**REPORT**

**Government Polytechnic, Pune-16**

**(An Autonomous Institute of Government of Maharashtra)**



**A**

**Micro Project Report On**

**“CAFE SHOP MANAGEMENT”**

## SUBMITTED BY:

1. **Sampada Sujay Kothari - 1906067**
2. **Mansi Sachin Kothale - 1906066**

## Under the Guidance of

Mrs. Lalita Korade Ma’am

**Government Polytechnic, Pune-16**

**(An Autonomous Institute of Government of Maharashtra)**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr no.** | **Title** | **Page no.** |
| **1** | **Abstract** | **3** |
| **2** | **Problem statement** | **4** |
| **3** | **Explanation** | **4** |
| **4** | **Features of OOP** | **5** |
| **5** | **Concepts used in the project** | **5** |
| **6** | **Source Code** | **7** |
| **7** | **Project output** | **19** |
| **8** | **Outcomes achieved** | **24** |
| **9** | **Conclusion** | **24** |
| **10** | **References** | **24** |

**Government Polytechnic, Pune-16**

**(An Autonomous Institute of Government of Maharashtra)**

**Abstract**

Object-oriented programming (OOP) is a programming paradigm based on the

concept of "objects", which can contain data and code: data in the form of fields

(often known as attributes or properties), and code, in the form of procedures

(often known as methods).

A feature of objects is that an object's own procedures can access and often

modify the data fields of itself (objects have a notion of this or self). In OOP,

computer programs are designed by making them out of objects that interact with

one another. OOP languages are diverse, but the most popular ones are class-

based, meaning that objects are instances of classes, which also determine their

types.

Many of the most widely used programming languages (such as C++, Java,

Python, etc.) are multi-paradigm and they support object-oriented programming

to a greater or lesser degree, typically in combination with imperative,

procedural programming.

**Government Polytechnic, Pune-16**

**(An Autonomous Institute of Government of Maharashtra)**

**Problem statement :**

Create a project on ‘Cafe Shop Management’ in which a customer can choose food items to order food, edit order list and finally ask for the bill so that his/her order can be completed by Cafe.

**Explanation :**

In project, first page is just introduction of Café and there starts the interface of user (customer) with application. Then he/she will select the choices for orders of food items and it will be added into the list of ordered food.

After ordering, when customer wishes to exit from order page, application will ask for confirm order or edit order. If he/she confirms the order, the email address is taken from customer and bill is generated and its information is stored in one file and sent to the email address of that particular customer. If he/she want to edit order, then he/she can delete or add more food items in the ordered list and finally the bill is generated and saved. This is the basic idea behind this application. For user friendliness of application, we have added some graphics.

So this application can be used by Cafe manager, so that instead of taking orders by writing, manager can take orders through this application and also bill and total amount is generated in the application itself. So don’t need to use paper for that, he will just send that bill to the email address of that customer.

**Government Polytechnic, Pune-16**

**(An Autonomous Institute of Government of Maharashtra)**

**Features of OOP**

In OOP, data is important than procedure.

Programs are divided into smaller parts known as ‘Objects’.

Data doesn’t move freely in the program. It is hidden and cannot be accessed by external functions. So there is security.

Objects communicate through functions with each other.

New data and functions can be added easily in the program. So modifying is efficient.

It follows bottom-up approach.

* All these features are important and they are implemented in our application. That’s why it is OOP based application. There are many languages that have these OOP features like C++, Java, Python, etc. We have studied OOP using C++ and developed this application based on that study using C++.

**OOP Concepts used in project :**

1. **Header files :**

**#include<iostream.h> :** The C++ header file declares a set of functions

for standard Input/Output. It also defines I/O stream objects such as cin,

cout, etc.

**#include<fstream.h> :** This data type represents the file stream generally,

and has the capabilities of both ofstream and ifstream which means it can

create files, write information to files, and read information from files.

**Government Polytechnic, Pune-16**

**(An Autonomous Institute of Government of Maharashtra)**

**#include<string.h> :**  It is used to perform various functionalities related to string manupulation like [strlen()](https://www.geeksforgeeks.org/strlen-function-in-c/), [strcmp()](https://www.geeksforgeeks.org/strcmp-in-c-cpp/), [strcpy()](https://www.geeksforgeeks.org/strcpy-in-c-cpp/), size(), etc.

**#include<conio.h> :** It is used to perform functionalities related to console like clear screen, hold screen, etc. Examples are getch(), clrscr().

**#include<graphics.h>** : It is used to implement various graphic functions to manipulate graphics on the console.

1. **Classes** : A class in C++ is the building block, that leads to Object-Oriented

programming. It is a user-defined data type, which holds its own data

members and member functions, which can be accessed and used by

creating an instance of that class. A class is like a blueprint for an object.

1. **Inheritance** : Inheritance is the capacity of a class to inherit the properties of another class.
2. **Functions** : Functions: A function is a group of statements that together performs a task.

Every C++ program has at least one function, which is main(), and all the

most trivial programs can define additional functions. Various forms of functions are there like inline functions, friend functions, member functions, etc.

1. **Switch Case:** Switch case statements are a substitute for long if statements

that compare a variable to several integral values.

1. **Loops:** A loop is used for executing a block of statements repeatedly until a

particular condition is satisfied. There are three main types of loops, while

loop, do-while loop, for loop.

1. **Operators:** An operator is a symbol that tells the compiler to perform

specific mathematical or logical manipulations. Types of operators are

arithmetic operator, relational operator, logical operator, bitwise operator

and assignment operator.

