

**Name: Jamshed Bashir**

**Enrollment No: 01-134212-072**

**Class: BS (CS) 2C**

**OBJECT ORIENTED PROGRAMMING**

**PROJECT**

**Submitted To:**

**Mam Erum Ashruf**

**Department of Computer Science**

**BAHRIA UNIVERSITY ISLAMABAD**

**Classes**:

1. Coffee (Base Class):
   1. Constructor: Initializes the name, the quantities, the id’s, and the rates of the items present in the coffee shop.
   2. Get-Details: Displays the menu list of the coffee shop. This menu list contains the ID’s, name’s, and the prices of the coffees.
2. Admin (Derived publicly by Base Class Coffee):
   1. Constructor: Initializes the username and password for the admin of the coffee shop.
   2. Set-User: Sets the username.
   3. Set-Pass: Sets the password.
   4. Get-User: Returns the username.
   5. Get-Pass: Returns the password.
   6. Edit-Details: In this function, admin can add items, delete items and change the details of the items (name, quantity or price).
   7. Payment-Details: This function shows all the orders delivered and their payments by file handling.
   8. Check-Avail: This function shows the quantities of the items present in the coffee shop.
3. Customer (Derived publicly by Base Class Coffee):
   1. Constructor: Initializes a member variable you.
   2. Del-Order: This function clears the members variables to memory space null so that user can give a new order.
   3. Set-Order: This function take the id ’s of the order as a parameter and set the member variable order.
   4. Get-Order: Assigns the name of the items according their id and returns them.
   5. Set-Pay: Sets the prices to the member variable according the id of the items assigned previously to the member variable order.
   6. Get-Pay: Returns the prices of the items according to their id’s.
   7. Delete-Order: This function make the global variable tw = 1 which then shows the order is deleted.
   8. Check-Quantity: This function checks whether the quantity given by the user is zero or smaller than zero or greater than the quantity available or the quantity of the item is zero and then gives an error. And if the quantity is greater than zero and smaller than the quantity of the item available then it subtracts the quantity from the actual quantity of the item available.

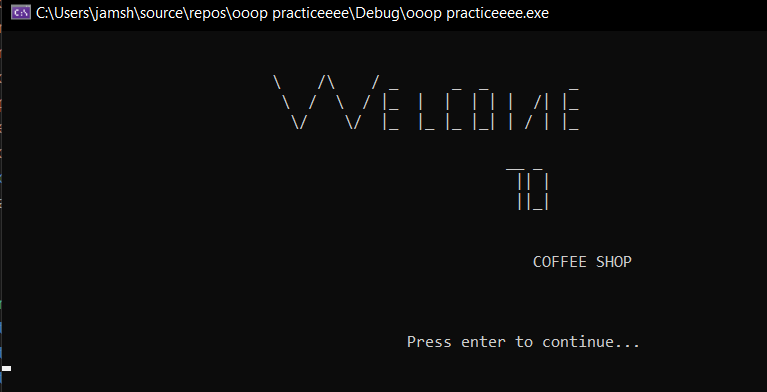
**UML Diagram**:

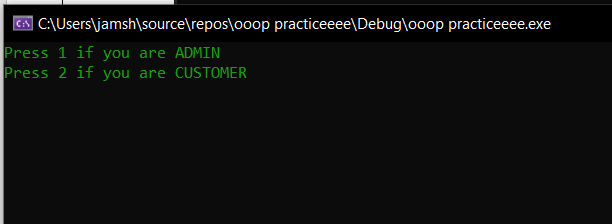
|  |
| --- |
| Coffee |
| Name: string  Quantity: int  Id: int  Rates: int |
| +Coffee ():  +getDetails (): void |

|  |
| --- |
| Customer |
| - order: int  - pay: int  - you: int |
| +Customer ():  +delOrder (): void  +setOrder(str[] : int, a : int) : int  +getOrder(a : int) : string  +setPay(you : int) : void  +getPay(count : int) : int  +deleteOrder() : void  +checkQuantity(a : int, b : int) : int |

