

BACS1013 & 1014 PROBLEM SOLVING AND PROGRAMMING

YEAR 1 – SEMESTER 1

COURSEWORK 1

NAME: PHANG WAI HONG

STUDENT.ID: 19WMR11261

PROGRAMME: RSY1

TUTORIAL GROUP: 1

TUTOR NAME: MR. SHOONG WAI KIN

**Declaration of Originality**

**Content**

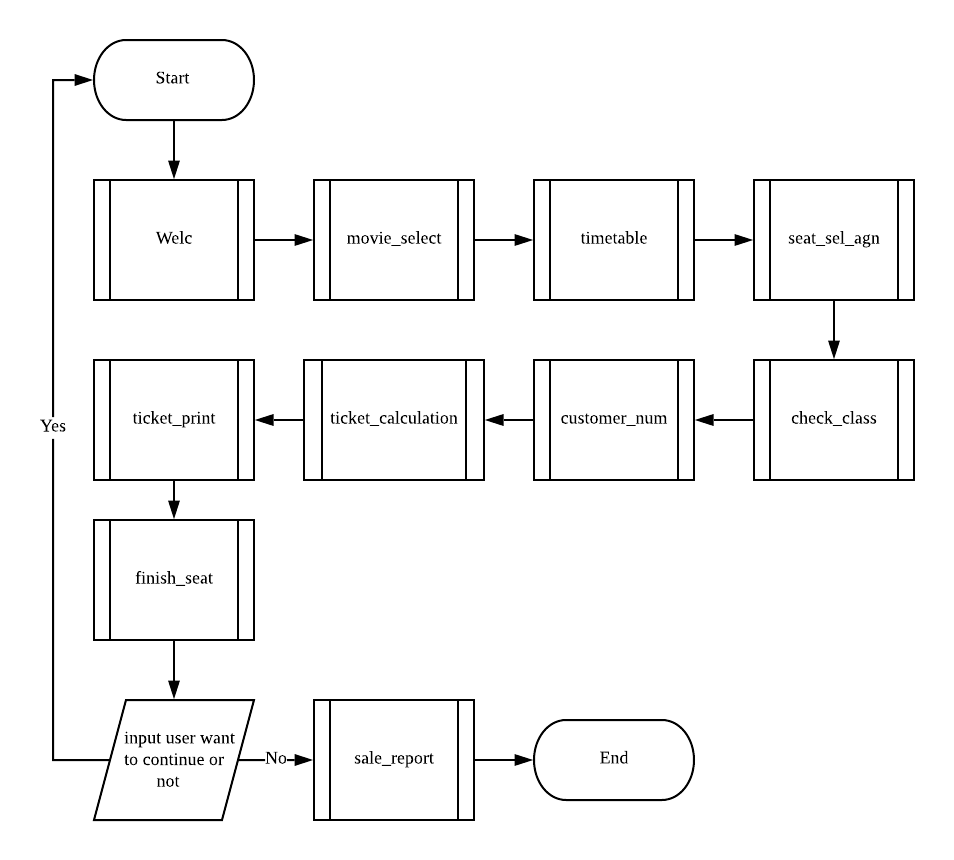
|  |  |  |
| --- | --- | --- |
| No. | Content | Page |
| 1 | Declaration of Originality | 1 |
| 2 | Content | 2 |
| 3 | Introduction | 3 |
| 4 | Method of solution (FlowChart) | 4 – 7 |
| 5 | Constants & Variable | 8 – 10 |
| 6 | Structure Chart | 11 |
| 7 | Program Testing & Output | 12 – 24 |
| 8 | Source Code | 25 – 51 |

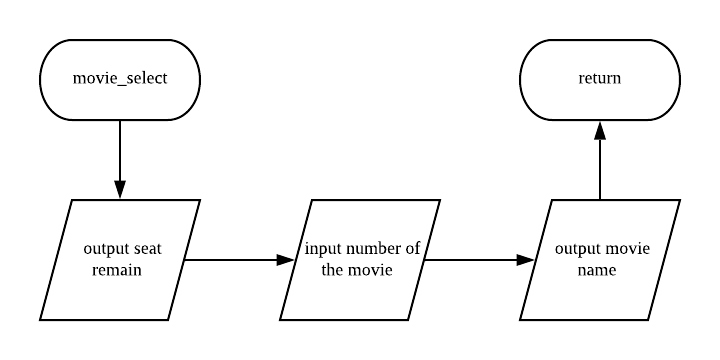
**Introduction**

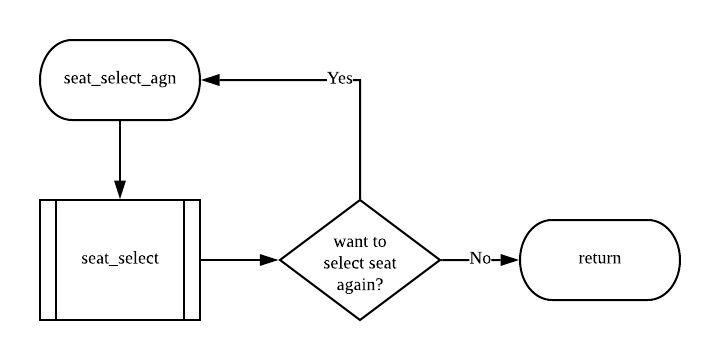
Danny Cinemax is a cinema that shows some popular movies on every Friday night, 8.00p.m. This cinema is opened for public included students. If students show their student’s card or student’s id, they can get some offer or cheaper price rate. In this program, we have inserted report on sale of ticket.

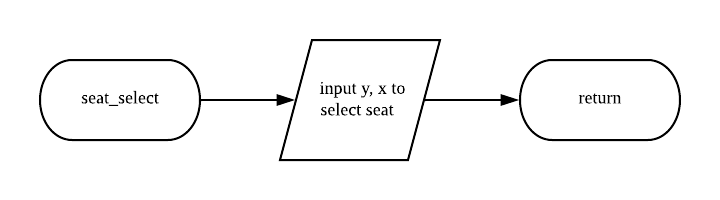
In this movie ticket system, we are allow customer to choose difference movies, seats, class of seat and the number of difference category of customer like adult, child and student. After that, this system will calculate the price of the tickets and allow user to key in their pay amount by cash only but not credit cards or online banking. After they done all the step of buying tickets, they can choose can to continue to buy next seat or not. If no, it will display the sales report of that day for management to check for their sales.

