# BAKERY MANAGEMENT SYSTEM


## GROUP MEMBERS:

MUNEEBA MEHMOOD (054)

AMNA ISMAEEL ABBASI (040)

AYESHA QAYUM (045)

NOOR UL AIN IMRAN (056)

**DEPARTMENT:** BSE-2B

**SUBMITTED TO:** DR. BUSHRA BASHIR


**SUBMISSION DATE:** 24 MAY 2021

# BAKERY MANAGEMENT SYSTEM

Bakery management system is a system that is especially build for providing the details and bills of bakery items like cakes, pastry and cookies on computer. This system helps the manager as well as customer to manage their ordered product easily. It consumes less time than writing a bill by hand and it makes it easy for the customer and manager to view the items with their rates and other basic information on computer.

**We used classes, objects, inheritance, polymorphism and files for update and delete record. Class diagram of this system is also mentioned.**

## PROCEDURE:

In this project, first of all we make the class named “Bakery” which stores the information of customer. It has a pure virtual function of “void cash” which is used for printing of bill. Due to the virtual function in this class, the class becomes an abstract class. By using the concept of inheritance, three derived classes are derived from the base class “Bakery ". The three classes are “cakes, cookies, pastry”.

In cake class, we have its additional functions i.e. “void show_cakes” which is used for displaying all the varieties of cakes and “void get_name” which is used for entering the user’s choice in order to get the item of that choice. The function of base class “Bakery” that stores the person’s information are again called by “this”, (pointer name), pointer in order to inherit the “get_name” function of base class" Bakery in function of “get_name” of the cake class. The virtual function of base class, void cash, is over-ride in this derived class to print the bill of cakes for the customer.

In cookies class, we have its additional functions i.e. “void show_cookies”; which is used for displaying all the varieties of cookies and “void get_name” which is used for entering the user’s choice in order to get the item of that choice, the function of base class that stores the person’s information are again called by “this” pointer in order to inherit the “get_name” function of base class in function of “get_name” of the cookies class. The virtual function of base class, void cash, is over-ride in this derived class to print the bill of cookies for the customer.

In pastry class, we have its additional functions i.e. “void show_pastry” which is used for displaying all the varieties of pastry and “void get_name” which is used for entering the user’s choice in order to get the item of that choice, the function of base class that stores the person’s information are again called by “this” pointer in order to inherit the “get_name” function of base class in function of “get_name” of the pastry class. The virtual function of base class, void cash, is over-ride in this derived class to print the bill of pastry for the customer.

In main function (), we make the objects of inherited classes that are cakes, cookies and pastry. And as we can’t make an object of base class because it is an abstract class. So, according to the concept of polymorphism, we declare three pointers of base class Bakery that are “*b1, *b2, *b3” to point the derived classes. By applying loop, inside the loop we put condition of if, else if and else. Under these conditions we specified the choice of user for the specific function and according to that choice the respected functions under the condition is called. “Void cash” function is called by the pointer of base class e.g. “b1->cash ()” for the functionality of respected class and remaining functions of inherited classes are called by the objects of that class using dot operator e.g. “P. show_pastry () “. At the end, all the functions are called and display all the required things.