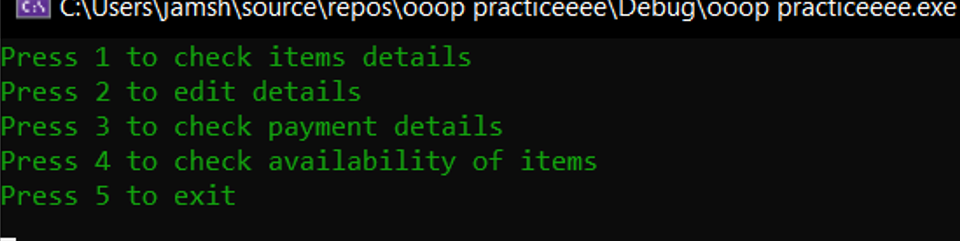
|  |
| --- |
| Admin |
| - user: string  - pass: string |
| +Admin ():  +setUser(a : string) : void  +setPass(a : string) : void  +getUser() : string  +getPass() : string  +editDetails(a : int) : void  +paymentDetails() : void  +checkAvail() : void |

**Screenshot:**

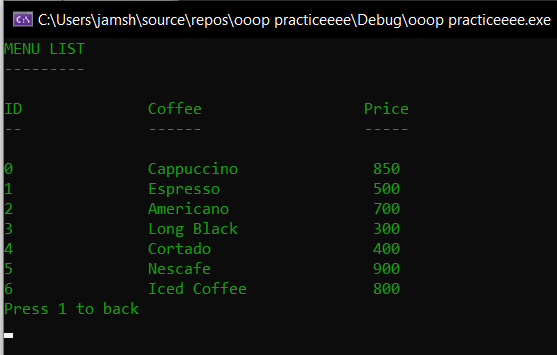
****

****

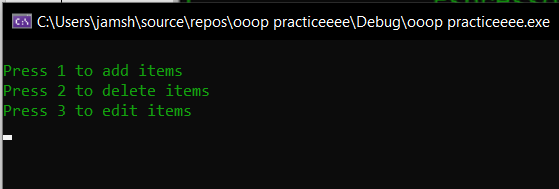
* **If we Press 1 and it will Login as Admin…**



