Shree Vallabh Ashram’s

M.G.M. Amin and V.N. Savani School

Killa Pardi



Computer Science Project:

Music Library Management

Aakash Yadav

12th B (Science)

Index

|  |  |  |
| --- | --- | --- |
| Sr. No. | Title | Page No. |
| 1 | Index | 2 |
| 2 | Certificate | 3 |
| 3 | Acknowledgement | 4 |
| 4 | Introduction | 5 |
| 5 | Oop Concepts | 6 |
| 6 | Header Files | 7 |
| 7 | Data Files Used | 9 |
| 8 | Classes & its Members | 10 |
| 9 | Source File | 15 |
| 10 | Glimpses of Program | 47 |
| 11 | Bibliography | 51 |



SHREE VALLABH ASHRAM’S

MGM AMIN & V N SAVANI SCHOOL

NH 8, Killa Pardi, Gujarat 396125

CERTIFICATE

This is to certify that the project work is a bonaffied work done by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of \_\_\_\_\_\_ Roll No. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ for fulfilment of the requirement of practical examination and project in the subject of

Computer Science during 2018-19.

The Project work was carried out under the supervision of Mr. Ashok Bist and certified.

The work recorded here is not a part of any thesis or dissertation or compiled from any other sources.

Subject Teacher Principal

Examiner School Stamp

ACKNOWLEDGEMENT

**The blessings of our Param Poojya Swamiji**

**has always helped us to strive for**

**excellence, ever and always.**

**I would also like to thank our Principal**

**Mr. R. P. Maurya under whose able hand we**

**got the opportunity to display our skills.**

**A thanks to our Vice- Principal Mr. Devendra**

**Singh who’s blessings are with us forever.**

**Not to forget, Mr. Ashok Bist, our Computer**

**Science teacher, without whom, this project**

**could not have been dreamt of.**

**Finally, a special thanks to my Project**

**partner Adam Suratwala, my Family and**

**my Friends who always helped me level up.**

Introduction

Life is one grand, sweet song, so start the music – Ronald Reagan.



Keeping in mind the above statement of former US President, my partner and I created this music library management program. It helps customers to make a rental on their own while maintaining the records of the library.

This program can be used by anyone who visits the library as it is simple, self directing and easy to use.

With some modifications, it can also be used for any library management.

C++ Concepts Used

Primary objective of C++ is OOP, which stands for Object Oriented Programming and it has the following basic concepts:

1. **Data Abstraction** refers to the act of representing essential features without including the background details or explanations.
2. **Encapsulation** is the wrapping of data and operations/ functions (that operate on the data).
3. **Modularity** is the property of a system that has been decomposed into a set of cohesive and loosely coupled modules.
4. **Inheritance** is the capability of one class of things to inherit capabilities or properties from another class.
5. **Polymorphism** is the ability for a message or data to be processed in more than one form.

Header Files

We have used the following header files for our program:

1. iostream.h : For input and output functions like cout and cin.
2. conio.h : For clrscr(), getch(), gotoxy(), functions.
3. fstream.h : For declaring classes are used with files in this program.
4. ctype.h : For toupper() function.
5. stdlib.h : For randomize() and system() functions.
6. string.h : For strcpy() function.
7. stdio.h : For gets() and puts() function.
8. dos.h : For date structure and extracting current date from system.

Some in built functions we have used are:

1. gotoxy() : To reach the specified position on the output screen using x and y coordinates.
2. clrscr() : To clear the output screen.
3. toupper() : To convert all the characters in a char variable to uppercase.
4. randomize():To create a random number in the specified range of integers.
5. strcpy() : To copy a values of one string to another.
6. gets() : To input a string.
7. puts() : To print a string.

(Note: In C++, string means an array of char variable.)

Data Files Used

* We have created and used the following data **Files** in our program:

1. Cas.txt : A data file to store all the information about Cassettes in the Music Library, including their Code, Name, Number, and Daily Rent.
2. Bill.txt : A data file to store the Bill Number of all the bills issued till now along with the Codes of Cassettes issued.

