
- To make the system more feasible
- To reduce large amount of paper work
- To make the system more reliable to avoid any ambiguity.
- To reduce the cost factor of the system
- To make the system more flexible.

# ADVANTAGES OF THE PROPOSED SYSTEM

- Converts all the manual work which is time consuming and error prone to fully automated system
- Helps in eliminating all the paper work, saves time and improves customer services.
- Makes the addition of items in the menu, deletion of items and modification of items in the menu easier and faster.
- C++ has support for most of the web servers available today
- Bills can be calculated more easily and with more accuracy
- Reduces pressure on the labour.
- Makes the system more feasible and flexible and thus retrieval of information becomes convenient.





# Functions Used in this program

**Create\_Product( ) :-** This function is used to create new product , with name, price, discount

**Show\_product( ) :-** This function is used to see the product list, with description and price

**Write\_product():- This** function is used to write in file

**Display\_product():-** This function is used to display all record from file

**Display\_sp(..):-** Function to read specific record from file

**Modify\_product():-**Function to modify record of the file

**Delete\_product():-** Function to delete record of the file

**Menu():-**Display all product price list

**Place\_order():-** Function to place order and generating bill for product

**Intro():-** Introduction function

**Admin\_menu():-** Administration menu function

**Main():-** Main function of program

# Working

This project mainly consist of three menus

- 😊 Customer
- 😊 Administrator
- 😊 Exit

Customer Menu shows product list with Product no., name and price. This menu is used to place order. The steps involved are:-

1. Enter the product no. of the product from the list
2. Enter the quantity
3. Then place your order

Administration menu consist of the following options:

1. Create Product
2. Display all product
3. Query
4. Modify product
5. Delete product
6. View product menu
7. Back to menu

Enter choice

- When we choose the first option i.e. Create product, we need to mention product no, name, price, and discount of the product to create product.

- When we choose the second option i.e. Display all product, then all the details of the product is displayed one by one with product no., name, price and discount.
- If we choose the third option i.e. Query, then we have to enter the product no. to see the details of the respective product.
- 4<sup>th</sup> option i.e. Modify product is used to modify the product list, we have to mention new product no., name, price and discount.
- When we choose the 5<sup>th</sup> option i.e. Delete product, then we have to mention the product no, of those product which is to be deleted.
- If we choose the 7<sup>th</sup> option i.e. View Product menu, then product menu is displayed.

Exit menu is used to come out of the program.

# SOURCE CODE

/\*

@@  
@@

@ COMPUTER SCIENCE PROJECT WORK @

@@  
@@

\*

\*

\* SOFTWARE FOR BILLING SYSTEM \*

\*

\*

\*\*\*INTRODUCTION\*\*\*

~~~~~

~~~~~

NAME -> TRIBHUWAN KUMAR

CLASS -> XII SEC -> F

~~~~~

~~~~~

\*/

//\*\*\*\*\*

\*\*\*\*\*

//

HEADER FILE USED

//\*\*\*\*\*

\*\*\*\*\*

#include<conio.h>

#include<stdio.h>

#include<process.h>

#include<fstream.h>

//\*\*\*\*\*

\*\*

//

CLASS USED IN PROJECT

//\*\*\*\*\*

\*\*\*

class product

{

int pno;

char name[50];

float price,qty,tax,dis;

public:



```
void create_product()
{
    cout<<"\nPlease Enter The Product No. of The Product :";
    cin>>pno;
    cout<<"\n\nPlease Enter The Name of The Product :";
    gets(name);
    cout<<"\nPlease Enter The Price of The Product : ";
    cin>>price;
    cout<<"\nPlease Enter The Discount (%) : ";
    cin>>dis;
}
```

```
void show_product()
{
    cout<<"\nThe Product No. of The Product : "<<pno;
    cout<<"\nThe Name of The Product : ";
    puts(name);
    cout<<"\nThe Price of The Product : "<<price;
    cout<<"\nDiscount : "<<dis;
}
```

```
int retpno()
{return pno;}
```

```
float retprice()
```

```
{return price;}
```

```
char* retname()
```

```
{return name;}
```

```
int retdis()
```

```
{return dis;}
```

```
}; //class ends here
```

```
//*****  
**
```

```
// global declaration for stream object, object
```

```
//*****  
***
```

```
fstream fp;
```

```
product pr;
```

```
//*****  
**
```

```
// function to write in file
```

```
//*****  
***
```

```

void write_product()
{
    fp.open("Shop.dat",ios::out|ios::app);
    pr.create_product();
    fp.write((char*)&pr,sizeof(product));
    fp.close();
    cout<<"\n\nThe Product Has Been Created ";
    getch();
}

//*****
**

// function to read all records from file

//*****
***

void display_all()
{
    clrscr();
    cout<<"\n\n\n\t\tDISPLAY ALL RECORD !!!\n\n";
    fp.open("Shop.dat",ios::in);
    while(fp.read((char*)&pr,sizeof(product)))
    {
        pr.show_product();
    }
}