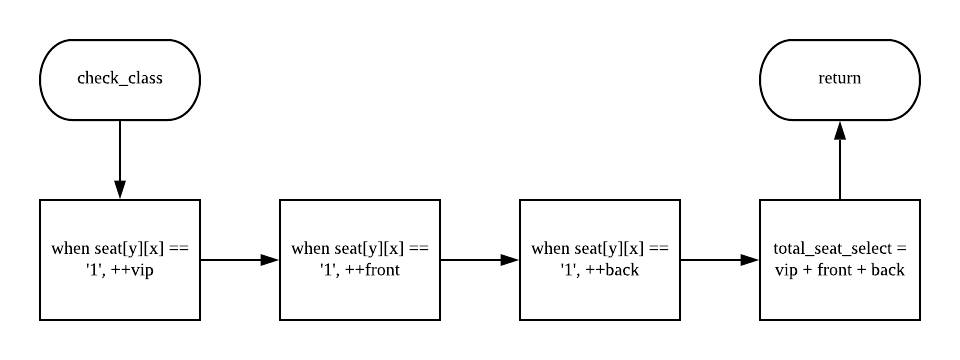
**Method of Solution**

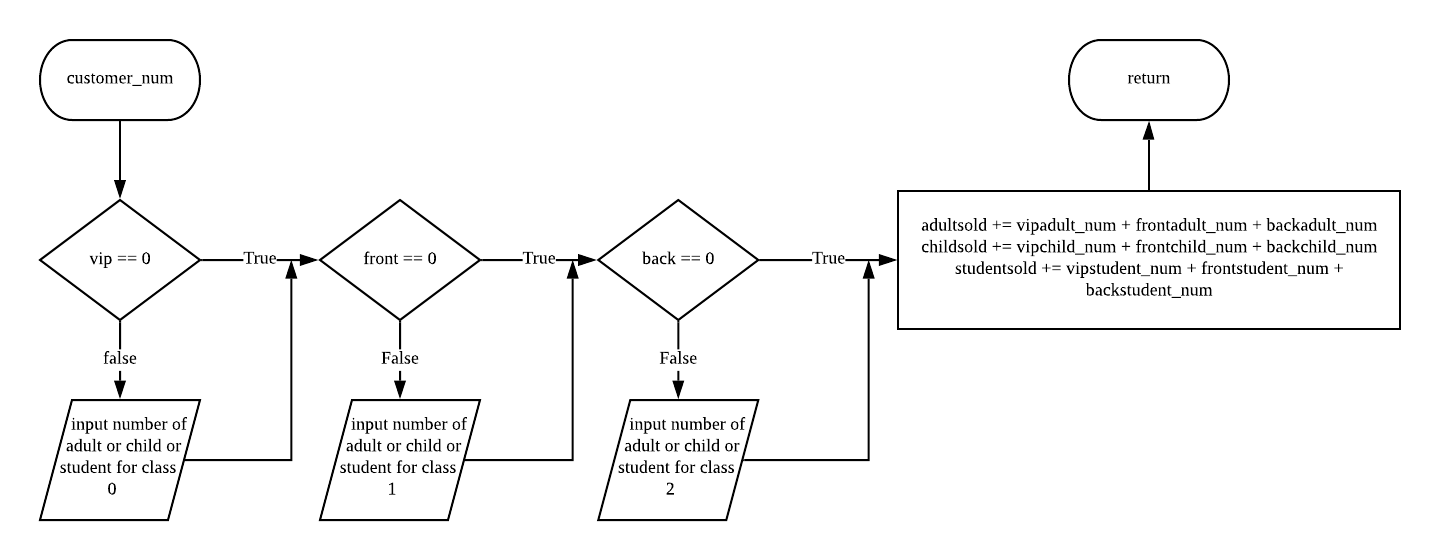
****

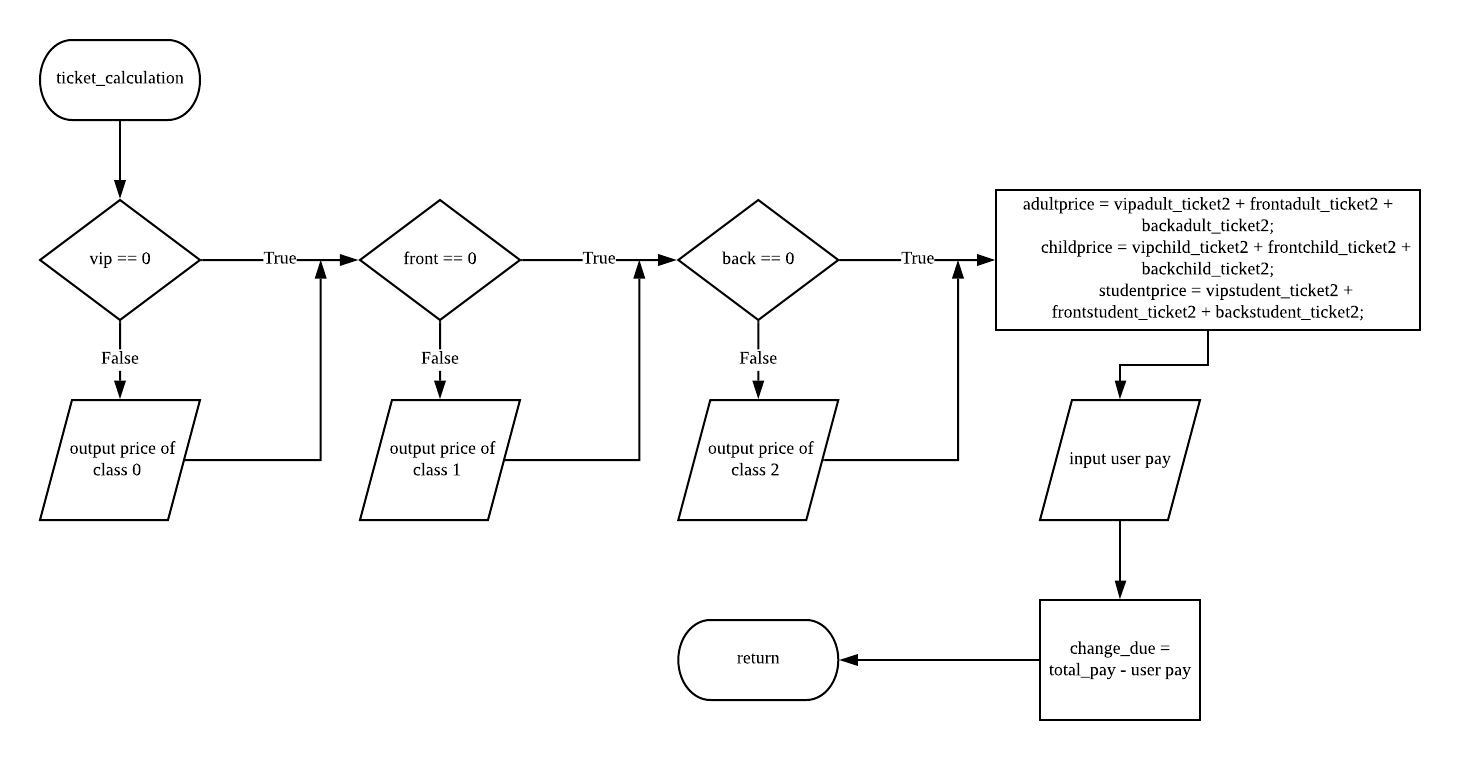
****

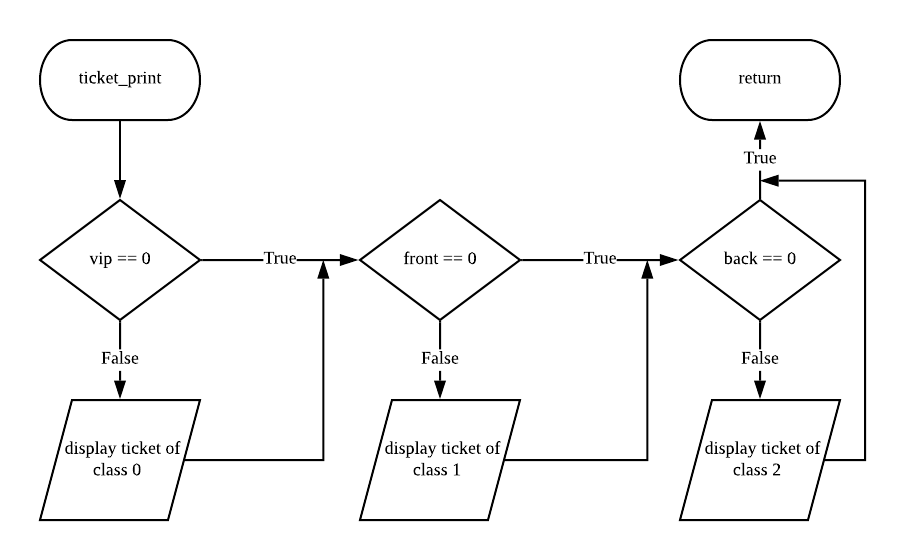
****

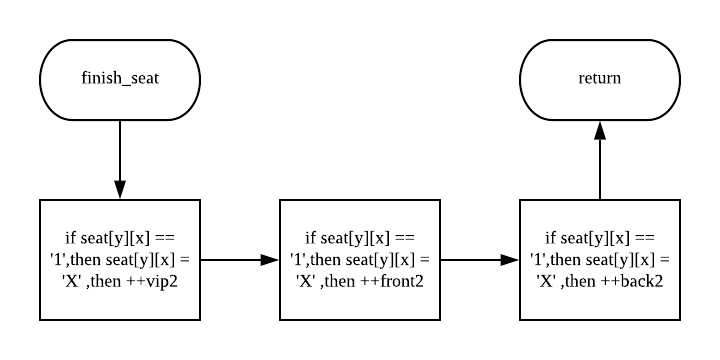
****

****

****

****

****

****

**Constants & Variables**

int movie\_num, y, x, seat\_icon3 = 254, vip, front, back,

vipadult\_num, vipchild\_num, vipstudent\_num,

frontadult\_num, frontchild\_num, frontstudent\_num,

backadult\_num, backchild\_num, backstudent\_num,

adultsold, childsold, studentsold,

total\_vip, total\_front, total\_back,

adultprice, childprice, studentprice,

vipadult\_ticket2, vipchild\_ticket2, vipstudent\_ticket2,

frontadult\_ticket2, frontchild\_ticket2, frontstudent\_ticket2,

backadult\_ticket2, backchild\_ticket2, backstudent\_ticket2,

vip2, front2, back2;

int total\_seat\_select;

int adult\_ticket, child\_ticket, student\_ticket, total\_to\_pay;

int total\_class\_left;

double cash\_payment, change\_due;

char seat[4][15],yes\_no;

string movie\_name[2] = { "FROZEN II", " Avenger 4: EndGame" };

//--for class 0--//

adult\_ticket = 20;

child\_ticket = 10;

student\_ticket = 12;

//--for class 1--//

adult\_ticket = 16;

child\_ticket = 6;

student\_ticket = 8;

//--for class 2--//

adult\_ticket = 18;

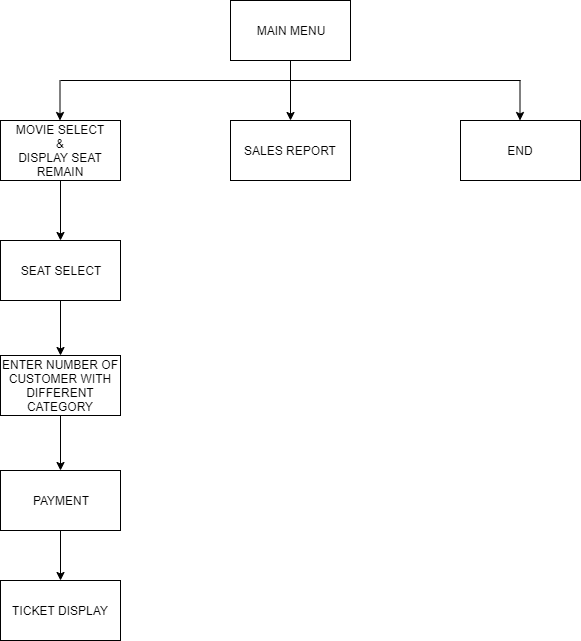
child\_ticket = 8;

student\_ticket = 10;

|  |  |  |  |
| --- | --- | --- | --- |
| **Constant** | **Value** | **Data Type** | **Purpose** |
| movie\_name | FRONZEN II,  Avenger 4: Endgame | STRING | To store the movie name in array |
| seat\_icon3 | 254 | INT | To set seat\_icon3 to ascii value |
| adultprice | 20 / 16 / 18 | INT | To set the different value of price for different class |
| childprice | 10 / 6 / 8 | INT | To set the different value of price for different class |
| studentprice | 12 / 8 / 10 | INT | To set the different value of price for different class |

|  |  |  |
| --- | --- | --- |
| **Variable** | **Data Type** | **Purpose** |
| movie\_num | INT | Save the number of movie select |
| y | INT | Save the seat select of row |
| x | INT | Save the seat select of column |
| vip | INT | Save the selected number of class 0 |
| front | INT | Save the selected number of class 1 |
| back | INT | Save the selected number of class 2 |
| vipadult\_num | INT | Save the number of adult for class 0 |
| vipchild\_num | INT | Save the number of child for class 0 |
| vipstudent\_num | INT | Save the number of student for class 0 |
| frontadult\_num | INT | Save the number of adult for class 1 |
| frontchild\_num | INT | Save the number of child for class 1 |
| frontstudent\_num | INT | Save the number of student for class 1 |
| backadult\_num | INT | Save the number of adult for class 2 |
| backchild\_num | INT | Save the number of child for class 2 |
| backstudent\_num | INT | Save the number of student for class 2 |
| adultsold | INT | Save the total number of adult ticket that sold |
| childsold | INT | Save the total number of child ticket that sold |
| studentsold | INT | Save the total number of student ticket that sold |
| total\_vip | INT | Save the total ticket price of class 0 |
| total\_front | INT | Save the total ticket price of class 1 |
| total\_back | INT | Save the total ticket price of class 2 |
| adultprice | INT | Save the total ticket price of adult |
| childprice | INT | Save the total ticket price of child |
| studentprice | INT | Save the total ticket price of student |
| vipadult\_ticket2 | INT | Save the total ticket price of adult for class 0 |
| vipchild\_ticket2 | INT | Save the total ticket price of child for class 0 |
| vipstudent\_ticket2 | INT | Save the total ticket price of student for class 0 |
| frontadult\_ticket2 | INT | Save the total ticket price of adult for class 1 |
| frontchild\_ticket2 | INT | Save the total ticket price of child for class 1 |
| frontstudent\_ticket2 | INT | Save the total ticket price of student for class 1 |
| backadult\_ticket2 | INT | Save the total ticket price of adult for class 2 |
| backchild\_ticket2 | INT | Save the total ticket price of child for class 2 |
| backstudent\_ticket2 | INT | Save the total ticket price of student for class 2 |
| vip2 | INT | Save the sold number of class 0 |
| front2 | INT | Save the sold number of class 1 |
| back2 | INT | Save the sold number of class 2 |
| total\_seat\_select | INT | Save the total number of seat selected |
| total\_to\_pay | INT | Save the total ticket price |
| total\_class\_left | INT | Save the total number of seat sold |
| total\_seat\_per | INT | Save the total seat sold percentage |
| cash\_payment | DOUBLE | Save the user pay amount |
| change\_due | DOUBLE | Save the balance of the payment |
| seat[4][5] | CHAR | Save the seat in 2D array |
| yes\_no | CHAR | Ask the user want to continue to buy next ticket or not |