1. **Constructors:** A constructor in C++ is a special method that is automatically

called when an object of a class is created. To create a constructor, use the

same name as the class, followed by parentheses ( ).

**Government Polytechnic, Pune-16**

**(An Autonomous Institute of Government of Maharashtra)**

1. **Pointers:** A pointer is a variable whose value is the address of another

variable. Like any variable or constant, you must declare a pointer before

you can work with it. The variable that stores the address of another variable

is what in C++ is called a pointer.

1. **Linked list :** A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations.  a linked list consists of nodes where each node contains a data field and a reference(link) to the next node in the list.
2. **Constructor** : A constructor in C++ is a special method that is automatically

called when an object of a class is created. To create a constructor, use the

same name as the class, followed by parentheses ( ).

1. **File handling** : Files are used to store data in a storage device permanently.

File handling provides a mechanism to store the output of a program in a file

and to perform various operations on it. In C++ we have a set of file

handling methods. In C++, files are mainly dealt by using three classes fstream, ifstream, ofstream.

1. **Computer graphics** : In computer graphics, we can draw graphics shapes like eclipse, line, rectangle, etc. on console using various functions in graphics library.

**Source code :**

#include<first.h>

#include<combo.h>

#include<choice.h>

#include<iostream.h>

#include<conio.h>

#include<string.h>

#include<dos.h>

#include<stdlib.h>

#include<food.h>

#include<graphics.h>

#include<fstream.h>

class customer

{

protected:

char a[10];

char email[25];

long mobno;

public:

customer(){}

customer(char a[10],char email[15], long mobno)

{

strcpy(this->a,a);

strcpy(this->email,email);

this->mobno = mobno;

}

inline void input\_name()

{

cin>>a;

}

inline void dis()

{

cout<<a;

}

char\* getName()

{

return a;

}

customer acceptCust(char a[]);

void disCust(customer);

};

void customer::disCust(customer c1)

{

cout<<"\n\t\t----------Customer details--------------";

cout<<"\n\t\tCustomer name : "<<c1.a;

cout<<"\n\t\tCustomer email ID : "<<c1.email;

cout<<"\n\t\tCusomer mobile no : "<<c1.mobno;

cout<<"\n\t\t--------------------------------"<<endl;

}

customer customer::acceptCust(char a[])

{

clrscr();

cleardevice();

cout<<"\n\t\tEnter your email ID : ";

cin>>email;

cout<<"\n\t\tEnter your mobile no : ";

cin>>mobno;

customer c1(a,email,mobno);

return c1;

}

int i=0;

class node

{

public:

char fname[30];

float a;

node \*np;

}\*stnode,\*tmp;

class Bill:public customer

{

int billno;

float total;

float cgst,sgst;

float finaltotal;

public:

Bill():customer()

{

char a1[30],a2[30];

billno = 23421;

total = calculate();

cgst = total \* 0.07;

sgst = total \* 0.07;

finaltotal = total + cgst + sgst;

}

void displayBill(char\*);

float calculate();

};

float Bill::calculate()

{

node \*temp;

temp=stnode;

total = 0;

while(temp!=NULL)

{

total = total + temp->a;

temp=temp->np;

}

return total;

}

void crlist(char name[],float b)

{

node \*fnode;

i++;

if(i==1)

{

stnode=new(node);

stnode->a=b;

strcpy(stnode->fname,name);

stnode->np=NULL;

tmp=stnode;

}

else

{

fnode=new (node);

fnode->a=b;

strcpy(fnode->fname,name);

fnode->np=NULL;

tmp->np=fnode;

tmp=tmp->np;

}

}

void del()

{ int key;

cout<<"Enter the id of order you want to delete:";

cin>>key;

struct node \*prev, \*cur;

/\* Check if head node contains key \*/

while (stnode != NULL && key==1)

{

// Get reference of head node

prev = stnode;

// Adjust head node link

stnode = stnode->np;

// Delete prev since it contains reference to head node

free(prev);

// No need to delete further

return;

}

prev = NULL;

cur = stnode;

/\* For each node in the list \*/

for(i=1;cur != NULL;i++)

{

// Current node contains key

if (i == key)

{

// Adjust links for previous node

if (prev != NULL)

prev->np = cur->np;

// Delete current node

free(cur);

// No need to delete further

return;

}

prev = cur;

cur = cur->np;

}

}

void dlist()

{

node \*temp;

temp=stnode;

while(temp!=NULL)

{

cout<<"\n\t\t| "<<temp->fname<<":"<<" "<<temp->a<<" Rs. |" ;

temp=temp->np;

}

}

void Bill::displayBill(char a[10])

{

clrscr();

cleardevice();

customer c1 = customer::acceptCust(a);

cout<<"\n\t\t==========Order Bill============="<<endl;

cout<<"\n\t\t\tBillno : "<<billno;

customer::disCust(c1);

cout<<"\n\n\t\tList of ordered food"<<endl;

dlist();

cout<<"\n\t\t================================"<<endl;

cout<<"\n\t\tTotal : "<<total;

cout<<"\n\n\t\tCGST : "<<cgst;

cout<<"\t\tSGST : "<<sgst;

cout<<"\n\n\t\tFinal amount : "<<finaltotal<<endl;

{

fstream file;

file.open("Customer",ios::out);

file<<"\n\t\t==========Order Bill============="<<endl;

file<<"\n\t\t\tBillno : "<<billno;

file<<"Name :";

file<<"\n\n\t\tList of ordered food"<<endl;

file<<"\n\t\t================================"<<endl;

file<<"\n\t\tTotal : "<<total;

file<<"\n\n\t\tCGST : "<<cgst;

file<<"\t\tSGST : "<<sgst;

file<<"\n\n\t\tFinal amount : "<<finaltotal<<endl;

node \*temp;

temp=stnode;

while(temp!=NULL)