```

```
cout<<"\n\n=====
=====\\n";
```

```
getch();
```

```
}
```

```
fp.close();
```

```
getch();
```

```
}
```

```
/**
**
```

```
// function to read specific record from file
```

```
/**
***
```

```
void display_sp(int n)
```

```
{
```

```
int flag=0;
```

```
fp.open("Shop.dat",ios::in);
```

```
while(fp.read((char*)&pr,sizeof(product)))
```

```
{
```

```
if(pr.retpno()==n)
```

```
{
```

```
clrscr();
```

```
pr.show_product();
```

```

flag=1;
}
}
fp.close();
if(flag==0)
cout<<"\n\nrecord not exist";
getch();
}

//*****
**

// function to modify record of file

//*****
***

void modify_product()
{
int no,found=0;
clrscr();
cout<<"\n\n\tTo Modify ";
cout<<"\n\n\tPlease Enter The Product No. of The Product";
cin>>no;
fp.open("Shop.dat",ios::in|ios::out);
while(fp.read((char*)&pr,sizeof(product)) && found==0)
{

```

```

if(pr.retpno()==no)
{
pr.show_product();
cout<<"\nPlease Enter The New Details of Product:"<<endl;
pr.create_product();
int pos=-1*sizeof(pr);
fp.seekp(pos,ios::cur);
fp.write((char*)&pr,sizeof(product));
cout<<"\n\n\t Record Updated";
found=1;
}
}
fp.close();
if(found==0)
cout<<"\n\n Record Not Found ";
getch();
}

//*****
**

// function to delete record of file

//*****
***

void delete_product()

```

```

{
int no;
clrscr();
cout<<"\n\n\n\tDelete Record";
cout<<"\n\nPlease Enter The product no. of The Product You
Want To Delete";
cin>>no;
fp.open("Shop.dat",ios::in|ios::out);
fstream fp2;
fp2.open("Temp.dat",ios::out);
fp.seekg(0,ios::beg);
while(fp.read((char*)&pr,sizeof(product)))
{
if(pr.retpno()!=no)
{
fp2.write((char*)&pr,sizeof(product));
}
}
fp2.close();
fp.close();
remove("Shop.dat");
rename("Temp.dat","Shop.dat");
cout<<"\n\n\tRecord Deleted ..";
getch();
}

```



```
//*****  
**
```

```
// function to display all products price list
```

```
//*****  
***
```

```
void menu()
```

```
{
```

```
clrscr();
```

```
fp.open("Shop.dat",ios::in);
```

```
if(!fp)
```

```
{
```

```
cout<<"ERROR!!! FILE COULD NOT BE OPEN\n\n\n Go To  
Admin Menu to create File";
```

```
cout<<"\n\n\n Program is closing ....";
```

```
getch();
```

```
exit(0);
```

```
}
```

```
cout<<"\n\n\t\tPRODUCT MENU\n\n";
```

```
cout<<"=====  
=====\n";
```

```
cout<<"P.NO.\t\tNAME\t\tPRICE\n";
```

```
cout<<"=====  
=====\n";
```

```
while(fp.read((char*)&pr,sizeof(product)))
{
    cout<<pr.retpno()<<"\t\t"<<pr.retname()<<"\t\t"<<pr.retprice()<<endl;
}
fp.close();
}
```

```

//*****
**

```

```
// function to place order and generating bill for Products
```

```
//*****
***
```

```
void place_order()
{
    int order_arr[50],quan[50],c=0;
    float amt,damt,total=0;
    char ch='Y';
    menu();
    cout<<"\n\n\n\n\n\n";
    ======"";
    cout<<"\n          -->>PLACE YOUR ORDER<<--";
```

```

cout<<"\n
=====\\n";
do{
    cout<<"\n\nEnter The Product No. Of The Product : ";
    cin>>order_arr[c];
    cout<<"\nQuantity in number : ";
    cin>>quan[c];
    c++;
    cout<<"\nDo You Want To Order Another Product ? (y/n)";
    cin>>ch;
    }while(ch=='y' ||ch=='Y');
    cout<<"\n\nThank You For Placing The Order";getch();clrscr();

    cout<<"\n*****INVOICE*****\\n";

    cout<<"\nPr No.\tPr Name\tQuantity \tPrice \tAmount
\tAmount after discount\\n";
    for(int x=0;x<=c;x++)
    {
        fp.open("Shop.dat",ios::in);
        fp.read((char*)&pr,sizeof(product));
        while(!fp.eof())
        {
            if(pr.retpno()==order_arr[x])
            {
                amt=pr.retprice()*quan[x];