## PROGRAM:

```
#include "stdafx.h"
#include<iostream>
#include"string"
#include<fstream>
using namespace std;

class bakery
{
protected:
string cust_name; //uppercase letters
public:
void get_name()
{
cout << "\n\nEnter your name(IN UPPERCASE): ";
cin >> cust_name;
cout << endl;
}
virtual void cash() = 0;
};

class cakes : public bakery
{

protected:
char cake_name;
```

```
int cake_size;
int bill;
char choice;
public:
       void setCakeInfo(string cakeName, string topping){  // will add the cake details
to csv file
    ofstream cakeInfo;
    cout<<".";
    cakeInfo.open("Bakery Cakes Details.csv",ios::app);
    cakeInfo<<cakeName<<","<<cake_size<<","<<topping<<","<<bill<<"\n";
    cakeInfo.close();
}

void show_cakes() //displaying cake details and prices
{

cout << "\n\nWE HAVE FOLLOWING VARIETY IN CAKES:\n";
cout << "Pineapple cakes (600 per pound)\nChocolate cakes (450 per pound)\nAlmond cakes
(300 per pound)\nCream cakes(450 per pound)\nCoffee cakes (500 per pound)\n\n\n";
cout << "Extra topping:\nFor Pineapple (300 Rs)\nFor Chocolate (200 Rs)\nFor Almond(150
Rs)\nFor Cream(250 Rs)\nFor Coffee(275 Rs)";
this->get_name();
}
void setName(){ // will add customer name to csv file
    ofstream cName;
    cName.open("Bakery Cakes Details.csv",ios::app);
    cName<<cust_name<<",";
    cName.close();
}
void get_name()
{
bakery::get_name(); //Enter cake name
cout << endl;
cout << "Enter cake name\n(P for Pineapple,C for Chocolate,A for Almond,R for Cream,O for
Coffee): ";
cin >> cake_name;
if (cake_name == 'P')
{
cout << "How many pounds do you want?: ";
cin >> cake_size;
cout << endl;
}
if (cake_name == 'C')
{
cout << "How many pounds do you want?: ";
cin >> cake_size;
cout << endl;
}
if (cake_name == 'A')
{
cout << "How many pounds do you want?: ";
cin >> cake_size;
cout << endl;
}
if (cake_name == 'R')
{
cout << "How many pounds do you want?: ";
cin >> cake_size;
```

```
cout << endl;
}
if (cake_name == 'O')
{
cout << "How many pounds do you want?: ";
cin >> cake_size;
cout << endl;
}
}
void cash()
{
cout << "------------YOUR FINAL BILL FOR CAKE IS------------" << endl;
cout << "RE-ENTER CAKE's CHOICE FOR BILLING.." << endl;
cin >> choice;
if (choice == 'P') //billing for pineapple cake
{
cout << "TOPINGS Y/N?";
cin >> choice;
if (choice == 'Y')
{
bill = (cake_size * 600) + 300;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCakeInfo("Pineapple","Yes"); // will add it in text file
}
else if (choice == 'N')
{
bill = cake_size * 600;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCakeInfo("Pineapple","No"); // will add it in text file
}
else
cout << "\n INVALID....ENTER AGAIN";
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
if (choice == 'C') //billing for chocolate cake
{
cout << "TOPINGS Y/N?";
cin >> choice;
if (choice == 'Y')
{
bill = (cake_size * 450) + 200;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCakeInfo("Chocolate","Yes");
}
else if (choice == 'N')
{
bill = cake_size * 450;
cout << "Your bill is: " << bill << endl;
 this->setName();
 this->setCakeInfo("Chocolate","No");
}
else
cout << "\n INVALID....ENTER AGAIN";
```

```
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
if (choice == 'A') //billing for almond cake
{
cout << "TOPINGS Y/N?";
if (choice == 'Y')
{
bill = (cake_size * 300) + 150;
cout << "Your bill is: " << bill << endl;
this->setName();
 this->setCakeInfo("Almond","Yes");
}
else if (choice == 'N')
{
bill = cake_size * 300;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCakeInfo("Almond","No");
}
else
cout << "\n INVALID....ENTER AGAIN";
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
if (choice == 'R') //billing for cream cake
{
cout << "TOPINGS Y/N?";
cin >> choice;
if (choice == 'Y')
{
bill = (cake_size * 450) + 250;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCakeInfo("Cream","Yes");
}