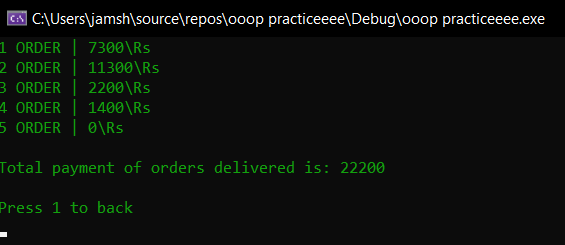
* If we Press 1



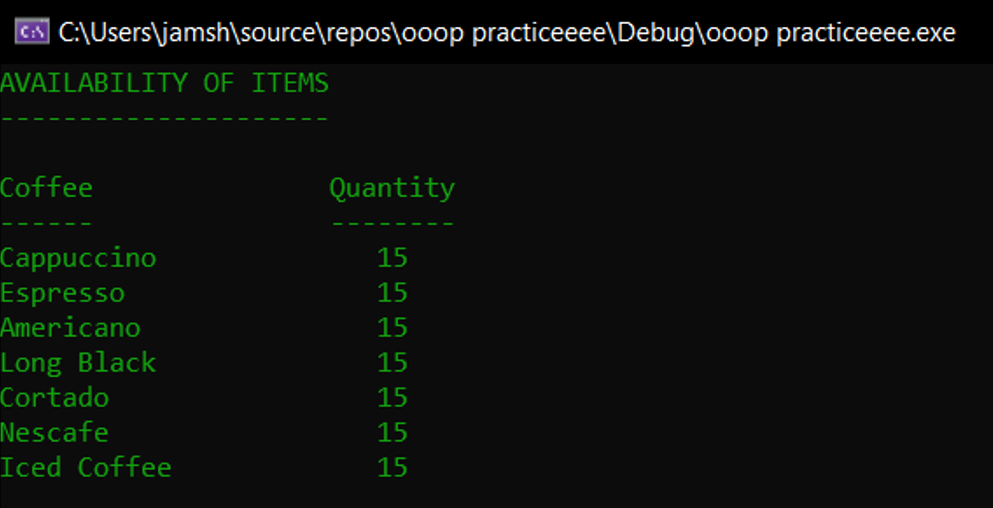
* If we Press 2



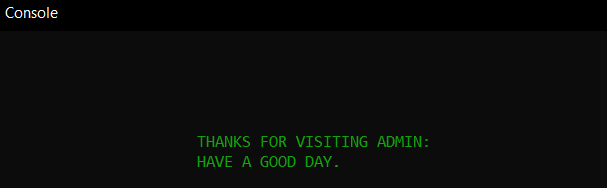
* If we Press 3:



* If we Press 4:



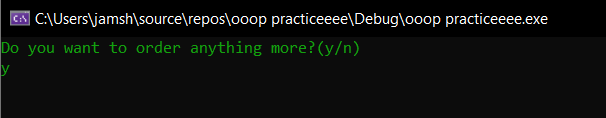
* If we press 5:



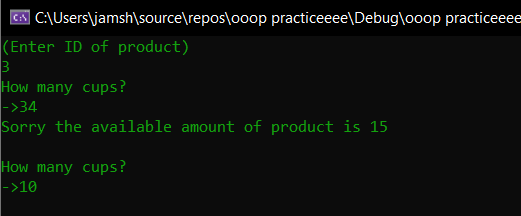
* **If we Press 2 it will Login as a Customer…**

****

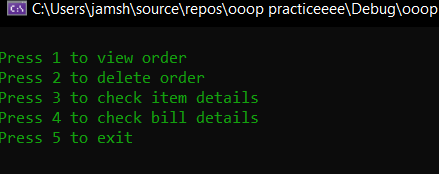
* **It is asking, If we want to order any other thing:**

****

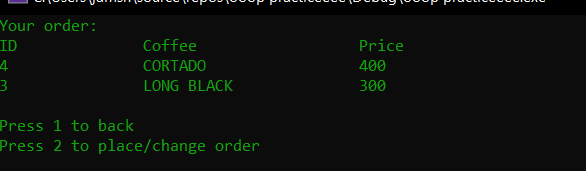
* **If we press “y” then It will ask which product to add:**

****

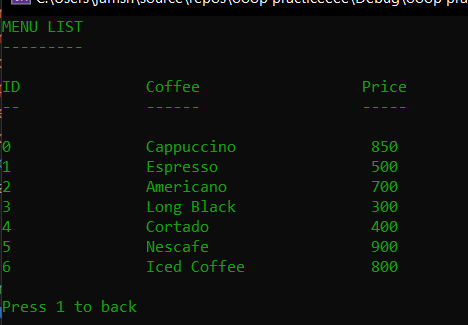
* **If we press “n” it will shift customer’s order to confirm:**

****

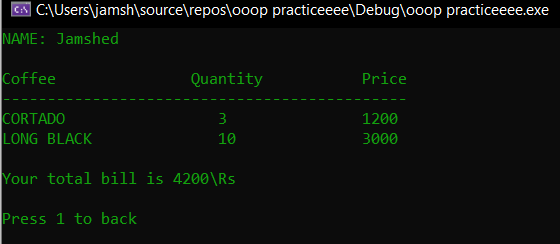
* **If we Press 1, it will show customer’s order:**

****

* **If we Press 3, it will show items of main menu:**

****

* **If we Press 4, it will show customer’s bill details:**

****

* **If we Press 5, it will terminate the program:**



**Source code:**

#include<iostream>

#include<string>

#include<vector>

#include<algorithm>

#include<iomanip>// isspace and isalpa characters

#include<Windows.h>// for sleep()

