Republic of the Philippines
RIZAL TECHNOLOGICAL UNIVERSITY
Cities of Mandaluyong and Pasig
COLLEGE OF ENGINEERING, ARCHITECHTURE AND TECHNOLOGY

COMPUTER PROGRAMMING 1

(ITC111L)

<MOVIE BUYING TICKETS SYSTEM >

GRADE

Submitted by:

Bondad, John Christopher P.
BSIT
CEIT-37-102P

Boragay, Vicente M.
BSIT
CEIT-37-102P

Collado, Jonalyn B.
BSIT
CEIT-37-102P

Malayo, Abegail B.
BSIT
CEIT-37-102P

Manansala, Raniel E.
BSIT
CEIT-37-102P

Melendez, Rafael Lewis R.
BSIT
CEIT-37-102P

Santos. Karl Roi S.
BSIT
CEIT-37-102P

Villanueva. Ron Adrian
BSIT
CEIT-37-102P

I. INTRODUCTION

We create a software program called "Movie Buying Tickets System." The Movie Buying Tickets System is a software program that allows its users and/or potential users to book movie tickets on the self-service kiosk you may find in the cinema. The developers aim to create a software program that enables movie tickets to be more accessible to users and/or potential users. Moreover, the software program formed allows the company to have an efficient product distribution channel, making the ticketing process faster than without the system.

The Movie Buying Tickets System is not one-sided. Both company and target users will gain from it. The software program is cost-efficient for the company. The cost allotted for employees' salaries will lessen because the company will not need to hire more employees. The fewer workers the company pays, the bigger the profit it will get. The software program allows the company to have a new product distribution channel, making the ticketing process faster than before. The Movie Buying Tickets System's target users are those people that love to watch movies in the cinema or, in other words, movie addicts. Through this software program, the target users could acquire the ticket by simply using the self-service kiosk. Aside from this, the software program will lessen the users' burden of not knowing what movie to choose because the list in this system is organized practically.

Features:

- See all movies available
- See how many tickets available per movie and price per ticket
- Search a movie
- Purchase a ticket

II. DESCRIPTION OF THE PROJECT

A simple movie buying ticket system is an easy way for users to purchase tickets. The system allows users to watch all movies, search for available movies and buy tickets. If you want to see all available movies, you have to press "1" to see all movies, available tickets and movie ticket prices. However, if the user wants to search for the desired movie, he just presses "2" and the movie title must type in upper case, but if the movie that the user is looking for is not in the list of available movies, the system will print "Error: This Movie is not available". When purchasing movie tickets, the user simply presses "3" and the system will ask for the title of the movie in upper case, for the number of the tickets they wish to purchase and name of the user. Once the questions/information provided have been answered, the system will print the receipt. This includes your name, the total amount of tickets purchased, and the selected movie and press any key to go back on main menu. However, if the user does not want to continue, press "4" to show the exit the program.

III. OBJECTIVES

The main goal of the study is to provide a system that is simple and effective in selling movie tickets intended specifically to:

1. Organize the list of movies in a practical way.
2. Raise corporate profits by reducing the number of employees needed at the movie theater counter, because it is available all the time, it is also not time-limited. This permits advertisements and promotions to run continuously, which can increase sales.
3. To make it simple for the client to buy the ticket when they are in the area within the theater where the user may easily use the services offered.

IV. SIGNIFICANCE OF THE STUDY

With the advancement of technology, this study seeks to provide a better method of purchasing movie tickets for those who will use it, as they stand to gain the most from the outcome.

Customer. The result of this program will be an advantage for those customers who will be watching movies in the cinemas since they are the ones who will be using and evaluating the program.

Cinema administrators and staff. The concept and development of this study could lay the foundation for a simpler and easier job for those working in cinema administration and its staff.

Students. Students pursuing comparable studies may use this study as a reference and guide.

V. SCOPE AND DELIMITATIONS

This study will be conducted to determine how people can be able to buy movie tickets through an Android kiosk app that focuses on buying tickets. This study will also identify the various selling factors at several malls. Therefore, this study can distinguish some comparisons, different pros and cons of buying movie tickets in malls nationwide between over the counter selling and android kiosk app.