else if (choice == 'N')
{
bill = cake_size * 450;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCakeInfo("Cream","No");
}
else
cout << "\n INVALID....ENTER AGAIN";
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
if (choice == 'O') //billing for coffee cake
{
cout << "TOPINGS Y/N?";
cin >> choice;
if (choice == 'Y')
{
bill = (cake_size * 500) + 270;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCakeInfo("coffee","Yes");
```

```
}
else if (choice == 'N')
{
bill = cake_size * 500;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCakeInfo("coffee","No");
}
else
cout << "\n INVALID....ENTER AGAIN";
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
}
};

class cookies : public bakery
{

protected:
char cookie_name;
int quantity; //In kilograms
int bill;
char choice;
public:

       void setCookieInfo(string cookieName){  // will add the cookies details to csv
file
    ofstream cookieInfo;
    cout<<".";
    cookieInfo.open("Bakery Cookies Details.csv",ios::app);
    cookieInfo<<cookieName<<","<<quantity<<","<<bill<<"\n";
    cookieInfo.close();
}

void setName(){  // will add customer name to csv file
    ofstream cName;
    cName.open("Bakery Cookies Details.csv",ios::app);
    cName<<cust_name<<",";
    cName.close();
}

void show_cookies() //displaying cookie details and prices
{
cout << "\n\nWE HAVE FOLLOWING VARIETY IN COOKIES:\n";
cout << "Chocolate Chip Cookies (1600 per kg)\nOatmeal Raisin Cookies (1450 per
kg)\nGingerSnaps (1300 per kg)\nShort Bread Cookies (1550 per kg)\nPeanut Butter Cookies
(1500 per kg)\n\n";
this->get_name();
}
void get_name()
{
bakery::get_name(); //Enter cookie name
cout << endl;
cout << "Enter cookie name\n(C for Chocolate Chip Cookies,R for Oatmeal Raisin Cookies ,G
for GingerSnaps ,S for Short Bread Cookies ,B for Peanut Butter Cookies): ";
cin >> cookie_name;
if (cookie_name == 'C')
```

```
{
cout << "\n\nHow many kg do you want to buy?: ";
cin >> quantity;
}
if (cookie_name == 'R')
{
cout << "\n\nHow many kg do you want to buy?: ";
cin >> quantity;
}
if (cookie_name == 'G')
{
cout << "\n\nHow many kg do you want to buy?: ";
cin >> quantity;
}
if (cookie_name == 'S')
{
cout << "\n\nHow many kg do you want to buy?: ";
cin >> quantity;
}
if (cookie_name == 'B')
{
cout << "\n\nHow many kg do you want to buy?: ";
cin >> quantity;
}
}
void cash()
{
cout << "-------------YOUR FINAL BILL FOR COOKIES IS------------- " << endl;
cout << "RE-ENTER COOKIE NAME..." << endl;
cin >> cookie_name;
if (cookie_name == 'C') //billing for chocolatechip cookie
{
bill = 1600 * quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCookieInfo("Chocolate Chip");
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
else if (cookie_name == 'R') //billing for oatmeal cookie
{
bill = 1450 * quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCookieInfo("Oatmeal Raisin");
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
else if (cookie_name == 'G') //billing for gingersnaps cookie
{
bill = 1300 * quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCookieInfo("Ginger Snaps");
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
else if (cookie_name == 'S') //billing for short bread cookie
```

```
{
bill = 1550 * quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCookieInfo("Short Bread");
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
else if (cookie_name == 'B') //billing for peanut butter cookie
{
bill = 1500 * quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setCookieInfo("Peanut Butter");
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
else //invalid entry
cout << "\n INVALID....ENTER AGAIN";
}
};

class pastry : public bakery
{

protected:
char pastry_name;
int pas_quantity; //In dozen
int bill;
char choice;
public:
       void setPastryInfo(string pastryName){  // will add pastry details to csv file