#include<cctype>

#include<fstream>// file handling

#include<conio.h>

using namespace std;

int tw = 0, payment;

class Coffee

{

protected:

vector<string> name;

vector<int> quantity;

vector<int> id;

vector<int> rates;

public:

Coffee()

{

name.push\_back("Cappuccino");

name.push\_back("Espresso");

name.push\_back("Americano");

name.push\_back("Long Black");

name.push\_back("Cortado");

name.push\_back("Nescafe");

name.push\_back("Iced Coffee");

rates.push\_back(850);

rates.push\_back(500);

rates.push\_back(700);

rates.push\_back(300);

rates.push\_back(400);

rates.push\_back(900);

rates.push\_back(800);

quantity.push\_back(15);

quantity.push\_back(15);

quantity.push\_back(15);

quantity.push\_back(15);

quantity.push\_back(15);

quantity.push\_back(15);

quantity.push\_back(15);

id.push\_back(0);

id.push\_back(1);

id.push\_back(2);

id.push\_back(3);

id.push\_back(4);

id.push\_back(5);

id.push\_back(6);

}

void getDetails()

{

int a;

system("CLS");

cout << "MENU LIST\n---------" << endl;

cout << endl;

cout << "ID\t\tCoffee\t\t\tPrice" << endl;

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

cout << endl;

for (int i = 0;i < name.size();i++)

{

a = name[i].length();

while (a < 11)

{

name[i] += " ";

a++;

}

cout << id[i] << "\t\t" << name[i] << "\t\t " << rates[i] << endl;

}

}

};

class Admin :public Coffee

{

string user;

string pass;

public:

Admin()

{

//Setting the username and password

user = "admin";

pass = "12345";

}

void setUser(string a)

{

user = a;

}

void setPass(string a)

{

pass = a;

}

string getUser()

{

return user;

}

string getPass()

{

return pass;

}

void editDetails(int a)

{

int d, b, e = 0;

string c;

id.push\_back(id.size());

if (a == 1)

{

system("CLS");

cout << endl;

cout << "\t\t\t\tADD ITEMS" << endl;

cout << endl;

cout << "Enter name of the item\n->";

cin >> c;

cout << "Enter quantity of the item\n->";

cin >> d;

cout << "Enter price of the item\n->";

cin >> b;

name.push\_back(c);

quantity.push\_back(d);

rates.push\_back(b);

cout << endl;

cout << "Item added successfully" << endl;

cout << endl;

}

else if (a == 2)

{

int a = 0;

getDetails();

cout << endl;

cout << "Enter ID of the quantity to be deleted\n->";

cin >> a;

name.erase(name.begin() + a);

rates.erase(rates.begin() + a);

quantity.erase(quantity.begin() + a);

cout << endl << "The item is deleted successfully" << endl;

cout << endl;

}

else if (a == 3)

{

int a = 0, b = 0, d, e;

string c;

getDetails();

cout << "Enter ID of the item to be changed\n->";

cin >> a;

da:

cout << "Press 1 to change name\nPress 2 to change Quantity\nPress 3 to change Price\n->";

cin >> b;

if (b == 1)

{

cout << "Enter name\n->";

cin >> c;

name[a] = c;

cout << endl << "The name of the item is changed successfully" << endl;

cout << endl;

}

else if (b == 2)

{

cout << "Enter Quantity\n->";

cin >> d;

quantity[a] = d;

cout << endl << "The quantity of the item is changed successfully" << endl;

cout << endl;

}

else if (b == 3)

{

cout << "Enter Price\n->";

cin >> e;

rates[a] = e;

cout << endl << "The price of the item is changed successfully" << endl;

cout << endl;

}

else

{

cout << "Invalid Entry" << endl;

goto da;

}

}

}

void paymentDetails()

{

int i = 0, count = 0, a=0, add = 0;

//Opening the file

ifstream payfile("order.txt");

while (!payfile.eof())

{

//Reading the file items

count++;

int item;

payfile >> item;

if (item == a)

{

break;

}

add += item;

cout << i + 1 << " ORDER | ";

cout << item << "\\Rs" << endl;

i++;

a = item;

}

cout << endl;

cout << "Total payment of orders delivered is: " << add << endl;

}

void checkAvail()

{

//Checking Quantity

cout << "Coffee\t\t Quantity" << endl;

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

for (int i = 0;i < name.size();i++)

{

while (name[i].size() < 11)

{

name[i] += " ";

}

cout << name[i] << "\t\t" << quantity[i] << endl;

}

}

};

class Customer :public Coffee

{

vector<int> order;

vector<int> pay;

int you;

public:

Customer()

{

you = 0;

}

void delOrder()

{

//Clearing the vectors

order.clear();

pay.clear();

}

void setOrder(int str[], int a)

{

//Adding ID's to vector

for (int i = 0;i < a / 2;i++)

{

order.push\_back(str[i]);

}

}

string getOrder(int a)

{

//Return order according to ID

string name1;

if (tw == 0)

{

switch (order[a])

{

case 0:

name1 = "CAPPUCCINO";

break;

case 1:

name1 = "ESPRESSO";

break;

case 2:

name1 = "AMERICANO";

break;

case 3:

name1 = "LONG BLACK";

break;

case 4:

name1 = "CORTADO";

break;

case 5:

name1 = "NESCAFE";

break;

case 6:

name1 = "ICED COFFEE";

break;

default:

return " ";

break;

}

while (name1.size() < 11)

{

name1 += " ";

}

return name1;

}

else if (tw == 1)

{

return "N/A ";

}

}

void setPay(int you)

{

//Setting the payment of orders

//by using iterators

vector<int>::iterator itr1 = order.begin();

vector<int>::iterator itr2 = order.end();

for (vector<int>::iterator itr = itr1;itr != itr2;itr++)

{

if (\*itr == 0)

{

pay.push\_back(rates[0]);

}

else if (\*itr == 1)

{

pay.push\_back(rates[1]);

}

else if (\*itr == 2)

{

pay.push\_back(rates[2]);

}

else if (\*itr == 3)

{

pay.push\_back(rates[3]);

}

else if (\*itr == 4)

{

pay.push\_back(rates[4]);

}

else if (\*itr == 5)

{

pay.push\_back(rates[5]);

}

else if (\*itr == 6)

{

pay.push\_back(rates[6]);

}

}

}

int getPay(int count)

{

//Returning payment

int a;

if (tw == 0)

{

a = pay[count];

return a;

}

else if (tw == 1)

{

return 0;

}

}

void deleteOrder()

{

//Logic used for deleting order

tw = 1;

}

int checkQuantity(int a, int b)

{

//Checking Quantity whether it is

//smaller than 0 or

//negative quantity or

//equal to 0 or

//greater than the quantity available or

//whether the item is available or not

static int count1 = 0;

int count = 0;

if (a < 0)

{

cout << "Please enter a positive amount" << endl;

count = 3;

return count;

}

if (a == 0)

{

cout << "Please enter some quantity" << endl;

count = 3;

return count;

}

else if (quantity[b] == 0)

{

cout << endl;

cout << "Sorry, the product is out of stock\nTry another product" << endl;

count = 1;

return count;

}

else if (a > quantity[b])

{

cout << "Sorry the available amount of product is " << quantity[b] << endl;

count = 2;

return count;

}

else if (a <= quantity[b])

{

quantity[b] -= a;

count = 0;

return count;

}

}

};

void intro()

{

cout << endl << endl;

cout << "\t\t\t \\ /\\ / \_ \_ \_ \_ " << endl;

cout << "\t\t\t \\ / \\ / |\_ | | | | |\ /| |\_ " << endl;

cout << "\t\t\t \\/ \\/ |\_ |\_ |\_ |\_| | \/ | |\_ " << endl;

cout << endl;

cout << "\t\t\t\t\t\t\t\_\_ \_ " << endl;

cout << "\t\t\t\t\t\t\t || |" << endl;

cout << "\t\t\t\t\t\t\t ||\_|" << endl;

cout << endl;

cout << endl;

cout << "\t\t\t\t\t\t COFFEE SHOP" << endl;

cout << endl << endl << endl;

cout << "\t\t\t\t\t Press enter to continue..." << endl;

cin.ignore();

system("CLS");