Classes and Their Members

* We have used the following classes in our program:

1. Class **Cassettes**:

|  |  |
| --- | --- |
| DATA MEMBERS | MEMBER FUNCTIONS |
| Int CasCode | DisplayCasList ( ) |
| char CasName [30] | Show ( int ) |
| Int TotalCas | Display ( int ) |
| float DailyRent | CalTotal (int [ ] ) |
|  | IncCasNo ( int ) |
|  | GetCasCode ( ) |
|  | DecCasNo ( ) |

* + 1. DisplayCasList(): To show cassette list to the buyer.
    2. Show() : To bring the cassette details from the file to the program.
    3. Display() : To show the selected cassettes to the buyer before proceeding to checkout.
    4. CalTotal() : To calculate the total daily rent of the selected cassettes by the user.
    5. GetCasCode() : It returns the required Cassette Code.
    6. DecCasNo() : To decrease the number of Cassettes in the file at the time of issue.
    7. IncCasNo() : To increase the number of Cassettes in the file after returning.

1. Class **Rent**:

|  |  |
| --- | --- |
| DATA MEMBERS | MEMBER FUNCTONS |
| No Data Members | MakeBill ( int [ ] ) |
| PrintBill ( int [ ] , float ) |
| Renting ( ) |
| Return ( ) |

1. MakeBill() : Asks confirmation regarding the cassettes and number of days for the issue.
2. PrintBill() : Prints the final bill showing date, issued cassettes and final charges.
3. Renting() : Asking the buyer regarding the number of cassettes to be issued and which ones to be issued.
4. Return() : To return the issued Cassettes and invoke IncCasNo() function.
5. Class **Bill**:

|  |  |
| --- | --- |
| DATA MEMBERS | MEMBER FUNCTIONS |
| Int BillNo | GetBillNo ( ) |
| Int CCode [5] | Increase ( int ) |
|  | InitBill ( int , int [ ] ) |

* + 1. GetBillNo() : Returns required Bill Number from the file ‘Bill.txt’.
    2. Increase() : Reads the corresponding Cassettes’ Code to Bill Number from ‘Bill.txt’ and increases their number in ‘Cas.txt’.
    3. InitBill() : Store the value of Bill Number generated and Cassette Codes in the file ‘Bill.txt’.

1. Class **Admin**:

|  |  |
| --- | --- |
| DATA MEMBERS | MEMBER FUNCTIONS |
| No Data Members | Password () |
| AdminMenu () |
| Add () |
| Delete () |
| Modify () |

1. Password () : Public member function to check the identity of administrator through password.
2. AdminMenu () : Function to call respective functions as per the choice of administrator.
3. Add () : Function to add records to Cas.txt’ file.
4. Modify () : Function to modify records of ‘Cas.txt’ file.
5. Delete () : Function to delete record as per given Cassettes Code from ‘Cas.txt’ file.

Source File

#include<iostream.h>

#include<fstream.h>

#include<string.h>

#include<stdio.h>

#include<conio.h>

#include<ctype.h>

#include<stdlib.h>

#include<dos.h>

class Cassettes

{

public:

void GetData ();

void Modify ();

void DisplayCasList ();

void Show ( int i );

void Display ( int TCasCode );

float CalTotal ( int TCasCode [5] );

void IncCasNo ( int TCasCode );

int GetCasCode ()

{

return CasCode;

}

void DecCasNo ()

{

TotalCas -= 1;

}

Cassettes ()

{

CasCode = 0;

strcpy ( CasName, "A" );

TotalCas = 0;

DailyRent = 0.0 ;

}

private:

int CasCode;

char CasName [30];

int TotalCas;

float DailyRent;

};

class Rent

{

void MakeBill ( int TCasCode [5] );

void PrintBill ( int TCasCode [5], float n );

public:

void Renting ();

void Return ();

};

class Bill

{

int BillNo;

int CCode [5];

public:

int GetBillNo ()

{

return BillNo;

}

void Increase ( int TBillNo );

void InitBill ( int TBillNo, int TCCode [5] );