This program can contribute to the rapidly changing society that involves certain innovations in technology as time passes by. Aside from customers, the target of this program is to help malls and cinema to save the budget that is allocated for wages. They can use the app rather than hire an actual person to accommodate customers.

VI. SCREEN OUTPUT

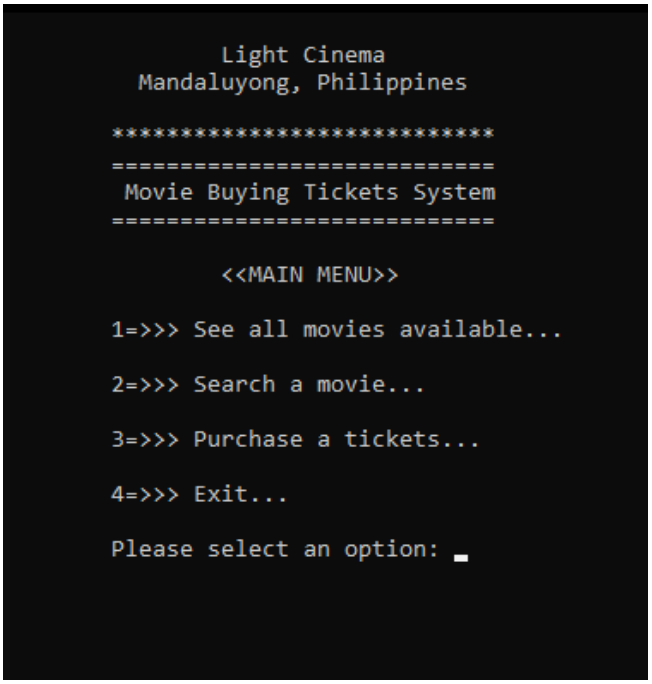
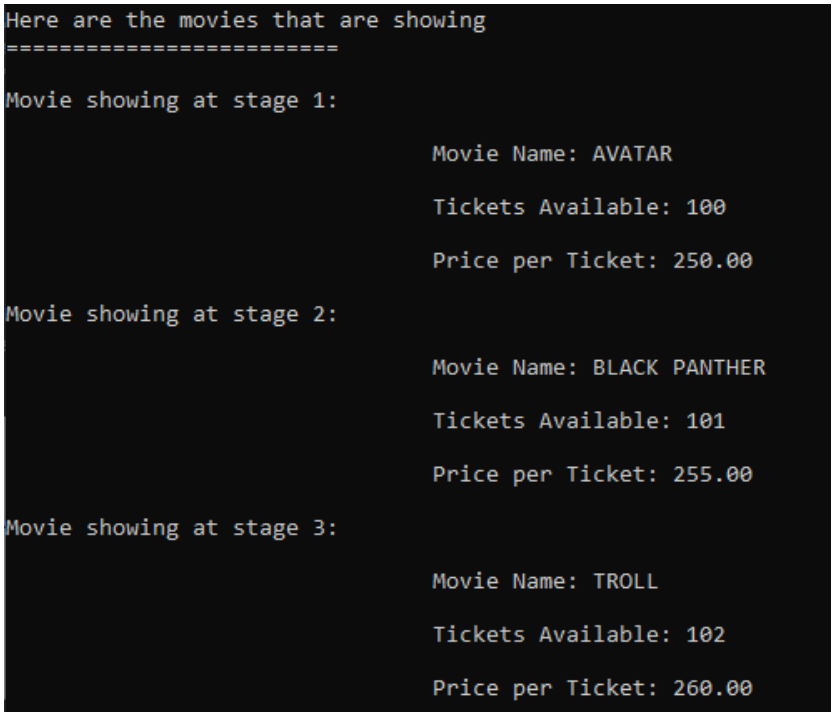