**Structure Chart**



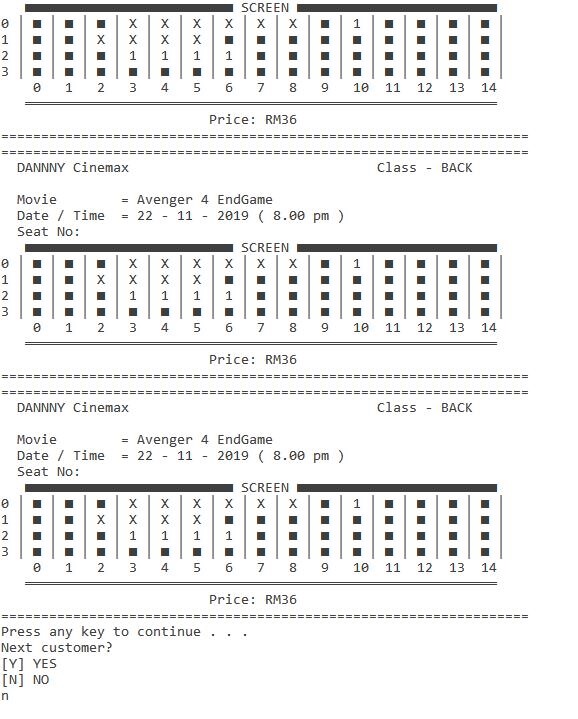
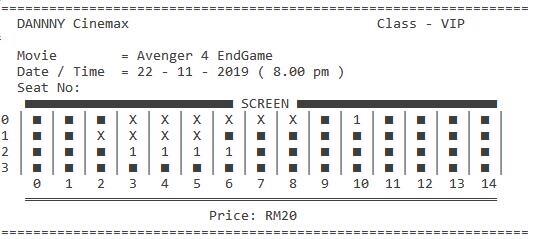
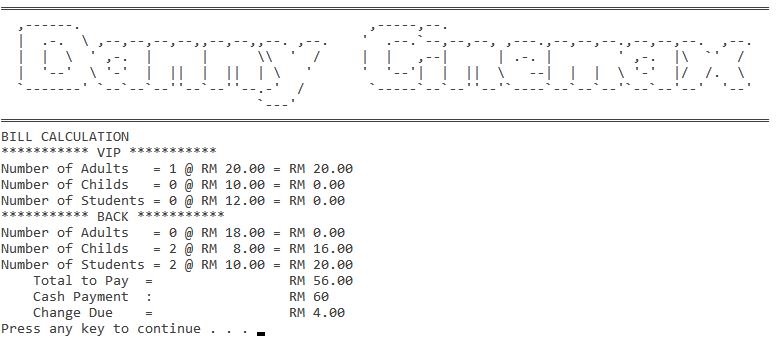
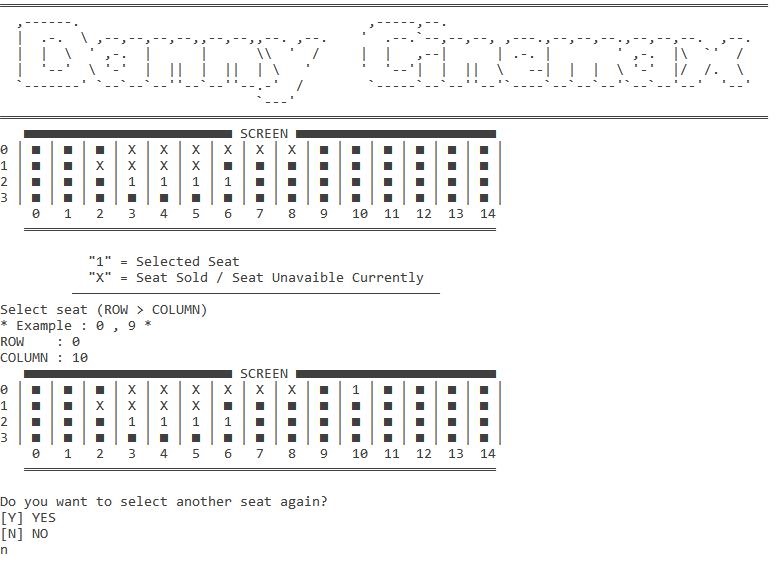
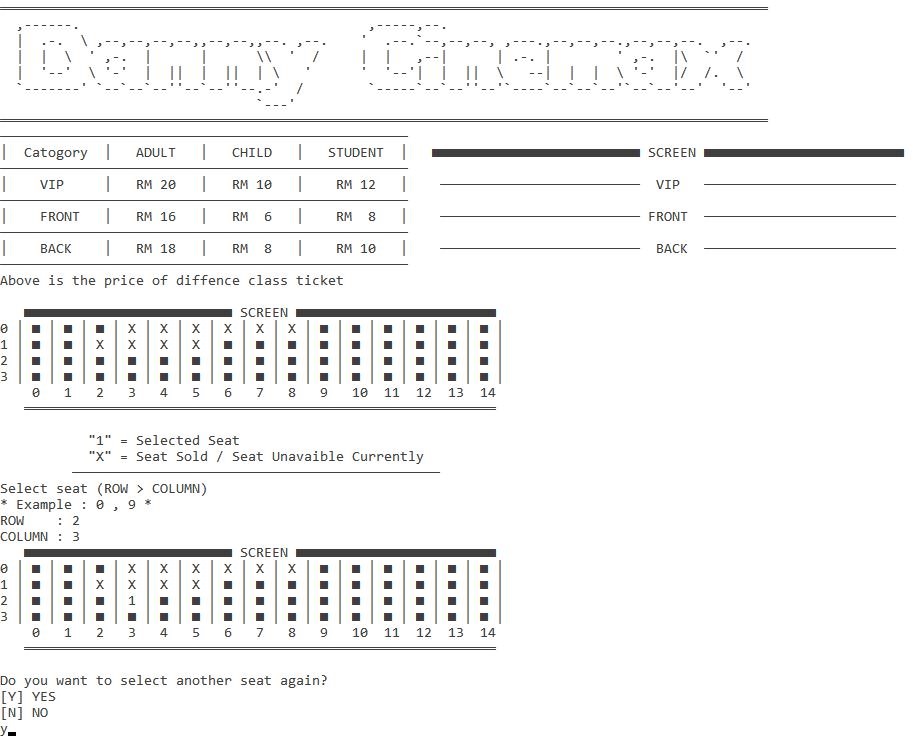
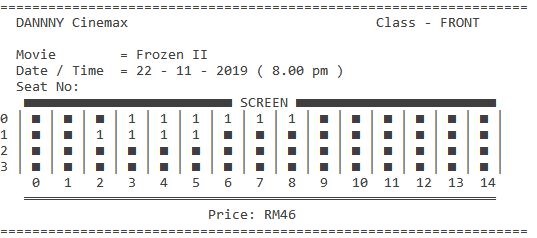
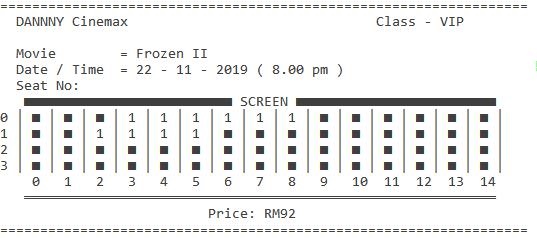
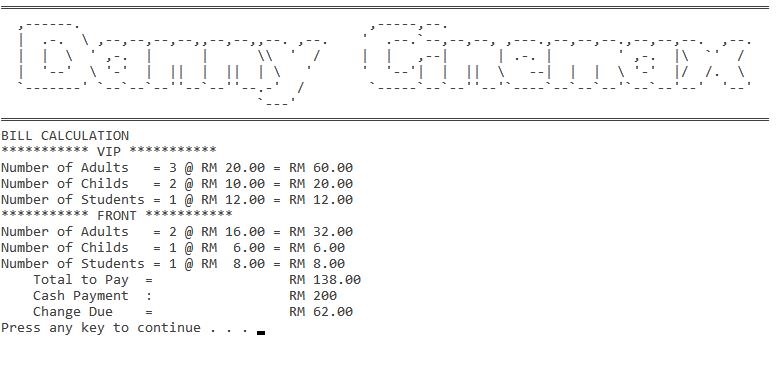
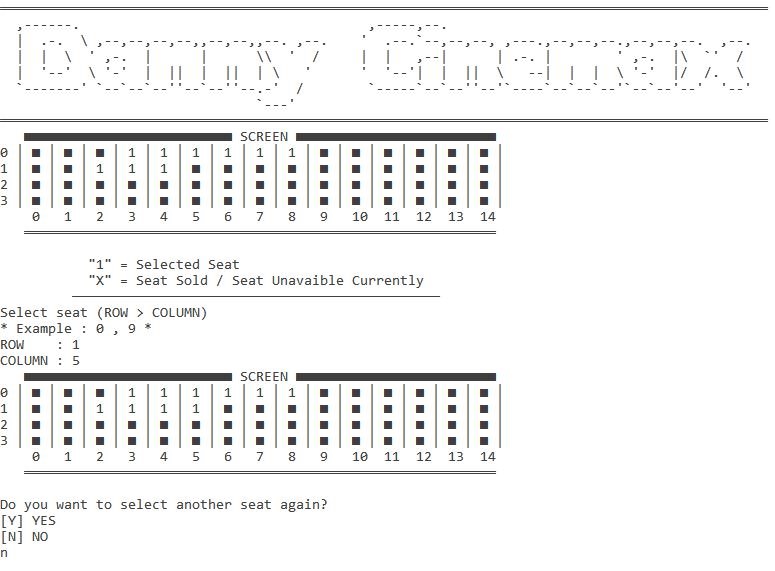
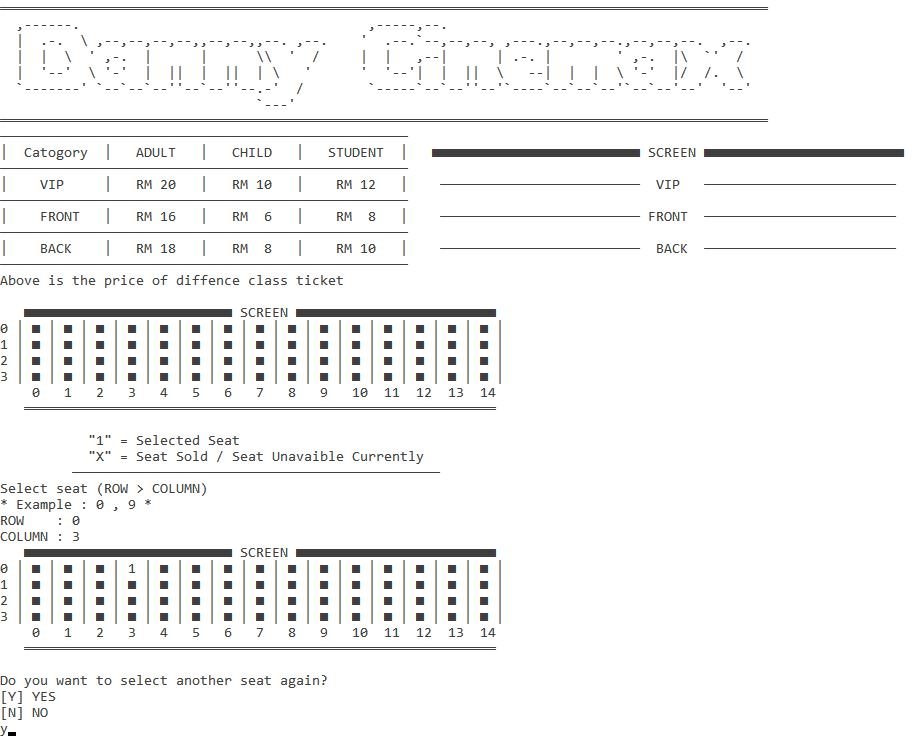
**Program Testing & Output**

Scenario 1:

\*one class is chosen twice (class 0)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Inputs** | | | **Expected Results / Outputs** | | | | | |
| *Loop* | *Seat Class* | *Adult* | *Child* | *Student* | *Total Seats* | *Seat Numbers* | *Ticket Charges* | | | |
| *Adult* | *Child* | *Stud* | *Total* |
| 1 | 0 | 3 | 2 | 1 | 6 | 0,3 – 0,8 | 3\*20 = 60 | 2\*10  =20 | 1\*12  =12 | 92 |
| 1 | 1 | 2 | 1 | 1 | 4 | 1,2 – 1,5 | 2\*16  =32 | 1\*6  =6 | 1\*17  =17 | 55 |
| 2 | 2 | 0 | 2 | 2 | 4 | 2,3 – 2,6 | 0\*18  =0 | 2\*8  =16 | 2\*10  =20 | 36 |
| 2 | 0 | 1 | 0 | 0 | 1 | 0,10 | 1\*20  =20 | 0\*10  =0 | 0\*12  =0 | 20 |
| .. | … | … | … | … |  |  |  |  |  |  |
|  | Totals | 6 | 5 | 4 | 15  Still available:  Class 0: 8  Class 1: 11  Class 2: 26 | | 112 | 42 | 49 | 203 |



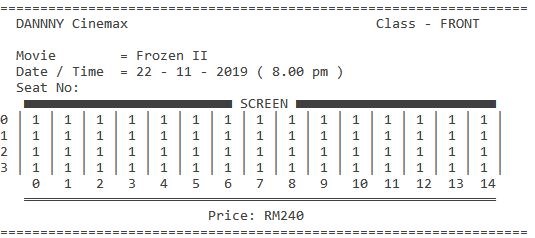
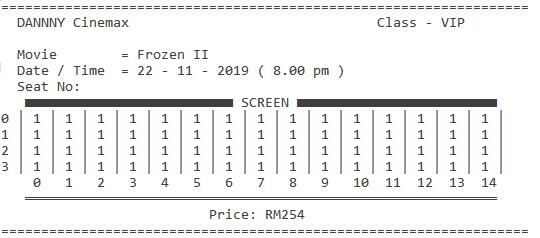
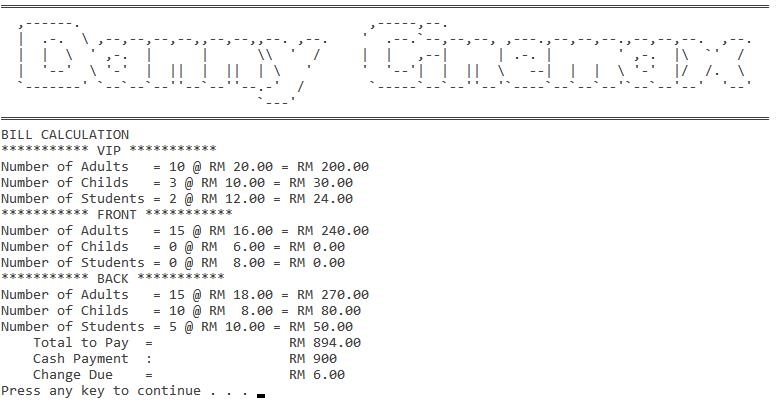
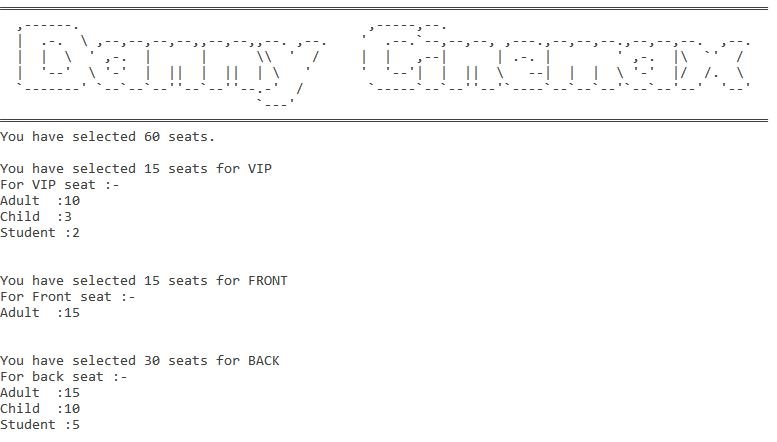
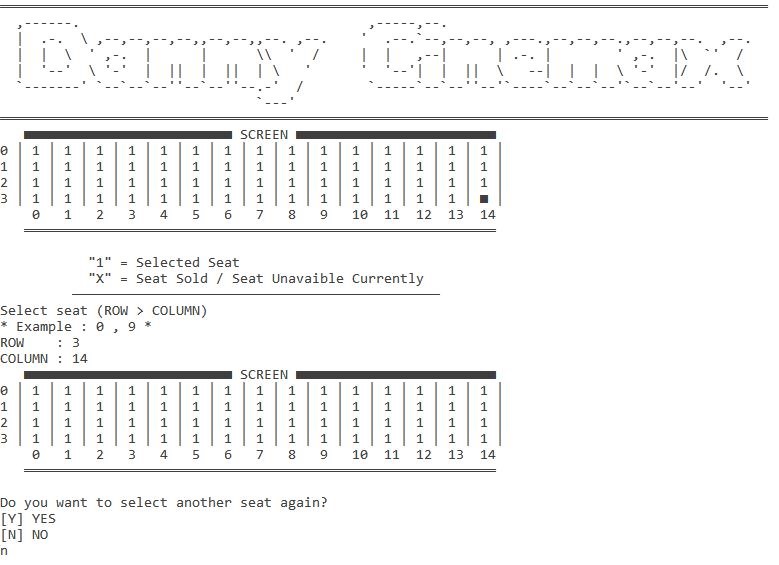
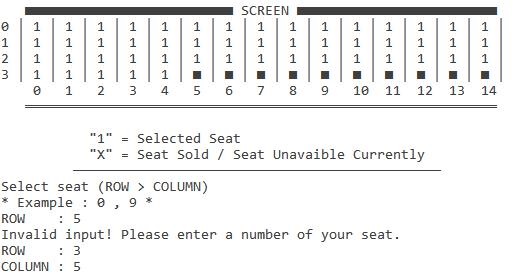
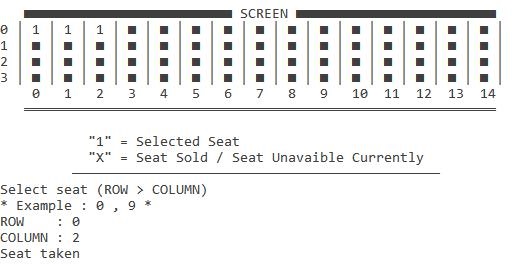
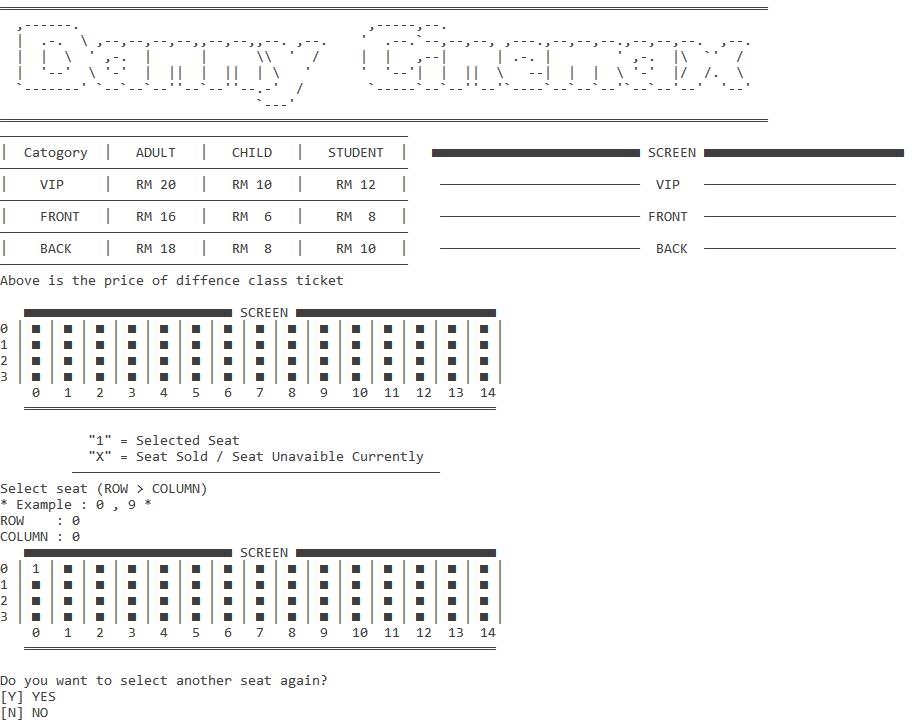


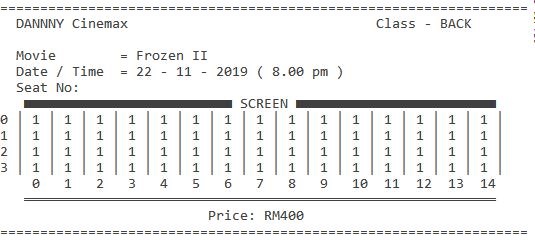
Scenario 2:

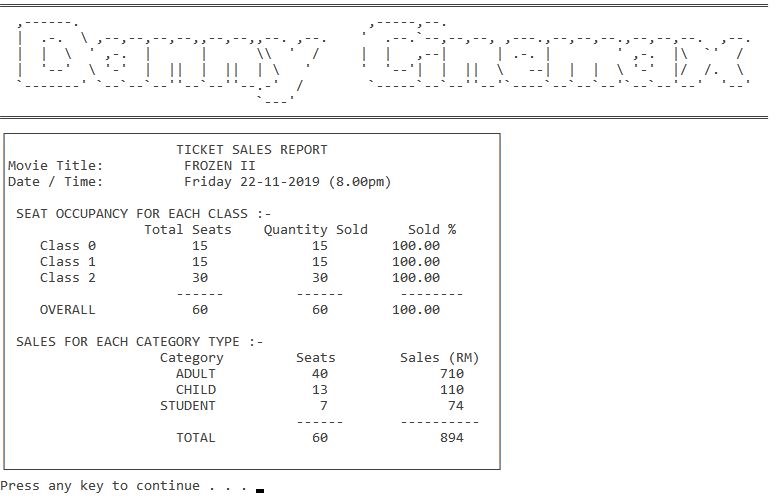
\*No more seats available

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **Inputs** | | | **Expected Results / Outputs** | | | | | |
| *Loop* | *Seat Class* | *Adult* | *Child* | *Student* | *Total Seats* | *Seat Numbers* | *Ticket Charges* | | | |
| *Adult* | *Child* | *Stud* | *Total* |
| 1 | 0 | 10 | 3 | 2 | 15 | 0,0 – 0,14 | 10\*20 = 200 | 3\*10  =30 | 2\*12  =24 | 254 |
| 1 | 1 | 15 | 0 | 0 | 15 | 1,0 – 1,14 | 15\*16  =240 | 0\*6  =0 | 0\*17  =0 | 240 |
| 1 | 2 | 15 | 10 | 5 | 30 | 2,0 – 2,14  3,0 – 3,14 | 15\*18  =270 | 10\*8  =80 | 5\*10  =50 | 400 |
|  | Totals | 40 | 13 | 7 | 60  Still available:  Class 0: 0  Class 1: 0  Class 2: 0 | | 710 | 110 | 74 | 894 |









Scenario 2:

\*Seat is sold out



**Source Code**

#include<iostream>

#include<cmath>

#include<string>

#include<iomanip>

#include<stdlib.h> //use to clear screen

#include<cctype>

#include<limits> //defines elements with the characteristics of arithmetic types

#include<fstream>

using namespace std;

void logo();

void welc();

void movie\_select(int& movie\_num, int& vip2, int& front2, int& back2);

void timetable();

void seat\_display(int& y, int& x, char seat[4][15]);

void seat\_select(int& y, int& x, char seat[4][15]);

void seat\_select\_agn(int& y, int& x, char seat[4][15]);

void check\_class(int& y, int& x, int& vip, int& front, int& back, char seat[4][15]);

void customer\_num(int& vip, int& front, int& back, int& vipadult\_num, int& vipchild\_num, int& vipstudent\_num, int& frontadult\_num, int& frontchild\_num, int& frontstudent\_num, int& backadult\_num, int& backchild\_num, int& backstudent\_num, int& adultsold, int& childsold, int& studentsold);

void ticket\_calculation(int& vip, int& front, int& back, int& vipadult\_num, int& vipchild\_num, int& vipstudent\_num, int& frontadult\_num, int& frontchild\_num, int& frontstudent\_num, int& backadult\_num, int& backchild\_num, int& backstudent\_num, int& total\_vip, int& total\_front, int& total\_back, int& adultprice, int& childprice, int& studentprice, int& vipadult\_ticket2, int& vipchild\_ticket2, int& vipstudent\_ticket2, int& frontadult\_ticket2, int& frontchild\_ticket2, int& frontstudent\_ticket2, int& backadult\_ticket2, int& backchild\_ticket2, int& backstudent\_ticket2);

void ticket\_print(int& movie\_num, int& vip, int& front, int& back, int& total\_vip, int& total\_front, int& total\_back, char seat[4][15]);

void finish\_seat(int& y, int& x, int& vip, int& front, int& back, int& vip2, int& front2, int& back2, char seat[4][15]);

void sale\_report(int& movie\_num, int& vip, int& front, int& back, int& vip2, int& front2, int& back2, int& vipadult\_num, int& vipchild\_num, int& vipstudent\_num, int& frontadult\_num, int& frontchild\_num, int& frontstudent\_num, int& backadult\_num, int& backchild\_num, int& backstudent\_num, int& adultsold, int& childsold, int& studentsold, int& adultprice, int& childprice, int& studentprice);

int main() {

//---declaration

int movie\_num, y, x, seat\_icon3 = 254, vip, front, back,

vipadult\_num, vipchild\_num, vipstudent\_num,

frontadult\_num, frontchild\_num, frontstudent\_num,

backadult\_num, backchild\_num, backstudent\_num,

adultsold, childsold, studentsold,

total\_vip, total\_front, total\_back,

adultprice, childprice, studentprice,

vipadult\_ticket2, vipchild\_ticket2, vipstudent\_ticket2,

frontadult\_ticket2, frontchild\_ticket2, frontstudent\_ticket2,

backadult\_ticket2, backchild\_ticket2, backstudent\_ticket2,

vip2, front2, back2;

char seat[4][15],yes\_no;

string movie\_name[2];

//---functions

for (y = 0; y < 4; y++)

for (x = 0; x < 15; x++)

seat[y][x] = seat\_icon3;

vip = 0;

front = 0;

back = 0;

vip2 = 0;

front2 = 0;

back2 = 0;

adultsold = 0;

childsold = 0;

studentsold = 0;

adultprice = 0;

childprice = 0;

studentprice = 0;

vipadult\_ticket2 = 0;

vipchild\_ticket2 = 0;

vipstudent\_ticket2 = 0;

frontadult\_ticket2 = 0;

frontchild\_ticket2 = 0;

frontstudent\_ticket2 = 0;

backadult\_ticket2 = 0;

backchild\_ticket2 = 0;

backstudent\_ticket2 = 0;

newcustomer:

do

{

welc();

movie\_select(movie\_num, vip2, front2, back2);

timetable();

seat\_select\_agn(y, x, seat);

check\_class(y, x, vip, front, back, seat);

customer\_num(vip, front, back, vipadult\_num, vipchild\_num, vipstudent\_num, frontadult\_num, frontchild\_num, frontstudent\_num, backadult\_num, backchild\_num, backstudent\_num, adultsold, childsold, studentsold);

ticket\_calculation(vip, front, back, vipadult\_num, vipchild\_num, vipstudent\_num, frontadult\_num, frontchild\_num, frontstudent\_num, backadult\_num, backchild\_num, backstudent\_num, total\_vip, total\_front, total\_back, adultprice, childprice, studentprice, vipadult\_ticket2, vipchild\_ticket2, vipstudent\_ticket2, frontadult\_ticket2, frontchild\_ticket2, frontstudent\_ticket2, backadult\_ticket2, backchild\_ticket2, backstudent\_ticket2);

ticket\_print(movie\_num, vip, front, back, total\_vip, total\_front, total\_back, seat);

finish\_seat(y, x, vip, front, back, vip2, front2, back2, seat);

do

{

cout << "Next customer?\n[Y] YES\n[N] NO\n";

cin >> yes\_no;

yes\_no = toupper(yes\_no);

if (yes\_no == 'Y')

{

system("cls");

goto newcustomer;

}

else if (yes\_no == 'N')

{

sale\_report(movie\_num, vip, front, back, vip, front, back, vipadult\_num, vipchild\_num, vipstudent\_num, frontadult\_num, frontchild\_num, frontstudent\_num, backadult\_num, backchild\_num, backstudent\_num, adultsold, childsold, studentsold, adultprice, childprice, studentprice);

}

else

{

cout << "Invalid input!\n";

}

} while (yes\_no != 'Y' && yes\_no != 'N');

} while (yes\_no != 'Y' && yes\_no != 'N');

return 0;

}

void welc() {

logo();

cout << endl << endl;

for (int line = 0; line <= 30; line++)

{

cout << " ";

}

for (int line = 40; line <= 65; line++)

{

cout << (char)205;

}

cout << endl << endl;

cout << setfill(' ') << setw(59) << " WELCOME TO DANNY CINEMAX \n\n";

for (int line = 0; line <= 30; line++)

{

cout << " ";

}

for (int line = 40; line <= 65; line++)

{

cout << (char)205;

}

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

system("pause");

}

void logo() {

system("cls");

for (int w = 1; w <= 96; w++)

{

cout << (char)205;

}

cout << endl;

cout << " ,------. ,-----,--. \n";

cout << " | .-. \\ ,--,--,--,--,,--,--,,--. ,--. ' .--.`--,--,--, ,---.,--,--,--.,--,--,--. ,--. \n";

cout << " | | \\ ' ,-. | | \\\\ ' / | | ,--| | .-. | ' ,-. |\\ `' / \n";

cout << " | '--' \\ '-' | || | || | \\ ' ' '--'| | || \\ --| | | \\ '-' |/ /. \\ \n";

cout << " `-------' `--`--`--''--`--''--.-' / `-----`--`--''--'`----`--`--`--'`--`--'--' '--' \n";

cout << " `---' \n";

for (int w = 1; w <= 96; w++)

{

cout << (char)205;

}

cout << endl;

}

void movie\_select(int& movie\_num, int& vip2, int& front2, int& back2) {

string movie\_name[2] = { "FROZEN II", " Avenger 4: EndGame" };

logo();

cout << "~~~~~~~~Movie~~~~~~~~~\n"; //display the movie that can let user to select

cout << " 1. Frozen II \n";

cout << " 2. Avenger 4 EndGame\n";

cout << "~~~~~~~~~~~~~~~~~~~~~~\n\n";

cout << "Seats available \n";

for (int i = 0; i <= 15; i++)

{

cout << (char)205 ;

}

cout << endl;

cout << "VIP - " << (15 - vip2) << endl;

cout << "FRONT - " << (15 - front2) << endl;

cout << "BACK - " << (30 - back2) << endl;

if ((vip2 + front2 + back2) == 60)

{

cout << "Seeat is sold out, please come back next time...\n\n\n\n";

exit(0);

}

cout << endl;

do //do the looping of input the selection of movie when user inputed wrong

{

cout << "Please Pick a Movie : ";

cin >> movie\_num;

if (cin.fail())

{

cin.clear();

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

}

switch (movie\_num)

{

case (1):

cout << "Your picked movie is " << movie\_name[movie\_num - 1] << endl;

break;

case (2):

cout << "Your picked movie is " << movie\_name[movie\_num - 1] << endl;

break;

default:

cout << "Invalid Number! Please keyin again!\n";

break;

}

} while (movie\_num != 1 && movie\_num != 2);

system("pause");

cout << "\033[2J\033[1;1H"; //clear screen

}

void timetable() {

system("cls");

logo();

for (int line = 0; line <= 50; line++)

{

cout << (char)196;

}

cout << endl;

cout << (char)179 << " Catogory " << (char)179 << " ADULT " << (char)179 << " CHILD " << (char)179 << " STUDENT " << (char)179 ;

cout << " "; //screen

for (int screen = 0; screen <= 25; screen++)

{

cout << (char)254;

}

cout << " SCREEN ";

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)254;

}

cout << endl;

for (int line = 0; line <= 50; line++)

{

cout << (char)196;

}

cout << endl;

cout << (char)179 << " VIP " << (char)179 << " RM 20 " << (char)179 << " RM 10 " << (char)179 << " RM 12 " << (char)179 ;

cout << " "; //vip seat

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)196;

}

cout << " VIP ";

for (int screen = 0; screen <= 23; screen++)

{

cout << (char)196;

}

cout << endl;

for (int line = 0; line <= 50; line++)

{

cout << (char)196;

}

cout << endl;

cout << (char)179 << " FRONT " << (char)179 << " RM 16 " << (char)179 << " RM 6 " << (char)179 << " RM 8 " << (char)179 ;

cout << " "; //front seat

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)196;

}

cout << " FRONT ";

for (int screen = 0; screen <= 23; screen++)

{

cout << (char)196;

}

cout << endl;

for (int line = 0; line <= 50; line++)

{

cout << (char)196;

}

cout << endl;

cout << (char)179 << " BACK " << (char)179 << " RM 18 " << (char)179 << " RM 8 " << (char)179 << " RM 10 " << (char)179 ;

cout << " "; //back seat

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)196;

}

cout << " BACK ";

for (int screen = 0; screen <= 23; screen++)

{

cout << (char)196;

}

cout << endl;

for (int line = 0; line <= 50; line++)

{

cout << (char)196;

}

cout << endl;

cout << "Above is the price of diffence class ticket\n";

cout << endl;

}

void seat\_display(int& y, int& x, char seat[4][15]) {

//-----display seat

cout << " ";

for (int screen = 0; screen <= 25; screen++)

{

cout << (char)254;

}

cout << " SCREEN ";

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)254;

}

cout << endl;

for (int y = 0; y < 4; y++)

{

cout << y << " " << (char)179 << " " << seat[y][0] << " " << (char)179 << " " << seat[y][1] << " " << (char)179 << " " << seat[y][2] << " " <<

(char)179 << " " << seat[y][3] << " " << (char)179 << " " << seat[y][4] << " " << (char)179 << " " << seat[y][5] << " " <<

(char)179 << " " << seat[y][6] << " " << (char)179 << " " << seat[y][7] << " " << (char)179 << " " << seat[y][8] << " " <<

(char)179 << " " << seat[y][9] << " " << (char)179 << " " << seat[y][10] << " " << (char)179 << " " << seat[y][11] << " " <<

(char)179 << " " << seat[y][12] << " " << (char)179 << " " << seat[y][13] << " " << (char)179 << " " << seat[y][14] << " " <<

(char)179 << " " << endl;

}

cout << " ";

for (int z = 0; z < 10; z++)

{

cout << fixed << setw(4) << z;

}

cout << " ";

for (int z = 10; z < 15; z++)

{

cout << fixed << setw(4) << z;

}

cout << endl;

cout << " ";

for (int w = 1; w <= 59; w++)

{

cout << (char)205;

}

cout << endl << endl;

}

void class\_display(){

//-----display the different of classes

cout << " "; //screen

for (int screen = 0; screen <= 25; screen++)

{

cout << (char)254;

}

cout << " SCREEN ";

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)254;

}

cout << endl;

cout << " "; //vip seat

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)196;

}

cout << " VIP ";

for (int screen = 0; screen <= 23; screen++)

{

cout << (char)196;

}

cout << endl;

cout << " "; //front seat

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)196;

}

cout << " FRONT ";

for (int screen = 0; screen <= 23; screen++)

{

cout << (char)196;

}

cout << endl;

cout << " "; //back seat

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)196;

}

cout << " BACK ";

for (int screen = 0; screen <= 23; screen++)

{

cout << (char)196;

}

cout << endl;

}

void seat\_select(int& y, int& x, char seat[4][15]) {

cout << "Select seat (ROW > COLUMN)\n\* Example : 0 , 9 \* \n";

cout << "ROW : ";

cin >> y;

while (cin.fail())

{

cin.clear(y);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "ROW : ";

cin >> y;

}

while (y < 0 || y > 3)

{

cout << "Invalid input! Please enter a number of your seat.\n";

cout << "ROW : ";

cin >> y;

}

cout << "COLUMN : ";

cin >> x;

while (cin.fail())

{

cin.clear(x);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "COLUMN : ";

cin >> x;

}

while (x < 0 || x > 14)

{

cout << "Invalid input! Please enter a number of your seat.\n";

cout << "COLUMN : ";

cin >> x;

}

if (seat[y][x] != '1' && seat[y][x] != 'X')

{

seat[y][x] = '1';

}

else

{

cout << "Seat taken\n";

cout << endl;

}

seat\_display(y, x, seat);

}

void seat\_select\_agn(int& y, int& x, char seat[4][15]) {

char seat\_agn;

do

{

seat\_display(y, x, seat);

cout << " \"1\" = Selected Seat\n \"X\" = Seat Sold / Seat Unavaible Currently\n ";

for (int screen = 0; screen <= 45; screen++)

{

cout << (char)196;

}

cout << endl;

seat\_select(y, x, seat);

cout << "Do you want to select another seat again?\n";

cout << "[Y] YES\n[N] NO\n";

cin >> seat\_agn;

seat\_agn = toupper(seat\_agn);

if (seat\_agn == 'Y')

{

system("cls");

logo();

seat\_select\_agn(y, x, seat);

}

if (seat\_agn != 'Y' && seat\_agn != 'N')

{

cout << "Invalid! \nPlease keyin a correct Alphabet!\n";

}

} while (seat\_agn != 'Y' && seat\_agn != 'N');

}

void check\_class(int& y, int& x, int& vip, int& front, int& back, char seat[4][15]) {

int total\_seat\_select;

vip = 0;

front = 0;

back = 0;

for (y = 0; y < 1; y++)

{

for (x = 0; x <= 14; x++)

{

if (seat[y][x] == '1' && seat[y][x] != 'X')

{

++vip;

}

}

}

for (y = 1; y < 2; y++)

{

for (x = 0; x <= 14; x++)

{

if (seat[y][x] == '1' && seat[y][x] != 'X')

{

++front;

}

}

}

for (y = 2; y < 4; y++)

{

for (x = 0; x <= 14; x++)

{

if (seat[y][x] == '1' && seat[y][x] != 'X')

{

++back;

}

}

}

total\_seat\_select = vip + back + front;

system("cls");

logo();

if (total\_seat\_select == 1)

{

cout << "You have selected " << total\_seat\_select << " seat.\n";

}

else

{

cout << "You have selected " << total\_seat\_select << " seats.\n";

}

}

void customer\_num

(int& vip, int& front, int& back, int& vipadult\_num, int& vipchild\_num, int& vipstudent\_num, int& frontadult\_num, int& frontchild\_num, int& frontstudent\_num, int& backadult\_num, int& backchild\_num, int& backstudent\_num, int& adultsold, int& childsold, int& studentsold)

{

vipadult\_num = 0, frontadult\_num = 0, backadult\_num = 0;

vipchild\_num = 0, frontchild\_num = 0, backchild\_num = 0;

vipstudent\_num = 0, frontstudent\_num = 0, backstudent\_num = 0;

cout << endl;

//-----user enter number seat of vip class

if (vip != 0)

{

do

{

cout << "You have selected " << vip << " seats for VIP\n";

cout << "For VIP seat :-\n";

cout << "Adult :";

cin >> vipadult\_num;

while (cin.fail())

{

cin.clear(vipadult\_num);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "Adult :";

cin >> vipadult\_num;

}

if (vipadult\_num == vip)

{

break;

}

cout << "Child :";

cin >> vipchild\_num;

while (cin.fail())

{

cin.clear(vipchild\_num);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "Adult :";

cin >> vipchild\_num;

}

if ((vipadult\_num + vipchild\_num) == vip)

{

break;

}

cout << "Student :";

cin >> vipstudent\_num;

while (cin.fail())

{

cin.clear(vipstudent\_num);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "Adult :";

cin >> vipstudent\_num;

}

if ((vipadult\_num + vipchild\_num + vipstudent\_num) == vip)

{

break;

}

} while ((vipadult\_num + vipchild\_num + vipstudent\_num) != vip);

cout << endl << endl;

}

//-----user enter number seat of front class

if (front != 0)

{

do

{

cout << "You have selected " << front << " seats for FRONT\n";

cout << "For Front seat :-\n";

cout << "Adult :";

cin >> frontadult\_num;

while (cin.fail())

{

cin.clear(frontadult\_num);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "Adult :";

cin >> frontadult\_num;

}

if (frontadult\_num == front)

{

break;

}

cout << "Child :";

cin >> frontchild\_num;

while (cin.fail())

{

cin.clear(frontchild\_num);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "Adult :";

cin >> frontchild\_num;

}

if ((frontadult\_num + frontchild\_num) == front)

{

break;

}

cout << "Student :";

cin >> frontstudent\_num;

while (cin.fail())

{

cin.clear(frontstudent\_num);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "Adult :";

cin >> frontstudent\_num;

}

if ((frontadult\_num + frontchild\_num + frontstudent\_num) == front)

{

break;

}

} while ((frontadult\_num + frontchild\_num + frontstudent\_num) != front);

cout << endl << endl;

}

//-----user enter number seat of back class

if (back != 0)

{

do

{

cout << "You have selected " << back << " seats for BACK\n";

cout << "For back seat :-\n";

cout << "Adult :";

cin >> backadult\_num;

while (cin.fail())

{

cin.clear(backadult\_num);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "Adult :";

cin >> backadult\_num;

}

if (backadult\_num == back)

{

break;

}

cout << "Child :";

cin >> backchild\_num;

while (cin.fail())

{

cin.clear(backchild\_num);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "Adult :";

cin >> backchild\_num;

}

if ((backadult\_num + backchild\_num) == back)

{

break;

}

cout << "Student :";

cin >> backstudent\_num;

while (cin.fail())

{

cin.clear(backstudent\_num);

cin.ignore(numeric\_limits < streamsize >::max(), '\n');

cout << "Please only enter NUMBER!\n";

cout << "Adult :";

cin >> backstudent\_num;

}

if ((backadult\_num + backchild\_num + backstudent\_num) == back)

{

break;

}

}while ((backadult\_num + backchild\_num + backstudent\_num) != back);

cout << endl << endl;

}

adultsold += vipadult\_num + frontadult\_num + backadult\_num;

childsold += vipchild\_num + frontchild\_num + backchild\_num;

studentsold += vipstudent\_num + frontstudent\_num + backstudent\_num;

system("cls");

}

void ticket\_calculation

(int& vip, int& front, int& back, int& vipadult\_num, int& vipchild\_num, int& vipstudent\_num, int& frontadult\_num, int& frontchild\_num, int& frontstudent\_num, int& backadult\_num, int& backchild\_num, int& backstudent\_num, int& total\_vip, int& total\_front, int& total\_back, int& adultprice, int& childprice, int& studentprice, int& vipadult\_ticket2, int& vipchild\_ticket2, int& vipstudent\_ticket2, int& frontadult\_ticket2, int& frontchild\_ticket2, int& frontstudent\_ticket2, int& backadult\_ticket2, int& backchild\_ticket2, int& backstudent\_ticket2)

{

int adult\_ticket, child\_ticket, student\_ticket, total\_to\_pay;

double cash\_payment, change\_due;

total\_vip = 0;

total\_front = 0;

total\_back = 0;

do

{

logo();

cout << "BILL CALCULATION\n";

//----------------------for class vip------------------------------//

if (vip != 0)

{

adult\_ticket = 20;

child\_ticket = 10;

student\_ticket = 12;

vipadult\_ticket2 = vipadult\_num \* adult\_ticket;

vipchild\_ticket2 = vipchild\_num \* child\_ticket;

vipstudent\_ticket2 = vipstudent\_num \* student\_ticket;

cout << "\*\*\*\*\*\*\*\*\*\*\* VIP \*\*\*\*\*\*\*\*\*\*\*\n";

//-------calculation of adult ticket

cout << "Number of Adults" << fixed << setw(5) << " = " << vipadult\_num;

cout << " @ " << "RM " << fixed << setw(5) << fixed << setprecision(2) << (double)adult\_ticket;

cout << " = " << "RM " << fixed << setprecision(2) << (double)vipadult\_ticket2;

cout << endl;

//------calculation of child ticket

cout << "Number of Childs" << fixed << setw(5) << " = " << vipchild\_num;

cout << " @ " << "RM " << fixed << setw(5) << fixed << setprecision(2) << (double)child\_ticket;

cout << " = " << "RM " << fixed << setprecision(2) << (double)vipchild\_ticket2;

cout << endl;

//------calculation of student ticket

cout << "Number of Students" << fixed << setw(1) << " = " << vipstudent\_num;

cout << " @ " << "RM " << fixed << setw(5) << fixed << setprecision(2) << (double)student\_ticket;

cout << " = " << "RM " << fixed << setprecision(2) << (double)vipstudent\_ticket2;

cout << endl;

total\_vip = vipadult\_ticket2 + vipchild\_ticket2 + vipstudent\_ticket2;

}

//----------------------------for class front-----------------------//

if (front != 0)

{

adult\_ticket = 16;

child\_ticket = 6;

student\_ticket = 8;

frontadult\_ticket2 = frontadult\_num \* adult\_ticket;

frontchild\_ticket2 = frontchild\_num \* child\_ticket;

frontstudent\_ticket2 = frontstudent\_num \* student\_ticket;

cout << "\*\*\*\*\*\*\*\*\*\*\* FRONT \*\*\*\*\*\*\*\*\*\*\*\n";

//-------calculation of adult ticket

cout << "Number of Adults" << fixed << setw(5) << " = " << frontadult\_num;

cout << " @ " << "RM " << fixed << setw(5) << fixed << setprecision(2) << (double)adult\_ticket;

cout << " = " << "RM " << fixed << setprecision(2) << (double)frontadult\_ticket2;

cout << endl;

//------calculation of child ticket

cout << "Number of Childs" << fixed << setw(5) << " = " << frontchild\_num;

cout << " @ " << "RM " << fixed << setw(5) << fixed << setprecision(2) << (double)child\_ticket;

cout << " = " << "RM " << fixed << setprecision(2) << (double)frontchild\_ticket2;

cout << endl;

//------calculation of student ticket

cout << "Number of Students" << fixed << setw(1) << " = " << frontstudent\_num;

cout << " @ " << "RM " << fixed << setw(5) << fixed << setprecision(2) << (double)student\_ticket;

cout << " = " << "RM " << fixed << setprecision(2) << (double)frontstudent\_ticket2;

cout << endl;

total\_front = frontadult\_ticket2 + frontchild\_ticket2 + frontstudent\_ticket2;

}

//------------------------for class back-------------------------------//

if (back != 0)

{

adult\_ticket = 18;

child\_ticket = 8;

student\_ticket = 10;

backadult\_ticket2 = backadult\_num \* adult\_ticket;

backchild\_ticket2 = backchild\_num \* child\_ticket;

backstudent\_ticket2 = backstudent\_num \* student\_ticket;

cout << "\*\*\*\*\*\*\*\*\*\*\* BACK \*\*\*\*\*\*\*\*\*\*\*\n";

//-------calculation of adult ticket

cout << "Number of Adults" << fixed << setw(5) << " = " << backadult\_num;

cout << " @ " << "RM " << fixed << setw(5) << fixed << setprecision(2) << (double)adult\_ticket;

cout << " = " << "RM " << fixed << setprecision(2) << (double)backadult\_ticket2;

cout << endl;

//------calculation of child ticket

cout << "Number of Childs" << fixed << setw(5) << " = " << backchild\_num;

cout << " @ " << "RM " << fixed << setw(5) << fixed << setprecision(2) << (double)child\_ticket;

cout << " = " << "RM " << fixed << setprecision(2) << (double)backchild\_ticket2;

cout << endl;

//------calculation of student ticket

cout << "Number of Students" << fixed << setw(1) << " = " << backstudent\_num;

cout << " @ " << "RM " << fixed << setw(5) << fixed << setprecision(2) << (double)student\_ticket;

cout << " = " << "RM " << fixed << setprecision(2) << (double)backstudent\_ticket2;

cout << endl;

total\_back = backadult\_ticket2 + backchild\_ticket2 + backstudent\_ticket2;

}

//------calculation of total payment

total\_to\_pay = total\_vip + total\_front + total\_back;

cout << fixed << setw(20) << "Total to Pay = " << fixed << setw(19) << "RM " << fixed << setprecision(2) << (double)total\_to\_pay;

cout << endl;

//------user input the cash amount

cout << fixed << setw(20) << "Cash Payment : " << fixed << setw(19) << "RM ";

cin >> cash\_payment;

if (cash\_payment < total\_to\_pay)

{

cout << "Your amount is not valid! Please enter a correct amount.\n";

system("pause");

cout << "\033[2J\033[1;1H";

}

} while (cash\_payment < total\_to\_pay);

//------calculation of change due

change\_due = (double)cash\_payment - total\_to\_pay;

cout << fixed << setw(20) << "Change Due = " << fixed << setw(19) << "RM " << fixed << setprecision(2) << (double)change\_due;

cout << endl;

adultprice = vipadult\_ticket2 + frontadult\_ticket2 + backadult\_ticket2;

childprice = vipchild\_ticket2 + frontchild\_ticket2 + backchild\_ticket2;

studentprice = vipstudent\_ticket2 + frontstudent\_ticket2 + backstudent\_ticket2;

system("pause");

}

void ticket\_print(int& movie\_num, int& vip, int& front, int& back, int& total\_vip, int& total\_front, int& total\_back, char seat[4][15]) {

system("cls");

//------------------------------print ticket for VIP----------------------------//

if (vip != 0)

{

for (int t = 1; t <= vip; t++)

{

for (int m = 0; m <= 65; m++)

{

cout << "=";

}

cout << "\n DANNNY Cinemax";

for (int s = 0; s <= 30; s++)

{

cout << " ";

}

cout << "Class - VIP";

cout << endl << endl;

cout << " Movie" << fixed << setw(10) << " = " << fixed << setw(5);

switch (movie\_num)

{

case (1):

cout << "Frozen II\n";

break;

case (2):

cout << "Avenger 4 EndGame\n";

break;

}

cout << " Date / Time " << fixed << setw(3) << " = " << fixed << setw(5) << "22 - 11 - 2019 ( 8.00 pm )";

cout << "\n Seat No: \n";

cout << " ";

for (int screen = 0; screen <= 25; screen++)

{

cout << (char)254;

}

cout << " SCREEN ";

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)254;

}

cout << endl;

for (int y = 0; y < 4; y++)

{

cout << y << " " << (char)179 << " " << seat[y][0] << " " << (char)179 << " " << seat[y][1] << " " << (char)179 << " " << seat[y][2] << " " <<

(char)179 << " " << seat[y][3] << " " << (char)179 << " " << seat[y][4] << " " << (char)179 << " " << seat[y][5] << " " <<

(char)179 << " " << seat[y][6] << " " << (char)179 << " " << seat[y][7] << " " << (char)179 << " " << seat[y][8] << " " <<

(char)179 << " " << seat[y][9] << " " << (char)179 << " " << seat[y][10] << " " << (char)179 << " " << seat[y][11] << " " <<

(char)179 << " " << seat[y][12] << " " << (char)179 << " " << seat[y][13] << " " << (char)179 << " " << seat[y][14] << " " <<

(char)179 << " " << endl;

}

cout << " ";

for (int z = 0; z < 10; z++)

{

cout << fixed << setw(4) << z;

}

cout << " ";

for (int z = 10; z < 15; z++)

{

cout << fixed << setw(4) << z;

}

cout << endl;

cout << " ";

for (int w = 1; w <= 59; w++)

{

cout << (char)205;

}

cout << endl;

cout << setw(35) << fixed << setprecision(2) << "Price: RM" << total\_vip;

cout << endl;

for (int m = 0; m <= 65; m++)

{

cout << "=";

}

cout << endl;

}

}

//----------------------------print ticket for FRONT---------------------------//

if (front != 0)

{

for (int t = 1; t <= front; t++)

{

for (int m = 0; m <= 65; m++)

{

cout << "=";

}

cout << "\n DANNNY Cinemax";

for (int s = 0; s <= 30; s++)

{

cout << " ";

}

cout << "Class - FRONT";

cout << endl << endl;

cout << " Movie" << fixed << setw(10) << " = " << fixed << setw(5);

switch (movie\_num)

{

case (1):

cout << "Frozen II\n";

break;

case (2):

cout << "Avenger 4 EndGame\n";

break;

}

cout << " Date / Time " << fixed << setw(3) << " = " << fixed << setw(5) << "22 - 11 - 2019 ( 8.00 pm )";

cout << "\n Seat No: \n";

cout << " ";

for (int screen = 0; screen <= 25; screen++)

{

cout << (char)254;

}

cout << " SCREEN ";

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)254;

}

cout << endl;

for (int y = 0; y < 4; y++)

{

cout << y << " " << (char)179 << " " << seat[y][0] << " " << (char)179 << " " << seat[y][1] << " " << (char)179 << " " << seat[y][2] << " " <<

(char)179 << " " << seat[y][3] << " " << (char)179 << " " << seat[y][4] << " " << (char)179 << " " << seat[y][5] << " " <<

(char)179 << " " << seat[y][6] << " " << (char)179 << " " << seat[y][7] << " " << (char)179 << " " << seat[y][8] << " " <<

(char)179 << " " << seat[y][9] << " " << (char)179 << " " << seat[y][10] << " " << (char)179 << " " << seat[y][11] << " " <<

(char)179 << " " << seat[y][12] << " " << (char)179 << " " << seat[y][13] << " " << (char)179 << " " << seat[y][14] << " " <<

(char)179 << " " << endl;

}

cout << " ";

for (int z = 0; z < 10; z++)

{

cout << fixed << setw(4) << z;

}

cout << " ";

for (int z = 10; z < 15; z++)

{

cout << fixed << setw(4) << z;

}

cout << endl;

cout << " ";

for (int w = 1; w <= 59; w++)

{

cout << (char)205;

}

cout << endl;

cout << setw(35) << fixed << setprecision(2) << "Price: RM" << total\_front;

cout << endl;

for (int m = 0; m <= 65; m++)

{

cout << "=";

}

cout << endl;

}

}

//---------------------------print ticket for BACK--------------------------//

if (back != 0)

{

for (int t = 1; t <= back; t++)

{

for (int m = 0; m <= 65; m++)

{

cout << "=";

}

cout << "\n DANNNY Cinemax";

for (int s = 0; s <= 30; s++)

{

cout << " ";

}

cout << "Class - BACK";

cout << endl << endl;

cout << " Movie" << fixed << setw(10) << " = " << fixed << setw(5);

switch (movie\_num)

{

case (1):

cout << "Frozen II\n";

break;

case (2):

cout << "Avenger 4 EndGame\n";

break;

}

cout << " Date / Time " << fixed << setw(3) << " = " << fixed << setw(5) << "22 - 11 - 2019 ( 8.00 pm )";

cout << "\n Seat No: \n";

cout << " ";

for (int screen = 0; screen <= 25; screen++)

{

cout << (char)254;

}

cout << " SCREEN ";

for (int screen = 0; screen <= 24; screen++)

{

cout << (char)254;

}

cout << endl;

for (int y = 0; y < 4; y++)

{

cout << y << " " << (char)179 << " " << seat[y][0] << " " << (char)179 << " " << seat[y][1] << " " << (char)179 << " " << seat[y][2] << " " <<

(char)179 << " " << seat[y][3] << " " << (char)179 << " " << seat[y][4] << " " << (char)179 << " " << seat[y][5] << " " <<

(char)179 << " " << seat[y][6] << " " << (char)179 << " " << seat[y][7] << " " << (char)179 << " " << seat[y][8] << " " <<

(char)179 << " " << seat[y][9] << " " << (char)179 << " " << seat[y][10] << " " << (char)179 << " " << seat[y][11] << " " <<

(char)179 << " " << seat[y][12] << " " << (char)179 << " " << seat[y][13] << " " << (char)179 << " " << seat[y][14] << " " <<

(char)179 << " " << endl;

}

cout << " ";

for (int z = 0; z < 10; z++)

{

cout << fixed << setw(4) << z;

}

cout << " ";

for (int z = 10; z < 15; z++)

{

cout << fixed << setw(4) << z;

}

cout << endl;

cout << " ";

for (int w = 1; w <= 59; w++)

{

cout << (char)205;

}

cout << endl;

cout << setw(35) << fixed << setprecision(2) << "Price: RM" << total\_back;

cout << endl;

for (int m = 0; m <= 65; m++)

{

cout << "=";

}

cout << endl;

}

}

system("pause");

}

void finish\_seat(int& y, int& x, int& vip, int& front, int& back, int& vip2, int& front2, int& back2, char seat[4][15]) {

if (vip != 0)

{

for (y = 0; y < 1; y++)

{

for (x = 0; x <= 14; x++)

{

if (seat[y][x] == '1' || seat[y][x] == 'X')

{

seat[y][x] = 'X';

++vip2;

}

}

}

}

if (front != 0)

{

for (y = 1; y < 2; y++)

{

for (x = 0; x <= 14; x++)

{

if (seat[y][x] == '1' || seat[y][x] == 'X')

{

seat[y][x] = 'X';

++front2;

}

}

}

}

if (back != 0)

{

for (y = 2; y < 4; y++)

{

for (x = 0; x <= 14; x++)

{

if (seat[y][x] == '1' || seat[y][x] == 'X')

{

seat[y][x] = 'X';

++back2;

}

}

}

}

}

void sale\_report

(int& movie\_num, int& vip, int& front, int& back, int& vip2, int& front2, int& back2, int& vipadult\_num, int& vipchild\_num, int& vipstudent\_num, int& frontadult\_num, int& frontchild\_num, int& frontstudent\_num, int& backadult\_num, int& backchild\_num, int& backstudent\_num, int& adultsold, int& childsold, int& studentsold, int& adultprice, int& childprice, int& studentprice)

{

int total\_class\_left;

total\_class\_left = vip2 + front2 + back2;

system("cls");

logo();

cout << (char)218;

for (int r = 0; r <= 60; r++)

{

cout << (char)196;

}

cout << (char)191;

cout << endl;

cout << (char)179;

cout << " TICKET SALES REPORT ";

cout << (char)179 << endl << (char)179;

cout << "Movie Title:" << fixed << setw(10) << " ";

if (movie\_num == 1)

{

cout << "FROZEN II";

}

else if (movie\_num == 2)

{

cout << "AVENGER 4 ENDGAME";

}

cout << " " << (char)179;

cout << endl;

cout << (char)179;

cout << "Date / Time:" << fixed << setw(10) << " " << "Friday 22-11-2019 (8.00pm";

cout << ") " << (char)179 << endl;

cout << (char)179 << fixed << setw(62) << (char)179 << endl;

cout << (char)179 << " SEAT OCCUPANCY FOR EACH CLASS :-" << fixed << setw(29) << (char)179 << endl;

cout << (char)179;

cout << fixed << setw(56) << "Total Seats Quantity Sold Sold %" << fixed << setw(6) << (char)179 << endl;

cout << (char)179 << fixed << setw(15) << "Class 0\t\t15" << fixed << setw(15) << vip2 << fixed << setw(14) << (double)((vip2 / 15.00) \* 100) << fixed << setw(8) << (char)179 << endl;

cout << (char)179 << fixed << setw(15) << "Class 1\t\t15" << fixed << setw(15) << front2 << fixed << setw(14) << (double)((front2 / 15.00) \* 100) << fixed << setw(8) << (char)179 << endl;

cout << (char)179 << fixed << setw(15) << "Class 2\t\t30" << fixed << setw(15) << back2 << fixed << setw(14) << (double)((back2 / 30.00) \* 100) << fixed << setw(8) << (char)179 << endl;

cout << (char)179 << fixed << setw(27) << " ------" << fixed << setw(15) << "------" << fixed << setw(15) << "--------" << fixed << setw(5) << (char)179 << endl;

cout << (char)179 << fixed << setw(15) << "OVERALL\t\t60" << fixed << setw(15) << total\_class\_left << fixed << setw(14) << (double)(100 \* (total\_class\_left / 60.00)) << fixed << setw(8) << (char)179 << endl;

cout << (char)179 << fixed << setw(62) << (char)179 << endl;

cout << (char)179 << " SALES FOR EACH CATEGORY TYPE :-" << fixed << setw(30) << (char)179 << endl;

cout << (char)179 << fixed << setw(27) << "Category" << fixed << setw(14) << "Seats" << fixed << setw(18) << "Sales (RM)" << fixed << setw(3) << (char)179 << endl;

cout << (char)179 << fixed << setw(26) << "ADULT" << fixed << setw(14) << adultsold << fixed << setw(17) << adultprice << fixed << setw(5) << (char)179 << endl;

cout << (char)179 << fixed << setw(26) << "CHILD" << fixed << setw(14) << childsold << fixed << setw(17) << childprice << fixed << setw(5) << (char)179 << endl;

cout << (char)179 << fixed << setw(26) << "STUDENT" << fixed << setw(14) << studentsold << fixed << setw(17) << studentprice << fixed << setw(5) << (char)179 << endl;

cout << (char)179 << fixed << setw(42) << "------" << fixed << setw(17) << "----------" << fixed << setw(3) << (char)179 << endl;

cout << (char)179 << fixed << setw(26) << "TOTAL" << fixed << setw(14) << (adultsold + childsold + studentsold) << fixed << setw(17) << (adultprice + childprice + studentprice) << fixed << setw(5) << (char)179 << endl;

cout << (char)179 << fixed << setw(62) << (char)179 << endl;

cout << (char)192;

for (int r = 0; r <= 60; r++)

{

cout << (char)196;

}

cout << (char)217;

cout << endl;

system("pause");

}