{

file<<"\n\t\t| "<<temp->fname<<":"<<" "<<temp->a<<" Rs. |" ;

temp=temp->np;

}

file.close();

}

}

void comboChoice();

void drinkChoice();

void burgerChoice();

void pizzaChoice();

void sandwichChoice();

void friesChoice();

void saladChoice();

void soupChoice();

void cakeChoice();

void iceChoice();

void editMenu();

void main()

{

dis();

int chmain;

clrscr();

customer c1;

delay(200);

cout<<"Welcome to Cafe coffee day !"<<endl;

delay(500);

cout<<"Please let us know your name"<<endl;

c1.input\_name();

cout<<"\n"<<"So ";

c1.dis();

cout<<", How may i help you";

do

{

cout<<"\n\t\t--------MAIN MENU--------";

cout<<"\n\t\t| 1 : Show me combos |\n\t\t| 2 : Let me choose meal |\n\t\t|";

if(i==0)

cout<<" 3 : Exit \t\t|";

else

cout<<" 3 : Show me my order(s)|";

cout<<"\n\t\t-------------------------";

cout<<"\nDear ";

c1.dis();

cout<<", please enter your choice : ";

cin>>chmain;

switch(chmain)

{

case 1:

comboChoice();

break;

case 2:

char chm;

do{

menu();

cout<<"\n\n\n\n\n\n\n\n\n\n\nSelect catagory (press < to go back): ";

cin>> chm;

switch(chm)

{

case '1':

clrscr();

cleardevice();

drinkChoice();

break;

case '2':

clrscr();

cleardevice();

sandwichChoice();

break;

case '3':

clrscr();

cleardevice();

soupChoice();

break;

case '4':

clrscr();

cleardevice();

cakeChoice();

break;

case '5':

clrscr();

cleardevice();

cleardevice();

iceChoice();

break;

case '6':

clrscr();

cleardevice();

burgerChoice();

break;

case '7':

clrscr();

cleardevice();

pizzaChoice();

break;

case '8':

clrscr();

cleardevice();

friesChoice();

break;

case '<':

clrscr();

cleardevice();

break;

default:

clrscr();

cout<<"\n\tInvalid choice.Try again..!";

}

}while(chm!='<');

break;

case 3:

int chbill;

cleardevice();

cout<<"\n\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

a:

cout<<"\n\t\tPlease tell us, do you want to confirm order?-"<<endl;

cout<<"\n\t\t Here is your list of orders.."<<endl;

dlist();

cout<<"\n\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

bill:

cout<<"\n\n\t\t| 1 : Confirm order and generate bill |";

cout<<"\n\t\t| 2 : Edit order |";

cout<<"\n\t\t| 3 : Cancel order |";

cout<<"\n\t\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_";

cout<<"\n\t\tEnter choice : ";

cin>>chbill;

switch(chbill)

{

case 1:

cleardevice();

Bill b;

b.displayBill(c1.getName());

break;

case 2:

del();

cleardevice();

goto a;

case 3:

break;

default:

cout<<"\n\t\tPlease enter valid choice..";

cleardevice();

goto bill;

}

break;

default:

cleardevice();

cout<<"\nInvalid choice...Please enter valid choice..!";

}

}while(chmain!=3);

cout<<"\n\t\t---------THANK YOU...VISIT AGIN......";

getch();

}

void burgerChoice()

{

char chbur;

disbur();

do

{

cout<<"\t\t\t\t\t";

cout<<"\t\tEnter here : ";

cin>>chbur;

switch(chbur)

{

case '1':

crlist("Cheese burger",40);

break;

case '2':

crlist("Double cheese burger",60);

break;

case '3':

crlist("Quarter pounder",70);

break;

case '4':

crlist("Cheese burger(Paneer)",80);

break;

case '<':

break;

default:

cout<<"\nInvalid choie...Try again.!";

}

}while(chbur!='<');

}

void soupChoice()

{

char chsoup;

soupdis();

while(chsoup!='<')

{

cout<<"\t\t\t\t\t";

cout<<"\t\tEnter here : ";

cin>>chsoup;

switch(chsoup)

{

case '1':

crlist("Tomato soup",20);

break;

case '2':

crlist("Chinese soup",30);

break;

case '3':

crlist("Knorr soup",35);

break;

case '<':

break;

default:

cout<<"\n\tInvalid choice..Try again.!";

}

}

}

void iceChoice()

{

char chice;

creamdis();

while(chice!='<')

{

cout<<"\t\t\t\t\t";

cout<<"\t\tEnter here : ";

cin>>chice;

switch(chice)

{

case '1':

crlist("Vanilla ice cream",20);

break;

case '2':

crlist("Chocolate ice cream",25);

break;

case '3':

crlist("Strawberry ice cream",25);

break;

case '4':

crlist("Pista ice cream",25);

break;

case '5':

crlist("Butterscotch ice cream",30);

break;

case '<':

break;

default:

cout<<"\n\tInvalid choice..Try again.!";

}

}

}

void saladChoice()

{

char chsalad;

saladdis();

while(chsalad!='<')

{

cout<<"\t\t\t\t\t";

cout<<"\t\tEnter here : ";

cin>>chsalad;

switch(chsalad)