Figure 1: This Figure is to select an option in main menu



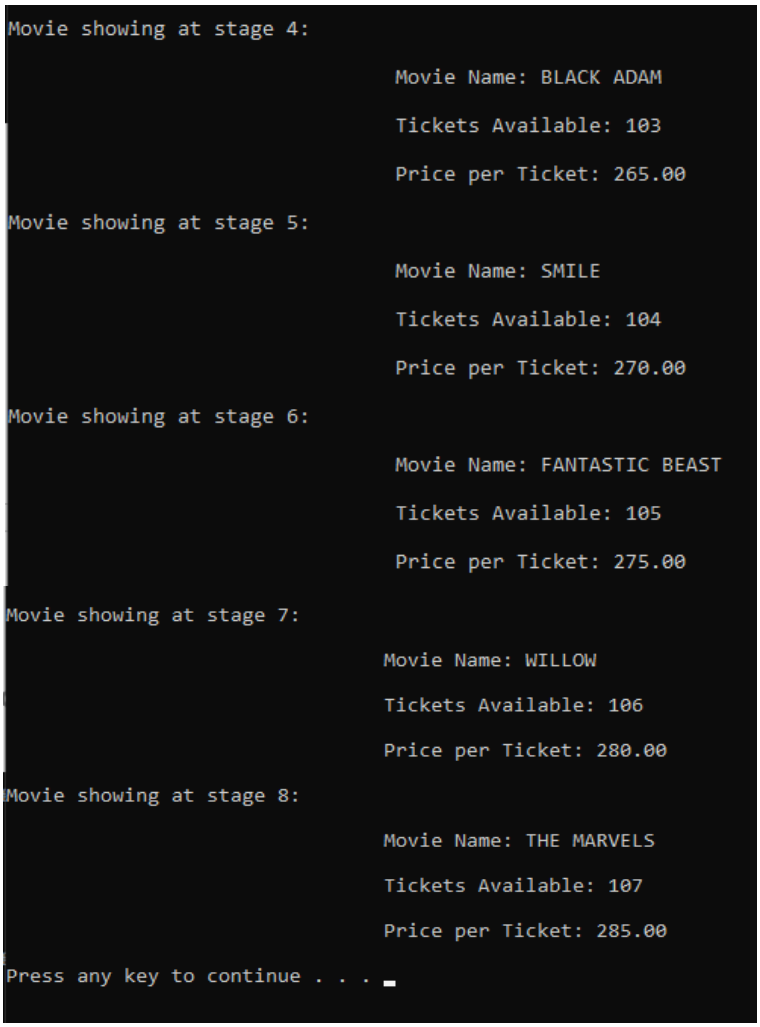


Figure 2: This figure shows the option "1" is to see all the movies available in the cinema; it displays the movie names and how many tickets available and price per ticket.

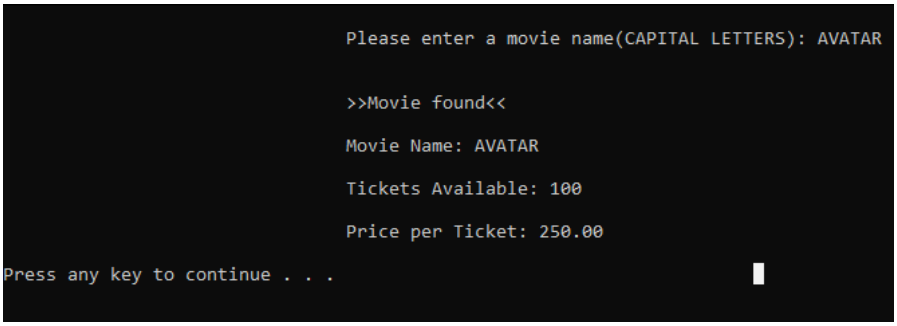


Figure 3: This shows how option 2 works; if the movie you search is in the program it will show "Movie found" and the name of a movie you search and tickets available and the price per ticket.

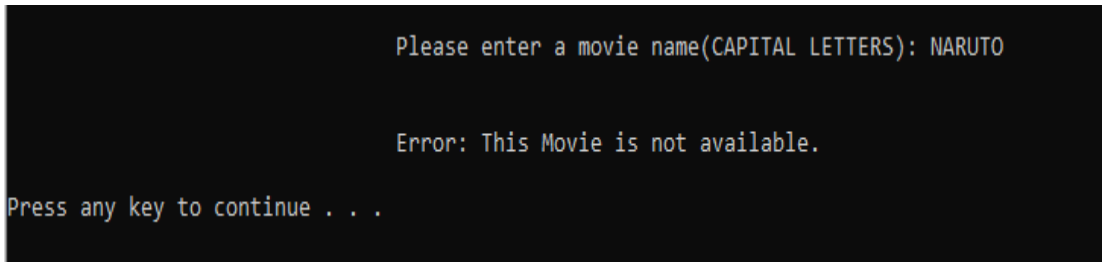


Figure 4: This figure if the movie you search in, not in the program it shows “Error: This Movie is not available.