};

class Admin

{

void AdminMenu ();

void Add ();

void Modify ();

void Delete ();

public:

void Password ();

};

void Cassettes :: GetData ()

{

cout << "\n\tCas Code : ";

cin >> CasCode;

cout << "\n\tCas Name : ";

gets ( CasName );

cout << "\n\tTotal no of Cas : ";

cin >> TotalCas;

cout << "\n\tCas Daily Rent : ";

cin >> DailyRent;

}

void Cassettes :: Modify ()

{

cout << "\n\tEnter New Details : \n";

char TCasName [30];

int TTotalCas = 0;

float TDailyRent = 0.0;

cout << "\n\tCassette Name : ";

gets ( TCasName );

cout << "\tNumber of Cassettes : ";

cin >> TTotalCas;

cout << "\tDaily Rent : ";

cin >> TDailyRent;

strcpy ( CasName, TCasName );

TotalCas = TTotalCas;

DailyRent = TDailyRent;

}

void Cassettes :: DisplayCasList ( void )

{

clrscr ();

int j = 0;

Cassettes C [10];

ifstream F;

gotoxy (21,1);

cout << ">>> List of Cassettes in Our Music Library <<<\n\n";

for ( int x = 0; x < 40; x++)

cout << "=-";

gotoxy (3,5);

cout << "Cassette Code";

gotoxy (22,5);

cout << "Cassette Name";

gotoxy (40,5);

cout << "No. of Cassettes";

gotoxy (61,5);

cout << "Daily Rent (in Rs.)\n";

for ( int y = 0; y < 40; y++)

cout << "\_\_";

F.open ( "Cas.txt", ios::in );

F.seekg ( 0, ios::beg );

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

{

j = i + 7;

F.read ( ( char \* ) & C[i], sizeof ( C[i] ) );

C[i].Show ( j );

}

F.close ();

cout << endl;

for ( int z = 0; z < 40; z++)

cout << "\_\_";

}

void Cassettes :: Show ( int j )

{

gotoxy (8,j);

cout << CasCode;

gotoxy (23,j);

puts ( CasName );

gotoxy (48,j);

cout << TotalCas;

gotoxy (69,j);

cout << DailyRent;

}

void Cassettes :: Display ( int TCasCode )

{

ifstream F;

F.open ( "Cas.txt", ios::in );

F.seekg ( 0, ios::beg );

while ( !F.eof () )

{

F.read ( ( char \* ) this, sizeof ( Cassettes ) );

if ( TCasCode == CasCode )

{

cout << "\n\tName : ";

puts ( CasName );

cout << "\tDaily Rent : " << DailyRent;

break;

}

else if ( TCasCode == 0 )

continue;

}

F.close ();

}

float Cassettes :: CalTotal ( int TCasCode [5] )

{

long Pos;

float total = 0;

fstream F;

F.open ( "Cas.txt", ios::in| ios::out| ios::app );

F.seekg ( 0, ios::beg );

for ( int i = 0; i < 5; i++)

{

while ( !F.eof () )

{

Pos = F.tellg ();

F.read ( ( char \* ) this, sizeof ( Cassettes ) );

if ( TCasCode [i] == GetCasCode () )

{

total += DailyRent;

DecCasNo ();

F.seekg ( Pos );

F.write ( ( char \* ) this, sizeof ( Cassettes ) );

}

else if ( TCasCode [i] == 0)

continue;

}

}

F.close ();

return total;

}

void Cassettes :: IncCasNo ( int TCasCode )

{

long Pos;

fstream F;

F.open ( "Cas.txt", ios::in| ios::out| ios::app );

F.seekg ( 0 );

while ( !F.eof () )

{

Pos = F.tellg ();

F.read ( ( char \* ) this, sizeof ( Cassettes ) );

if ( TCasCode == GetCasCode () )

{

TotalCas+=1;

F.seekg ( Pos );

F.write ( (char \* ) this, sizeof ( Cassettes ) );

break;

}

}

F.close();

}

void Rent :: Renting ()

{

clrscr ();

int N;

char ch;

int TCasCode [5];

Cassettes C;