    ofstream cookieInfo;
    cout<<".";
    cookieInfo.open("Bakery Pastry Details.csv",ios::app);
    cookieInfo<<pastryName<<","<<pas_quantity<<","<<bill<<"\n";
    cookieInfo.close();
}

void setName(){  // will add customer name to csv file
    ofstream cName;
    cName.open("Bakery Pastry Details.csv",ios::app);
    cName<<cust_name<<",";
    cName.close();
}

void show_pastry() //displaying pastry details and prices
{
cout << "\n\nWE HAVE FOLLOWING VARIETY IN PASTRY:\n";
cout << "ShortCrust Pastry (2000 per dozen)\nFilo Pastry (2450 per dozen)\nChoux Pastry
(3300 per dozen)\nFlaky Pastry (2550 per dozen)\nPuff Pastry (3000 per dozen)\n\n";
this->get_name();
}
void get_name()
{
bakery::get_name(); //Enter pastry name
cout << endl;
mkl:
```

```
cout << "Enter pastry name\n(S for ShortCrust Pastry ,F for Filo Pastry ,C for Choux
Pastry ,L for Flaky Pastry ,P for Puff Pastry): ";
cin >> pastry_name;
if (pastry_name == 'C')
{
cout << "\n\nHow many dozen do you want to buy?: ";
cin >> pas_quantity;
}
if (pastry_name == 'S')
{
cout << "\n\nHow many dozen do you want to buy?: ";
cin >> pas_quantity;
}
if (pastry_name == 'F')
{
cout << "\n\nHow many dozen do you want to buy?: ";
cin >> pas_quantity;
}
if (pastry_name == 'L')
{
cout << "\n\nHow many dozen do you want to buy?: ";
cin >> pas_quantity;
}
if (pastry_name == 'P')
{
cout << "\n\nHow many dozen do you want to buy?: ";
cin >> pas_quantity;
}
}
void cash()
{
cout << "-------------YOUR FINAL BILL FOR PASTRY IS-------------- " << endl;
cout << "RE-ENTER PASTRY NAME..." << endl;
cin >> pastry_name;
if (pastry_name == 'C') //billing for choux pastry
{
bill = 3300 * pas_quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setPastryInfo("Choux");
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
else if (pastry_name == 'S') //billing for shortcrust pastry
{
bill = 2000 * pas_quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setPastryInfo("Short Crust");
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING**********\nHAVE A
NICE DAY AHEAD :)";
}
else if (pastry_name == 'F') //billing for filo pastry
{
bill = 2450 * pas_quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setPastryInfo("Fillo");
```

```
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING*********\nHAVE A
NICE DAY AHEAD :)";
}
else if (pastry_name == 'L') //billing for flaky pastry
{
bill = 2550 * pas_quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setPastryInfo("Flaky");
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING*********\nHAVE A
NICE DAY AHEAD :)";
}
else if (pastry_name == 'P') //billing for puff pastry
{
bill = 3000 * pas_quantity;
cout << "Your bill is: " << bill << endl;
this->setName();
this->setPastryInfo("Puff");
cout << "\n*******THANK YOU SOO MUCH " << cust_name << " FOR SHOPPING*********\nHAVE A
NICE DAY AHEAD :)";
}
else //invalid option
cout << "\n INVALID....ENTER AGAIN";
}
};


int main()
{
cout << "**************************WELCOME TO THE SWEET
DELICE**********************************" << endl << "------------WE WILL TRY TO
PROVIDE YOU THE BEST QUALITY ITEMS---------------";
cout << "\n\nWHAT DO YOU WANT TO ORDER?" << endl;
cout << "\nCHOOSE 1 FOR CAKE, 2 FOR COOKIES, 3 FOR PASTERIES..." << endl;
bakery *b1, *b2, *b3;
cakes C;
cookies CO;
pastry P;
b1 = &C;
b2 = &CO;
b3 = &P;
int a;
for (int i = 0; i < 10; i++)
{
cout << "\n\n\n************************CHOOSE ONE CHOICE AMONG THE FOLLOWING
CATEGORIES************************" << "\n1. CAKE\n2. COOKIE\n3. PASTRY" << endl;
cout << "\nENTER YOUR CHOICE? " << endl;
cin >> a;
if (a == 1)
{
C.show_cakes();
b1->cash();
}
else if (a == 2)
{
CO.show_cookies();
b2->cash();
}
```

```
else if (a == 3)
{
P.show_pastry();
b3->cash();
}
else
cout << "\n SORRY! THE OPTION YOU ENTERED DOES NOT EXIST....\nKINDLY TRY AGAIN.....";
}
system("pause");
return 0;
}
```

# CLASS DIAGRAM: