#include <iostream>

using namespace std;

class coffee\_shop{

int order, rear=-1, front=-1, n=10;

int queue[10];

public:

void enqueue();

void dequeue();

bool isFull();

bool isEmpty();

void display();

coffee\_shop(int od, int re, int fo){

od = order;

re = rear;

fo = front;

}

};

bool coffee\_shop :: isFull(){

if(rear==n-1){

return 1;

}

else{

return 0;

}

}

bool coffee\_shop :: isEmpty(){

if(rear<front || front == n || rear==-1){

return 1;

}

else{

return 0;

}

}

void coffee\_shop :: enqueue(){

if(isFull()){

cout << "Sorry queue is already full please come after some time\n";

}

else{

cout << "Please place your order\n";

cin >> order;

rear = rear + 1;

queue[rear] = order;

front = 0;

}

}

void coffee\_shop :: dequeue(){

if(isEmpty()){

cout << "Sorry you don't have any order\n";

}

else{

int temp = queue[front];

front = front + 1;

}

}

void coffee\_shop :: display(){

for(int i=front; i<=rear; i++){

cout << queue[i] << endl;

}

}

int main(){

int o, r, f;

coffee\_shop customer(o, r, f);

int task;

do{

cout << "Please select a task\n";

cout << "1.Place order\n2.Delete order\n3.Exit\n";

cin >> task;

switch(task){

case 1:

cout << "Please place your order\n";

customer.enqueue();

break;

case 2:

cout << "We are deleting your order\n";

customer.dequeue();

break;

case 3:

cout << "Your order is as follows\n";

customer.display();

break;

default:

cout << "Please enter some valid task\n";

}

}

while(task != 3);

return 0;

}

***OUTPUT -***

Please select a task

1.Place order

2.Delete order

3.Exit

1

Please place your order

Please place your order

34

Please select a task

1.Place order

2.Delete order

3.Exit

2

We are deleting your order

Please select a task

1.Place order

2.Delete order

3.Exit

3

Your order is as follows

------------------

(program exited with code: 0)

Press any key to continue . . .