```

```
damt=amt-(amt*pr.retdis()/100);
```

```
cout<<"\n"<<order_arr[x]<<"\t"<<pr.retname()<<"\t"<<quan[x]<<"\t\t"<<pr.retprice()<<"\t"<<amt<<"\t\t"<<damt;
```

```
total+=damt;
```

```
}
```

```
fp.read((char*)&pr,sizeof(product));
```

```
}
```

```
fp.close();
```

```
}
```

```
cout<<"\n\n\t\t\t\t\tTOTAL = "<<total;
```

```
getch();
```

```
}
```

```
/**  
**
```

```
// INTRODUCTION FUNCTION
```

```
/**  
***
```

```
void intro()
```

```
{
```

```
clrscr();
```

```
gotoxy(25,3);
```

```

cout<<"----->> SUPER MARKET <<-----";
gotoxy(25,7);
cout<<"  ---->> BILLING <<----";
gotoxy(25,11);
cout<<"  --->> PROJECT <<---";
cout<<"\n\n\n\n\n\n\n\nMADE BY : Tribhuwan Kumar";
cout<<"\n\nSCHOOL : Oxford Public School";
getch();

}

```

```

//*****
**

```

```

// ADMINISTRATOR MENU FUNCTION

```

```

//*****
***

```

```

void admin_menu()
{
clrscr();
char ch2;
cout<<"\n\n\n\tADMINISTRATION MENU \n";
cout<<"\t_____";
cout<<"\n\n\t1:CREATE PRODUCT";
cout<<"\n\n\t2:DISPLAY ALL PRODUCTS";
cout<<"\n\n\t3:QUERY ";

```

```
cout<<"\n\n\t4:MODIFY PRODUCT";
cout<<"\n\n\t5:DELETE PRODUCT";
cout<<"\n\n\t6:VIEW PRODUCT MENU";
cout<<"\n\n\t7:BACK TO MAIN MENU";
cout<<"\n\n\tPlease Enter Your Choice (1-7) ";
ch2=getche();
switch(ch2)
{
case '1': clrscr();
write_product();
break;
case '2': display_all();break;
case '3':
int num;
clrscr();
cout<<"\n\n\tPlease Enter The Product No. ";
cin>>num;
display_sp(num);
break;
case '4': modify_product();break;
case '5': delete_product();break;
case '6': menu();
getch();
case '7': break;
```

```
default:cout<<"\a";admin_menu();  
}  
}
```

```
//*****  
**
```

```
// THE MAIN FUNCTION OF PROGRAM
```

```
//*****  
***
```

```
void main()
```

```
{  
char ch;  
intro();  
do  
{  
clrscr();  
cout<<"\n-->> Welcome to Super Market Billing System  
Application <<-- \n";
```

```
cout<<"\t*****  
*****\n\n";
```

```
cout<<"\n\n\n\tMAIN MENU\n";
```

```
cout<<"\t_____";
```

```
cout<<"\n\n\t01: CUSTOMER";
```

```
cout<<"\n\n\t02: ADMINISTRATOR";
```



```

cout<<"\n\n\t03: EXIT";
cout<<"\n\n\tPlease Select Your Option ";
ch=getche();
switch(ch)
{
case '1': clrscr();
place_order();
getch();
break;
case '2': admin_menu();
break;
case '3':exit(0);
default :cout<<"\a";
}
}while(ch!='3');
}

//*****
**

// END OF PROJECT

//*****
**

```

# Output

```
----->> SUPER MARKET <<-----
```

```
---->> BILLING <<----
```

```
--->> PROJECT <<---
```

MADE BY : Tribhuwan Kumar

SCHOOL : Oxford Public School

```
-->> Welcome to Super Market Billing System Application <<--  
*****
```

MAIN MENU

01: CUSTOMER

02: ADMINISTRATOR

03: EXIT

Please Select Your Option \_

## PRODUCT MENU

=====		
P.NO.	NAME	PRICE
=====		
1	Book	500
2	Pen	50
3	Pencil	10
4	Soap	50
5	Watch	5000

=====

-->>PLACE YOUR ORDER<<--

=====

Enter The Product No. Of The Product :

=====

-->>PLACE YOUR ORDER<<--

=====

Enter The Product No. Of The Product : 1

Quantity in number : 5

Do You Want To Order Another Product ? (y/n)y

Enter The Product No. Of The Product : 2

Quantity in number : 4

Do You Want To Order Another Product ? (y/n)n

Thank You For Placing The Order\_

\*\*\*\*\*INVOICE\*\*\*\*\*

Pr No.	Pr Name	Quantity	Price	Amount	Amount after discount
1	Book	5	500	2500	2250
2	Pen	4	50	200	190
				TOTAL = 2440	

ADMINISTRATION MENU

---

1:CREATE PRODUCT

2:DISPLAY ALL PRODUCTS

3:QUERY

4:MODIFY PRODUCT

5:DELETE PRODUCT

6:VIEW PRODUCT MENU

7:BACK TO MAIN MENU

Please Enter Your Choice (1-7) \_

Please Enter The Product No. of The Product :5

Please Enter The Name of The Product :Shoes

Please Enter The Price of The Product : 2000

Please Enter The Discount (%) : 5

The Product Has Been Created \_

#### PRODUCT MENU

=====		
P.NO.	NAME	PRICE
=====		
1	Book	500
2	Pen	50
3	Pencil	10
4	Soap	50
6	Watch	5000
5	Shoes	2000

# Bibliography

- ✓ Wikipedia
- ✓ Google.com
- ✓ Computer Science with C++ Sumita arora
- ✓ Object Oriented Programing in C++ by Robert Lafore