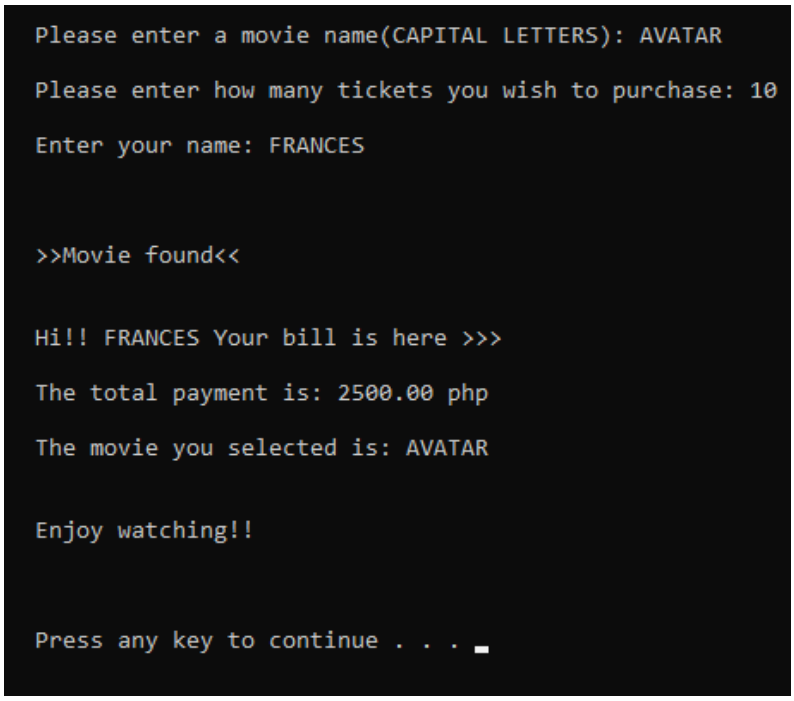


Figure 5: : It shows option 3 or how to purchase a ticket, its required to enter the movie you want to watch and how many tickets to want to purchase and name of the user, then it will show your bill. It shows your name, the total payment, and the movie you selected.

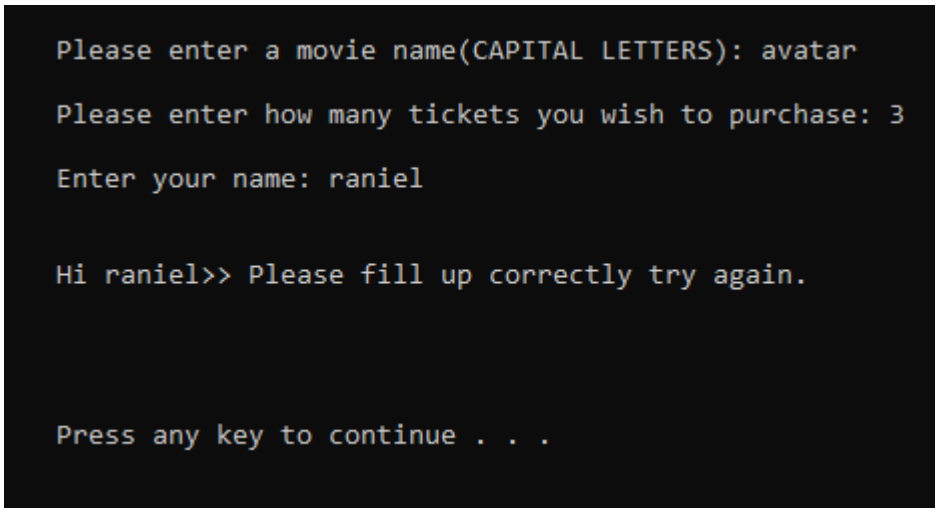


Figure 6: This figure if you purchase a ticket and the movie name you enter is not upper case, it will show your name and it will say “Please fill up correctly, try again”.

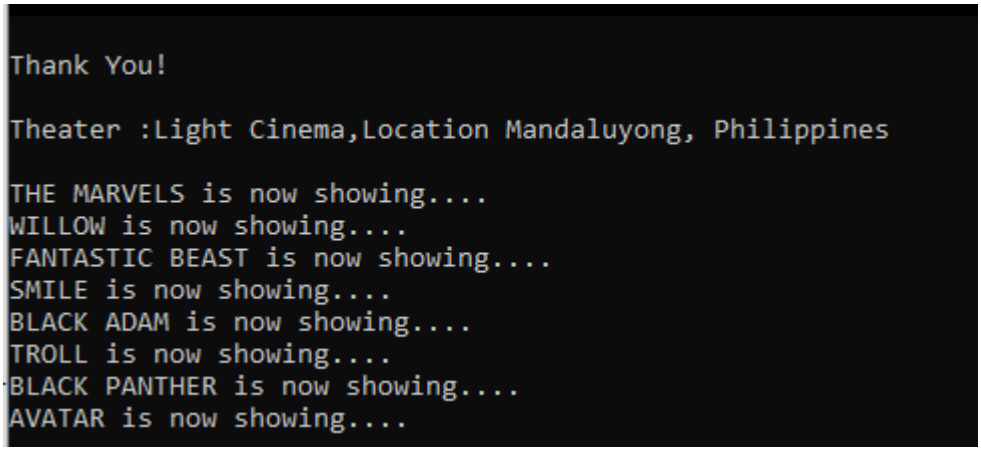


Figure 7: This figure shows the exit display if you press option 4 or to exit the program, it will show the name of the theater, location and the movies that are showing.

VII. SOURCE CODE

```
#include <iostream>
#include <string>
using namespace std;

class Movie
{
private:
	string movie_name;
	int tickets_available;
	double ticket_price;
public:
	Movie();
	Movie(string my_movie_name, int my_tickets_available, double my_ticket_price);
	~Movie();

	string get_movie_name();
	int get_tickets_available();
	double get_ticket_price();
	void set_movie_name(string new_movie_name);
	void set_tickets_available(int new_tickets_available);
	void set_ticket_price(double new_ticket_price);

	double PurchaseTicket(int new_tickets_available);
	void Display();
};


class MovieTicketMain
{
private:
	Movie * p_MovieList;
	string theater_name;
	string theater_location;
public:
	MovieTicketMain();
	MovieTicketMain(string my_theater_name, string my_theater_location);
	~MovieTicketMain();
	void Init(Movie * p_my_MovieList, int array_size);
	void Run();
	void DisplayMainMenu();
	void ViewMovies();
	Movie * SearchMovie();
	Movie * FindMovie(string find_movie_name);
	void PurchaseTicket();
};
```

```
const int Movie_Array = 8;

int main ()
{
	Movie movie_objects[Movie_Array];

	MovieTicketMain * p_MovieTicketMain = new MovieTicketMain ("Light
Cinema","Mandaluyong, Philippines");
	p_MovieTicketMain->Init(movie_objects, Movie_Array);
	p_MovieTicketMain->Run();

	delete p_MovieTicketMain;
	return 0;
}

MovieTicketMain::MovieTicketMain()
{
	p_MovieList =NULL;
}


MovieTicketMain::MovieTicketMain(string my_theater_name, string my_theater_location)
{
	theater_name = my_theater_name;
	theater_location = my_theater_location;
	p_MovieList = NULL;
}


MovieTicketMain::~MovieTicketMain()
{
	cout << "Theater :" << theater_name << ",Location " << theater_location << endl;
	cout << endl;
}

Movie::~Movie()
{
	cout << movie_name << " is now showing...." << endl;
}


void MovieTicketMain::Init(Movie * p_my_MovieList, int array_size)
{
	string movie_titles[Movie_Array] =
	  {"AVATAR",
	   "BLACK PANTHER",
	   "TROLL",
		"BLACK ADAM",
		"SMILE",
```

```
                "FANTASTIC BEAST",
                "WILLOW",
                "THE MARVELS"};

        int theater_tickets[Movie_Array] = {100,101,102,103,104,105,106,107};

        double movie_price[Movie_Array] = {250,255,260,265,270,275,280,285};

        Movie * p = p_my_MovieList;

        for(int i=0;i <= Movie_Array- 1;i++){
                p->set_movie_name(movie_titles[i]);
                p->set_tickets_available(theater_tickets[i]);
                p->set_ticket_price(movie_price[i]);
                p++;
        }

        p_MovieList = p_my_MovieList;

}

void MovieTicketMain::Run()
{
        int option;

        do{

                DisplayMainMenu();

                cout << endl << "\t\t\t\tPlease select an option: ";

                cin >> option;


                switch (option)
                {

                case 1:
                        ViewMovies();
                        system("PAUSE");
                        system("CLS");
                        break;

                case 2:

                        SearchMovie();
                        system("PAUSE");
                        system("CLS");
                        break;
```

```
                case 3:

                        PurchaseTicket();
                        break;

                case 4:
                        system("CLS");

                        cout <<"\nThank You!\n" << endl;
                        break;

                default:
                        system("CLS");

                        cout <<"Please choose a valid option:" << endl
                                << "==============================" << endl;
                        cin.ignore();

                }
        }
        while (option != 4);
}

void MovieTicketMain::DisplayMainMenu()
{
            cout << endl;
                cout<<"\t\t\t\t       " << theater_name << endl;
                cout<<"\t\t\t\t  " << theater_location << endl;
                cout<< endl <<"\t\t\t\t*****************************  " << endl
                <<"\t\t\t\t============================" << endl
                <<"\t\t\t\t Movie Buying Tickets System" << endl
                <<"\t\t\t\t============================\n" << endl
                <<"\t\t\t\t\t<<MAIN MENU>>\n " << endl
                <<"\t\t\t\t1=>>> See all movies available...\n" << endl
                <<"\t\t\t\t2=>>> Search a movie...\n" << endl
                <<"\t\t\t\t3=>>> Purchase a tickets...\n" << endl
                <<"\t\t\t\t4=>>> Exit..." << endl;

}

void MovieTicketMain::ViewMovies()
{

        Movie * p_view_movies = p_MovieList;
   system("CLS");
        cout << endl
                <<"Here are the movies that are showing " << endl
                <<"=========================" << endl << endl;
```

```
        for (int i = 0; i < Movie_Array; i++){
                cout << "Movie showing at stage "  << i + 1  << ": " << endl;
                p_view_movies->Display();
                ++p_view_movies;
        }
}


Movie * MovieTicketMain::SearchMovie()
{
        string search_movie_name;
   system("CLS");
        cout << "\n\t\t\t\tPlease enter a movie name(CAPITAL LETTERS): ";
        cin.sync();
        std::getline(std::cin, search_movie_name);

        Movie * p = FindMovie(search_movie_name);

        if (p != NULL){
                p->Display();
        }
        else {
                cout << endl << "\n\t\t\t\tError: This Movie is not available." << endl << endl;
                return NULL;
        }
        return p;
}


Movie * MovieTicketMain::FindMovie(string find_movie_name)
{

        Movie * p_find_movie = p_MovieList;

        for (int i = 0; i < Movie_Array; i++){
                if (p_find_movie->get_movie_name() == find_movie_name){
                        cout << endl << "\n\t\t\t\t>>Movie found<<" << endl;
                        return p_find_movie;
                }
                else {
                        p_find_movie++;

                }
        }
        return NULL;
}

void Movie::Display()
```

```
{
                cout.setf(ios::fixed, ios::floatfield);
                cout.setf(ios::showpoint);
                cout.precision(2);
                cout << endl << "\t\t\t\tMovie Name: " << movie_name << endl
                        <<"\n\t\t\t\tTickets Available: " << tickets_available << endl
                        <<"\n\t\t\t\tPrice per Ticket: " << ticket_price << endl << endl;

}

void MovieTicketMain::PurchaseTicket()
{
        string purchase_movie_name;
        string firstname;

        int total_tickets;
   system("CLS");
        cout << "\n\n\t\t\t\tPlease enter a movie name(CAPITAL LETTERS): ";
        cin.sync();
        std::getline(std::cin, purchase_movie_name);

        cout << "\n\t\t\t\tPlease enter how many tickets you wish to purchase: ";
        cin >> total_tickets;

        cout <<"\n\t\t\t\tEnter your name: ";
        cin >> firstname;

        cout << endl;

        Movie * p = FindMovie(purchase_movie_name);

                if (p != NULL){
                double total_cost =     p-> PurchaseTicket(total_tickets);
                                if ( total_cost == -1){
                                        cout << "\n\t\t\t\tHi  " << firstname << endl;
                                        cout << "\n\t\t\t\tThis movie has been sold out" << endl
                                                << "\n\t\t\t\tor not have enough tickets" << endl;

                                }
                                else {
                                        cout << endl;

                                        cout << "\n\t\t\t\tHi!! " << firstname <<" Your bill is here
>>>\n";

                                        cout << "\n\t\t\t\tThe total payment is: " << total_cost << "
php" << endl;

                                        cout << "\n\t\t\t\tThe movie you selected is: " <<
```

```
purchase_movie_name << endl;
                                        cout << endl;
                                        cout << "\n\t\t\t\tEnjoy watching!!";
                                }

                                cout<<"\n\n\n\n\t\t\t\t";
                                system("PAUSE");
                                system("CLS");
                        }
                        else {
                                cout << "\n\t\t\t\tHi " << firstname << ">> Please fill up correctly, try again."
<< endl;
                                cout<<"\n\n\n\n\t\t\t\t";
                                system("PAUSE");
                                system("CLS");
                        }

}


Movie::Movie()
{
        movie_name = "";
        tickets_available = 0;
        ticket_price = 0.0;
}


Movie::Movie(string my_movie_name, int my_tickets_available, double my_ticket_price)
{
        movie_name = my_movie_name;
        tickets_available = my_tickets_available;
        ticket_price = my_ticket_price;
}


string Movie::get_movie_name()
{
        return movie_name;
}


int Movie::get_tickets_available()
{
        return tickets_available;
}


double Movie::get_ticket_price()
```

```
{
	return ticket_price;
}


void Movie::set_movie_name(string new_movie_name)
{
	movie_name = new_movie_name;
}


void Movie::set_tickets_available(int new_tickets_available)
{
	tickets_available = new_tickets_available;
}


void Movie::set_ticket_price(double new_ticket_price)
{
	ticket_price = new_ticket_price;
}


double Movie::PurchaseTicket(int purchase_tickets_available)
{
	double total_price;

	if (purchase_tickets_available <= tickets_available) {
		total_price = purchase_tickets_available * ticket_price;
		tickets_available = tickets_available - purchase_tickets_available;
		return total_price;
	} else {
		return 0;
	}
}
```

VIII. CONCLUSION

C++ is a general-purpose programming language with a bias toward system development due to its easy access to hardware-level resources, effective compilation, and adaptable approach to high-level abstractions. When developing a system like a movie ticket buying system that utilizes the C++ programming language, it is crucial to comprehend and put into practice advanced programming ideas because they can make particular tasks simple and convenient. Utilizing the full capacity of the C++ language can assist the user in reducing the tasks involved in computation, encoding, and compilation. We have encountered a lot of problems when it comes to buying tickets recently. So, in this generation, applying our knowledge about our growing technologies can definitely help us develop more high-tech sites and devices, like System Development (Movie ticket selling) that is made of advanced C++ language, thus resulting to be more convenient, user friendly, and hassle-free.

Again, this System Development goal is to employ C++ to make it simple and practicable for users to purchase movie tickets via android kiosks. The android kiosks are the only ones that will serve us to choose a movie that we want, which results in a more convenient way than waiting in line for an hour. The usage of C++ will make the system user-friendly, speed up transactions.

To PROFESSOR:

Final Project Specification:

-Choose any system/game you want to create. Examples: Banking System, Grading System, Library System, etc.

-The program should include all the topics in ITC111/ITC11L.

-The system will be created by group (min of 3 and max of 4 members).

-Deadline will be during the Final Exam Week.

-Submit the Final Project files to this GDrive link: (Note to Prof: Insert Gdrive link here)

-In the GDrive, create a project folder and name it by the surnames of your group members. Your project folder should contain the: (a) Screen record/video of the running program showing the different features of your program (b) Project Documentation

Note: **The template for your project documentation is attached here.**

Note: **Only the group leader will upload the files to the project folder.**