{

case '1':

crlist("Salad for dietitians",30);

break;

case '2':

crlist("Chocolate salad",30);

break;

case '3':

crlist("Jam salad(vegies)",45);

break;

case '<':

break;

default:

cout<<"\n\tInvalid choice..Try again.!";

}

}

}

void cakeChoice()

{

char chcake;

cakedis();

while(chcake!='<')

{

cout<<"\t\t\t\t\t";

cout<<"\t\tEnter here : ";

cin>>chcake;

switch(chcake)

{

case '1':

crlist("Chocolate cake",110);

break;

case '2':

crlist("Black forest cake",130);

break;

case '3':

crlist("Strawberry cake",150);

case '4':

crlist("Kaju pista cake",200);

break;

case '<':

break;

default:

cout<<"\n\tInvalid choice..Try again.!";

}

}

}

void friesChoice()

{

char chfr;

friesdis();

while(chfr!='<')

{

cout<<"\t\t\t\t\t";

cout<<"\t\tEnter here : ";

cin>>chfr;

switch(chfr)

{

case '1':

crlist("French fries",35);

break;

case '2':

crlist("French fries(cheese)",50);

break;

case '3':

crlist("Spicy french fries",40);

break;

case '<':

break;

default:

cout<<"\n\tInvalid choice..Try again.!";

}

}

}

void comboChoice()

{

char chcombo;

combo();

while(chcombo!='<')

{

cout<<"\t\tEnter here : ";

cin>>chcombo;

switch(chcombo)

{

case '1':

crlist("Mac combo",200);

break;

case '2':

crlist("Burger combo",350);

break;

case '3':

crlist("Meal combo",250);

break;

case '4':

crlist("Cold combo",350);

break;

case '5':

crlist("Breakfast combo",400);

case '<':

break;

default:

cout<<"\n\tInvalid choice..!Try agin..";

}

}

cleardevice();

}

void drinkChoice()

{

char chdrink;

dridis();

while(chdrink!='<')

{

cout<<"\t\t\t\t\t";

cout<<"\t\tEnter here : ";

cin>>chdrink;

switch(chdrink)

{

case '1':

crlist("Hot coffee",35);

break;

case '2':

crlist("Cold coffee",50);

break;

case '3':

crlist("Cold coffee(crush)",70);

break;

case '4':

crlist("Coca cola",110);

break;

case '<':

break;

default:

cout<<"\n\tInvalid choice..!";

}

}

}

void pizzaChoice()

{

char chpiz;

pizzadis();

while(chpiz!='<')

{

cout<<"\t\t\t\t\t";

cout<<"\t\tEnter here : ";

cin>>chpiz;

switch(chpiz)

{

case '1':

crlist("Paneer pizza",75);

break;

case '2':

crlist("Vegie cheesie pizza",85);

break;

case '3':

crlist("Onion pizza",85);

break;

case '4':

crlist("Extra cheese pizza",100);

break;

case '<':

break;

default:

cout<<"\n\tInvalid choice..Try again..!";

}

}

}

void sandwichChoice()

{

char chsan;

sandwichdis();

while(chsan!='<')

{

cout<<"\t\t\t\t\t";

cout<<"\t\tEnter here : ";

cin>>chsan;

switch(chsan)

{

case '1':

crlist("Jam sandwich",30);

break;

case '2':

crlist("Chocolate sandwich",40);

break;

case '3':

crlist("Vegie cheesie sandwich",60);

break;

case '<':

break;

default:

cout<<"\n\tInvalid choice..Try again.!";