}

void extro(int a)

{

if (a == 0)

{

system("CLS");

cout << "\t\t\tYOUR ORDER IS READY\n\t\t\tGET IT AND ENJOY\n\t\t\tTHANKS FOR ORDERING" << endl;

Sleep(1500);

system("CLS");

cout << endl << endl << endl;

}

else if (a == 1)

{

system("CLS");

cout << endl << endl << endl << endl << endl;

cout << "\t\t\t\tThanks for visiting VOID-COFFEE SHOP" << endl;

cout << "\t\t\t\t (Available on your one touch)" << endl;

cout << endl;

cout << "\t\t\t\tYour order will be ready in few minutes..." << endl;

cout << endl;

Sleep(2000);

system("CLS");

cout << endl << endl << endl;

}

cout << "\t\t \_\_\_\_ \_\_\_\_\_ \_\_\_\_ \_\_\_\_ " << endl;

cout << "\t\t || ) || || || || || |||| || || \_\_\_ \_\_\_\_ \_\_\_ \_\_\_\_\_\_ " << endl;

cout << "\t\t ||\_\_\_\_) ||\_\_|| ||\_\_|| ||\_// || ||\_\_|| | || |\\ | || \\ / || | // || || || \\ / " << endl;

cout << "\t\t || ) || || || || || \\ || || || | || | \\ | || \\ / ||\_ |-\\ ---- || || \\ / " << endl;

cout << "\t\t ||\_\_\_\_) || || || || || \\ || || || |\_\_\_|| | \\| || \\/ ||\_ | \\ \_\_\_|| || || || " << endl;

cout << endl;

cout << "\t\t\t\tProject BY:\n\t\t\t\t\t JAMSHED BASHIR \n\t\t\t\t\t 01-134212-072\n\t\t\t\t\t " << endl;

cout << endl;

cout << "\t\t\t\tSubmitted TO:\n\t\t\t\t\t MAM ERUM ASHRUF" << endl;

Sleep(130);

exit(0);

}

void system\_color(int color)

{

HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(hConsole, color);

}

void loading(int load\_time, int bar\_length)

{

int color\_code;

for (int i = 0;i <= 1;i++) {

color\_code = 1 + i;

system\_color(color\_code);

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

cout << "\t\t\t\t\t\t\tLOADING Please wait ..... \n\n";

char x = 178;

for (int i = 0; i < bar\_length; i++)

{

cout << x;

Sleep(load\_time);

}

system("CLS");

}

}

int main()

{

//Introduction of project

intro();

loading(1, 120);

Admin admin;

Customer cust;

vector<int> id;

vector<int> quant;

//File variable

ofstream payfile;

int back = 0, back1 = 0, back2 = 0, back3 = 0;

int count = 0, count2 = 0, count3 = 0;

string name, j, k, l = "";

int a, c, b, d, e=0, f = 0, g = 0, h = 0, count1 = 0;

int i = 0, m = 0, n = 0;

int ch;

do

{

cout << "Press 1 if you are ADMIN\nPress 2 if you are CUSTOMER" << endl;

cin >> a;

//ADMIN

if (a == 1)

{

system("CLS");

do

{

back1 = 0;

//Username: admin

cout << "Username: ";

cin >> j;

//Password: 12345

cout << "Password: ";

ch = \_getch();

while (ch != 13 )

{

k.push\_back(ch);

cout << '\*';

ch = \_getch();

}

if (j == admin.getUser() && k == admin.getPass())

{

cout << "\n\n\n\n\n\n\n\t\t\t\t\t\t ACCESS GRANTED" << endl;

loading(1, 60);

}

else

{

cout << endl;

cout << "Invalid Username or Password" << endl;

cout << "Try Again" << endl;

back1 = 1;

cout << endl;

}

} while (back1 == 1);

cout << endl;

qw:

do

{

system("CLS");

n = 0;

//Admin Details

cout << "Press 1 to check items details\nPress 2 to edit details\nPress 3 to check payment details\nPress 4 to check availability of items\nPress 5 to exit" << endl;

cin >> i;

if (i == 1)

{

//To check item details

admin.getDetails();

cout << "Press 1 to back" << endl;

cin >> n;

}

else if (i == 2)

{

//To change ite details

system("CLS");

ha:

cout << "\nPress 1 to add items\nPress 2 to delete items\nPress 3 to edit items" << endl;

cin >> m;

if (m == 1 || m == 2 || m == 3) {

admin.editDetails(m);

cout << "Press 1 to back" << endl;

cin >> n;

}

else {

cout << "\nError enter a correct number:";

goto ha;

}

}

else if (i == 3)

{

//To check payment of orders

system("CLS");

admin.paymentDetails();

cout << endl;

cout << "Press 1 to back" << endl;

cin >> n;

}

else if (i == 4)

{

//To check quantity of items

system("CLS");

cout << "AVAILABILITY OF ITEMS" << endl;

cout << "---------------------" << endl;

cout << endl;

admin.checkAvail();

cout << endl;

cout << "Press 1 to back" << endl;

cin >> n;

}

else if (i == 5)

{

//Exit

system("CLS");

loading(1, 60);

cout << endl << endl << endl << endl << endl;

cout << "\t\t\t\t\t\tTHANKS FOR VISITING ADMIN:" << endl;

cout << "\t\t\t\t\t\tHAVE A GOOD DAY." << endl;

Sleep(100);

exit(0);

}

else

{

cout << "Invalid entry" << endl;

goto qw;

}

} while (n == 1);

}

else if (a == 2)//Customer

{

system("CLS");

int namechk=0;

do

{

namechk = 0;

cout << "Enter your Name: " << endl;

cin.ignore();

getline(cin, name);

loading(1, 60);

for (int i = 0;i< name.size();i++)

{

if (isalpha(name[i]) || isspace(name[i]))

{

namechk = 0;

continue;

}

else

{

cout << "Please enter your name in alphabets" << endl;

namechk = 1;

}

}

cout << endl;

} while (namechk == 1);

b:

count = 0;

cust.delOrder();

cust.getDetails();

cout << endl;

a:

cout << "Place your Order:" << endl;

do

{

do

{

b = 0;count2 = 0;f = 0;back3 = 0;

cout << "(Enter ID of product)" << endl;

cin >> b;

for (int i = 0;i < 7;i++)

{

if (b != i)

{

count2++;

}

else if (b == i)

{

//ID added to vector

id.push\_back(b);

count++;

break;

}

}

if (count2 == 7)

{

cout << "Invalid Entry" << endl;

cout << endl;

goto a;

}

do

{

g = 0;

//Number of cups to order

cout << "How many cups?" << "\n->";

cin >> d;

switch (cust.checkQuantity(d, b))

{

case 1:

f = 1;

cout << endl;

break;

case 2:

g = 1;

cout << endl;

break;

case 3:

g = 1;

cout << endl;

break;

default:

quant.push\_back(d);

break;

}

} while (g == 1);

} while (f == 1);

do

{

//To try one more coffee

back2 = 0;

loading(1, 60);

cout << "Do you want to order anything more?(y/n)" << endl;

cin >> l;

if (l == "y" || l == "Y")

{

back3 = 1;

count1++;

loading(1, 60);

}

else if (l == "n" || l == "N")

{

back3 = 0;

back2 = 0;

loading(1, 60);

}

else

{

back2 = 0;

cout << endl;

cout << "Invalid Entry" << endl;

cout << "Enter y for yes\nEnter n for no" << endl;

cout << endl;

back2 = 1;

}

} while (back2 == 1);