}

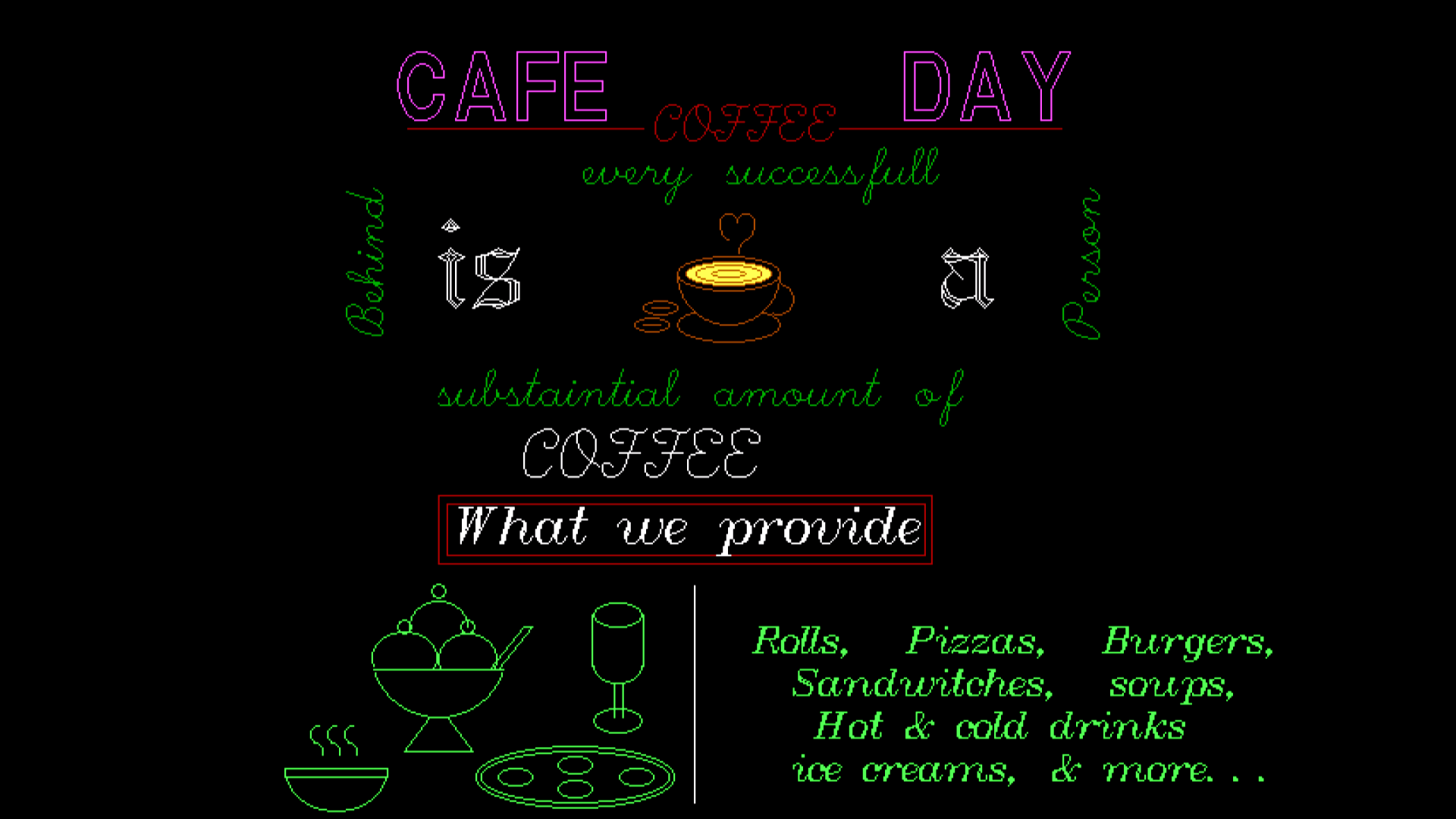
}

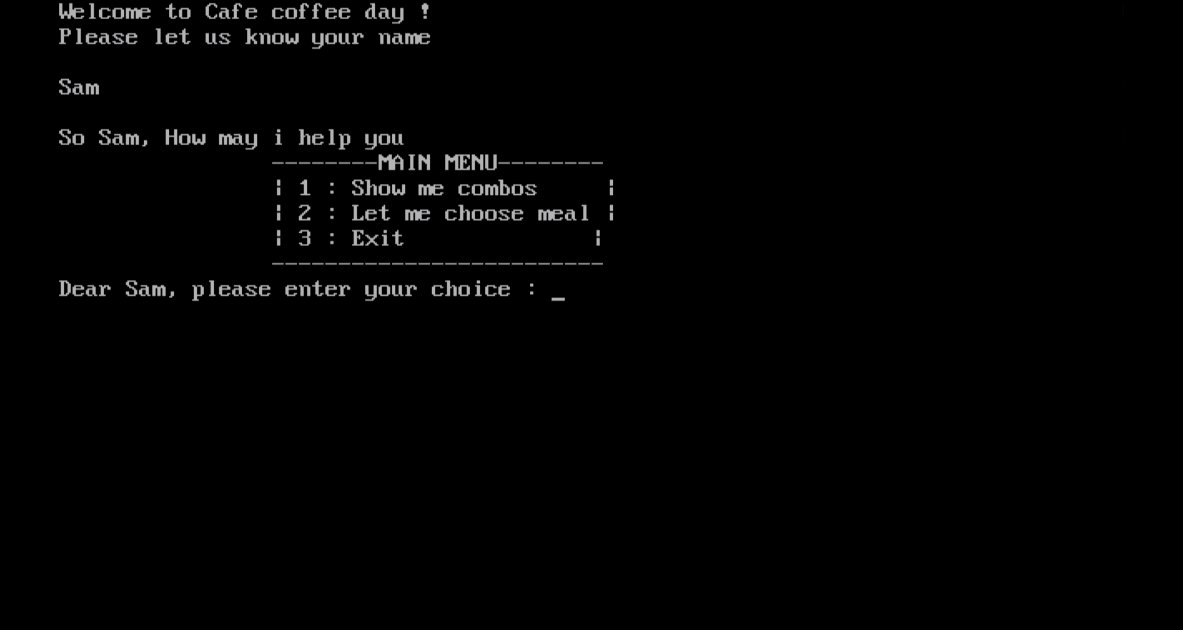
}

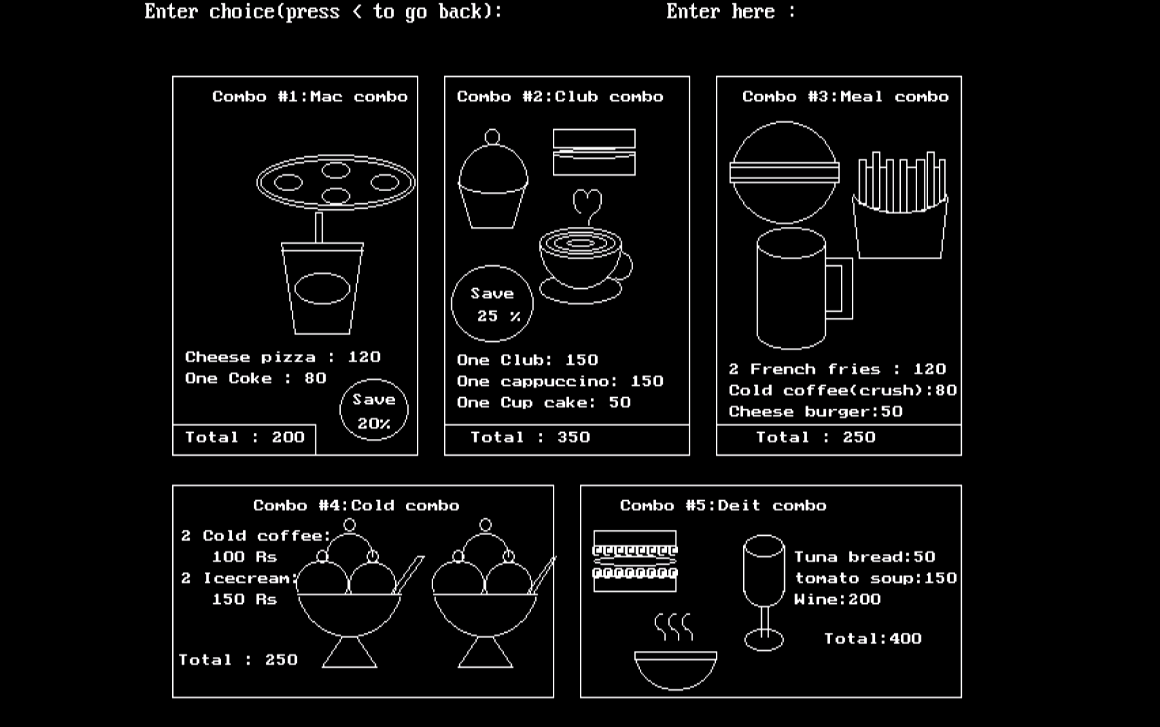
**Government Polytechnic, Pune-16**

**(An Autonomous Institute of Government of Maharashtra)**

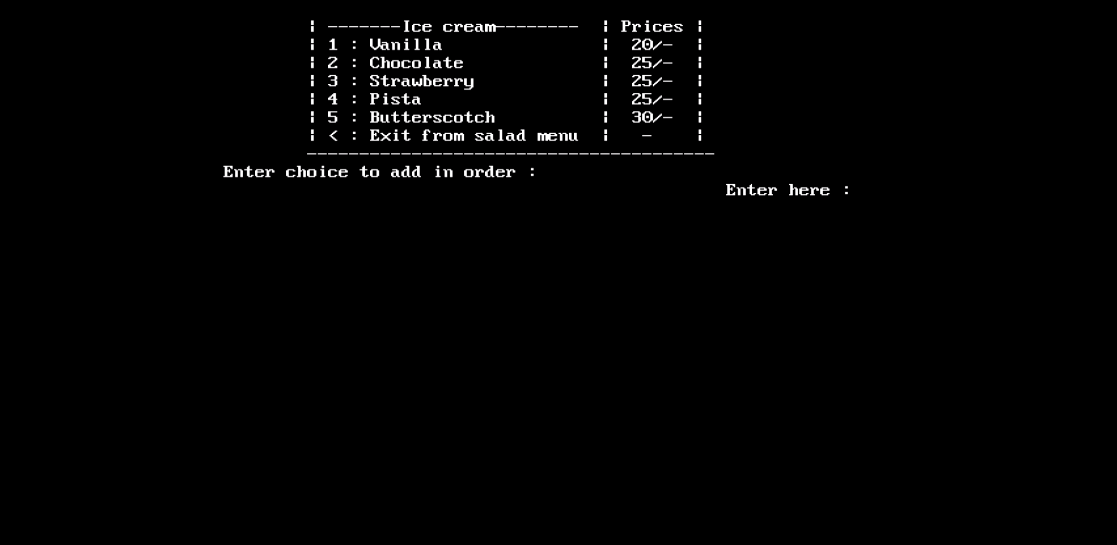
**Project Output :**



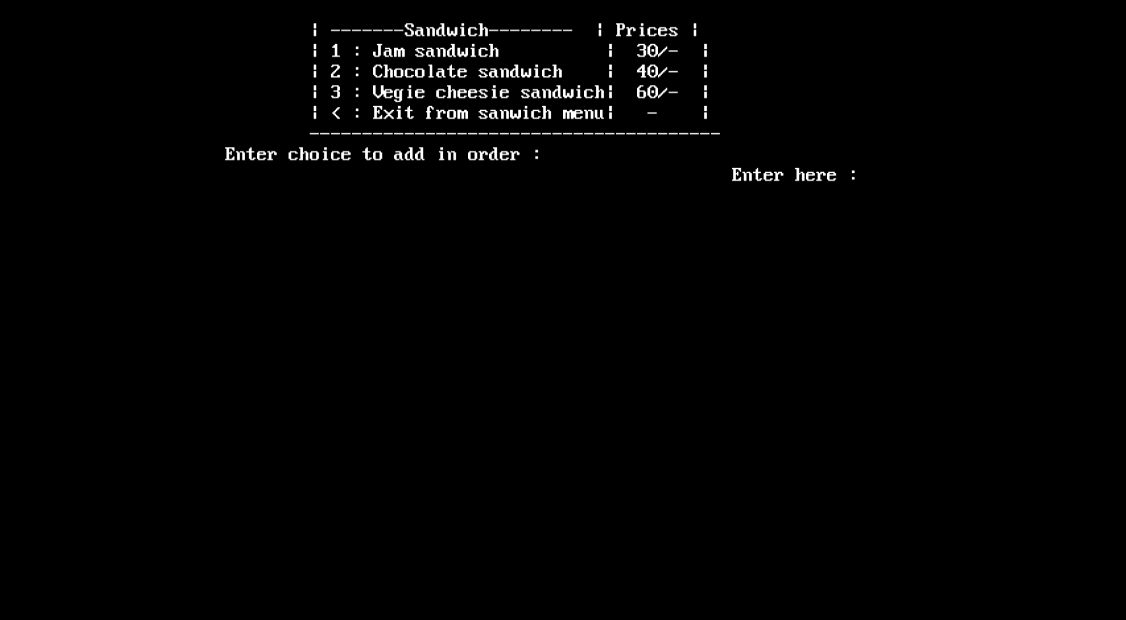






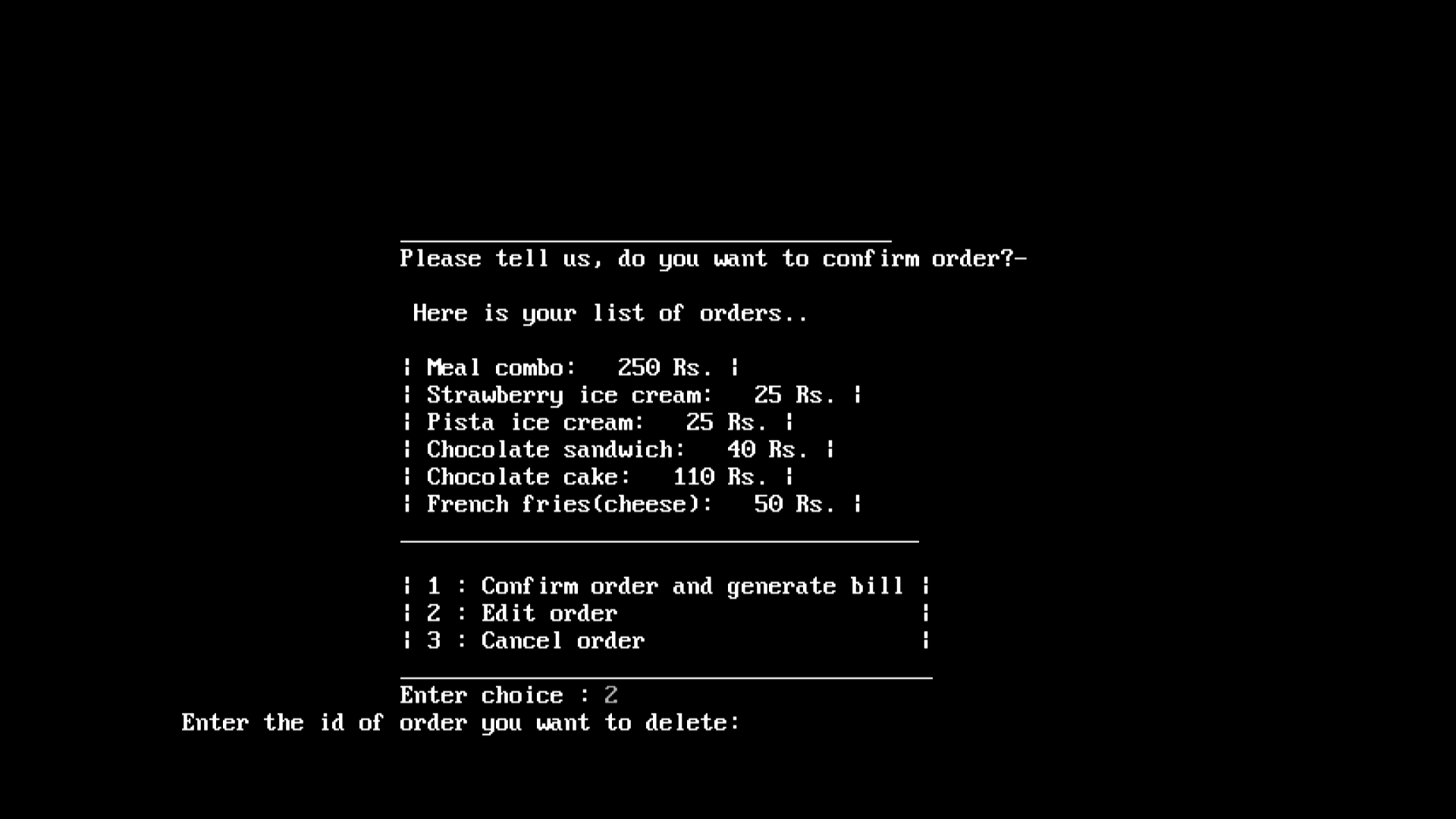


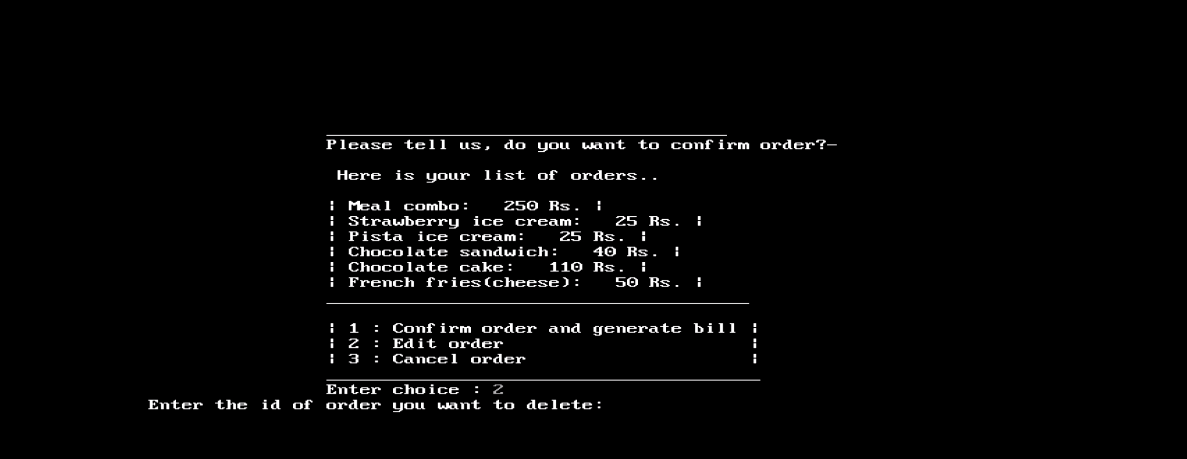




****

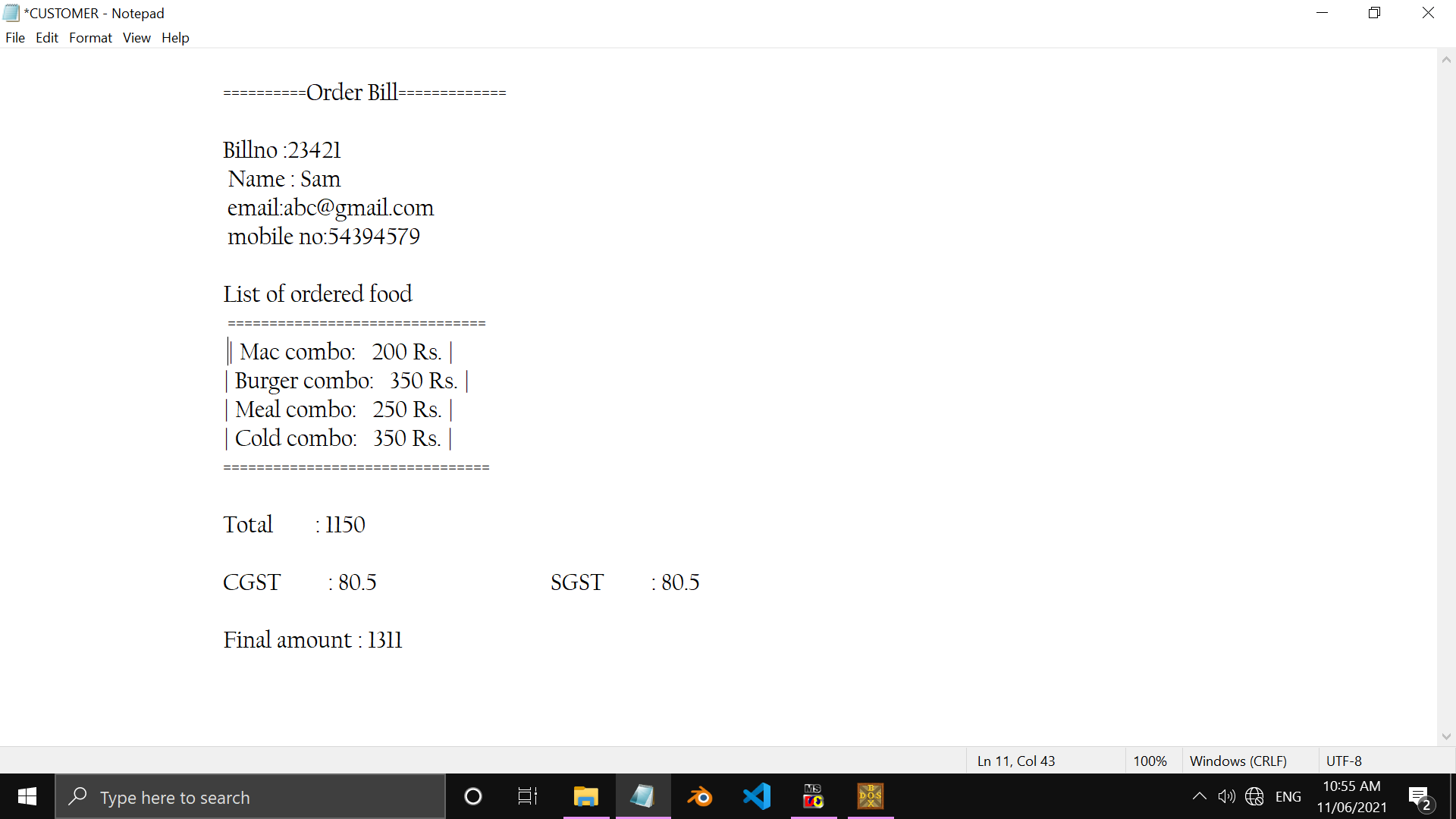
****

****

****

****

**Text file(File handling )**

****

**Government Polytechnic, Pune-16**

**(An Autonomous Institute of Government of Maharashtra)**

**Outcomes achieved :**

1. Enhanced our knowledge in C++ and also about OOP.
2. Improved our coding and skills by solving different compilation and logical errors.
3. Taught us how to develop a project in real time.
4. Learnt how to manage files in the C++ using file handling.
5. Developed the knowledge and concept of user friendliness applications by use of graphics.
6. Understood the necessity of team work and how to share ideas with each other and develop a proper micro project.

**Conclusion :**

Thus, from this project, we have studied the concepts of OOP using C++ in detailed manner and also we understood that how a micro project is developed with a team work. We used OOP concepts and also the graphics that we have studied in last term in the micro project “Cafe Shop Management using C++”.

**References** :

* **Guidance :** Mrs. Lalita Korade
* **Platform :** Turbo C++
* **Languages :** OOP using C++, Computer graphics
* **Book we used :** C++ by E. Balguruswamy.
* **Sites we used :** www.geeksforgeeks.org , www.tutorialspoint.com