} while (back3 == 1);

//Itereators of the vectors

vector<int>::iterator itr1 = id.begin();

vector<int>::iterator itr2 = id.end();

vector<int>::iterator itr3 = quant.begin();

vector<int>::iterator itr4 = quant.end();

count \*= 2;

int\* str = new int[count];

int ir = 0;

for (vector<int>::iterator itr = itr1;itr != itr2;itr++)

{

//Add ID to the array

str[ir] = \*itr;

ir++;

}

for (vector<int>::iterator itr = itr3;itr != itr4;itr++)

{

//Add Quantity to the array

str[ir] = \*itr;

ir++;

}

//Setting order

cust.setOrder(str, count);

cout << endl;

cout << "Your order is placed. Please Wait..." << endl;

//Setting the payment of the order

cust.setPay(count);

Sleep(100);

system("CLS");

cout << "While your order is under progress\nYou can enjoy some of the features:" << endl;

loading(1, 60);

cout << endl;

while (true)

{

do

{

//Customer Features

cout << "Press 1 to view order\nPress 2 to delete order\nPress 3 to check item details\nPress 4 to check bill details\nPress 5 to exit" << endl;

if (count3 > 1)

{

//Writing the payment in file

if (tw == 0)

{

Sleep(900);

payfile.open("order.txt", ios\_base::app);

payfile << payment << endl;

payfile.close();

extro(0);

}

}

cin >> c;

if (c == 1)

{

//Viewing Order

int a;

count3++;

system("CLS");

if (tw == 1)

{

count3 = 0;

a = count;

cout << "You have'nt ordered anything" << endl;

b = 0;

for (int i = 0;i < a;i++)

{

str[i] = 0;

}

}

cout << "Your order:\n";

cout << "ID\t\tCoffee\t\t\tPrice" << endl;

{

a = count / 2;

for (int i = 0;i < a;i++)

{

cout << str[i] << "\t\t" << cust.getOrder(i) << "\t\t" << setw(3) << cust.getPay(i) << endl;

}

}

cout << endl;

cout << "Press 1 to back\nPress 2 to place/change order" << endl;

cin >> h;

if (h == 2)

{

//Changing order

system("CLS");

tw = 0;

id.clear();

quant.clear();

goto b;

}

system("CLS");

}

else if (c == 2)

{

//Deleting Order

system("CLS");

cust.deleteOrder();

cout << "Your ordered is deleted.." << endl;

cout << "Press 1 to back" << endl;

cin >> h;

system("CLS");

}

else if (c == 3)

{

//Checking Item Details

system("CLS");

cust.getDetails();

cout << endl;

cout << "Press 1 to back" << endl;

cin >> h;

system("CLS");

}

else if (c == 4)

{

//Bill of the order

int a;

count3++;

int b = 0;

if (tw == 1)

{

a = count;

count3 = 0;

cout << "You have'nt ordered anything" << endl;

b = 0;

for (int i = 0;i < a;i++)

{

str[i] = 0;

}

}

system("CLS");

cout << "NAME: " << name << endl;

cout << endl;

cout << "Coffee\t\t Quantity\t\tPrice" << endl;

cout << "---------------------------------------------" << endl;

b = count / 2;

a = count / 2;

for (int i = 0;i < a;i++)

{

cout << cust.getOrder(i) << "\t\t" << str[b] << "\t\t" << cust.getPay(i) \* str[b] << endl;

payment = payment + (cust.getPay(i) \* str[b]);

b++;

}

cout << endl;

cout << "Your total bill is " << payment << "\\Rs" << endl;

cout << endl;

cout << "Press 1 to back" << endl;

cin >> h;

system("CLS");

}

else if (c == 5)

{

//Path to exit

payfile.open("order.txt", ios\_base::app);

payfile << payment << endl;

payfile.close();

extro(1);

}

} while (h == 1);

}

}

else

{

cout << "Invalid Entry" << endl;

back = 1;

}

} while (back == 1);

system("pause");

return 0;

}