C.DisplayCasList ();

cout << "\n\tEnter the number of Cassettes which you want to Rent ( Maximum 5 ) : ";

cin >> N;

cout << "\n\tEnter the Cassette Codes which you want to Rent :\n ";

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

{

again1:

cout << "\t";

cin >> TCasCode [i];

if ( TCasCode [i] >= 0 && TCasCode [i] <= 10 )

continue;

else

{

cout << "\t\tPlease enter a VALID character.\n";

goto again1;

}

}

for ( int j = N; j <= 5; j++ )

TCasCode [j] = 0;

again2:

cout << "\tConfirm ? (Y or N) ... ";

cin >> ch;

ch = toupper( ch );

if ( ch == 'Y' )

MakeBill ( TCasCode );

else if ( ch == 'N' )

cout << " \t\t\t\Thanks For Visiting.\n";

else

{

cout << "\t\tPlease enter a VALID character.\n";

goto again2;

}

}

void Rent :: MakeBill ( int TCasCode[5] )

{

clrscr ();

Cassettes C;

char ch;

char c;

float n;

gotoxy (4,4);

cout << "You have selected the following Cassettes to Issue : \n\n";

for ( int i = 0; i < 5; i++)

C.Display ( TCasCode [i] );

again1:

cout << "\n\n\tProceed To Checkout ? (Y/N) ... ";

cin >> ch;

ch = toupper( ch );

if ( ch == 'Y' )

{

again2:

cout << "\n\tEnter the no. of Days you want to take the cassettes : ";

cin >> n;

if ( n > 0 && n <= 31 )

PrintBill ( TCasCode, n );

else

{

cout << "\tPlease enter a VALID number.\n";

goto again2;

}

}

else if ( ch == 'N' )

{

cout << "\n\tPlease try again.\n";

Renting ();

}

else

{

cout << "\tEnter a VALID character.";

goto again1;

}

}

void Rent :: PrintBill ( int TCasCode [5], float n )

{

clrscr ();

int d1 = 0, m1 = 0, y1 = 0;

struct date x; // For extracting system date

getdate ( &x );

d1 = x.da\_day;

m1 = x.da\_mon;

y1 = x.da\_year;

int d = d1, m = m1, y = y1;

for ( float i = 1; i <= n; i++ )

{

d += 1;

if ( m == 12 )

if ( d > 31 )

{

y += 1;

m = 1;

d = 1;

}

else if ( m == 1|| m == 3|| m == 5|| m == 7|| m == 8|| m == 10 )

if ( d > 31 )

{

m += 1;

d = 1;

}

else if ( m == 4|| m == 6|| m == 9|| m == 11 )

if ( d > 30 )

{

m += 1;

d = 1;

}

else if ( m == 2 )

{

if ( y % 100 != 0 && y % 4 == 0 && y % 100 == 0 && y % 400 == 0 )

if ( d > 29 )

{

m += 1;

d = 1;

}

else

if ( d > 28 )

{

m += 1;

d = 1;

}

}

}

randomize ();

int TBill = 0;

TBill = random ( 900 ) + 100;

Bill B;

ofstream F;

F.open ( "Bill.txt", ios::out| ios::app );

B.InitBill ( TBill, TCasCode );

F.write ( ( char \* ) & B, sizeof ( B ) );

F.close ();

float total = 0;

float sum = 0;

Cassettes C;

gotoxy (35,2);

cout << "BASS BOOSTED";

gotoxy (60,4);

cout << "Date : " << d1 << "-" << m1 << "-" << y1;

gotoxy (5,4);

cout << "Bill No. : " << TBill;

gotoxy (5,5);

cout << "Return Date : " << d << "- " << m << "- " << y;

gotoxy (5,7);

cout << "Cassettes Codes : ";

for ( int j = 0; j < 5; j++ )

{

if ( TCasCode [j] != 0 )

cout << TCasCode [j] << ", ";

else

break;

}

total = C.CalTotal ( TCasCode );

gotoxy (5,9);

cout << "Total Daily Rent : Rs. " << total;

gotoxy (5,10);

cout << "Number of Days : " << n;

gotoxy (5,11);

cout << "Safety Charge : Rs. 200.0";

gotoxy (40,11);

cout << "(At the time of return show the";

gotoxy (40,12);

cout << "bill and get safety charge back.)";

gotoxy (5,14);

cout << "Total Charges : Rs.";

sum = ( n \* total ) + 200;

cout << sum;

gotoxy (8,16);

cout << "Please Bring The Hard Copy of This Bill on Return of The Cassette.";

gotoxy (32,19);

cout << "Thank You For Visiting.";

}

void Rent :: Return ()

{

clrscr();

Bill B;

int TBill, Temp = 0;

again:

gotoxy (55,3);

cout << "0 - Exit.";

gotoxy (5,3);

cout << "Enter Your Bill No. : ";

cin >> TBill;

ifstream F;

F.open ( "Bill.txt", ios::in );

F.seekg ( 0, ios::beg );

while ( !F.eof () )

{

F.read ( ( char \* ) this, sizeof ( Bill ) );

Temp = B.GetBillNo ();

if ( Temp == TBill )

{

gotoxy (5,5);

cout << "Please Collect Your Safety Charge. ";

B.Increase ( TBill );

gotoxy (34,7);

cout << "Pleasure Doing Business With You.";

gotoxy (45,9);

cout << "Visit Again.";

}

else if ( TBill == 0 )

break;

else

{

gotoxy (5,5);

cout<<"Sorry, Your Bill No. Does Not Exist!!! Try Again.";

gotoxy (5,7);

system ( "pause" );

clrscr ();

goto again;

}

}

F.close ();

}

void Bill :: Increase ( int TBillNo )

{

long Pos;

Cassettes C [5];

ifstream F;

fstream File;

F.open ( "Bill.txt", ios::in );

File.open ( "Cas.txt", ios::in| ios::out| ios::app );

for ( int i = 0; i < 5; i++)

{

F.seekg ( 0, ios::beg );

while ( !F.eof () )

{

F.read ( ( char \* ) this, sizeof ( Bill ) );

if ( TBillNo == GetBillNo () )

{

C[i].IncCasNo ( CCode [i] );

while ( !File.eof () )

{

Pos = File.tellg ();

File.read ( ( char \* ) & C [i], sizeof ( C [i] ) );

if ( C [i].GetCasCode () == CCode [i] )

{

File.seekg ( Pos );

File.write ( ( char \* ) & C [i], sizeof ( C [i] ) );

break;

}

}

break;

}

else

continue;

}

}

F.close ();

File.close ();

}

void Bill :: InitBill ( int TBillNo, int TCCode [5] )

{

BillNo = TBillNo;

CCode [0] = TCCode [0];

CCode [1] = TCCode [1];

CCode [2] = TCCode [2];

CCode [3] = TCCode [3];

CCode [4] = TCCode [4];

}

void Admin :: Password ()

{

clrscr ();

Cassettes C;

char P [6];

gotoxy (35,5);

cout << "BASS BOOSTED";

gotoxy (32,7);

cout << "Administrator Menu";

again:

gotoxy (32,10);

cout << "Password : ";

for ( int i = 0; i < 5; i++ )

{

P [i] = getch ();

printf ( "\*" );

}

gotoxy (34,13);

if ( P [0] == 'm' && P [1] == 'u' && P [2] == 's' && P [3] == 'i' && P [4] == 'c' )

cout << "Access Granted.";

else

{

cout << "Access Aborted.";

goto again;

}

gotoxy (29,20);

system ( "pause" );

AdminMenu ();

}

void Admin :: AdminMenu ()

{

clrscr ();

int ch;

cout << "\n\n\n\tAdministrator Menu : \n";

cout << "\n\t1 - Add Cassettes.\n";

cout << "\n\t2 - Modify Cassettes.\n";

cout << "\n\t3 - Delete Cassettes.\n";

gotoxy (55,5);

cout << "0 - Exit.";

again:

gotoxy (20,10);

cout << "\n\n\tYour Choice ... ";

cin >> ch;

gotoxy (29,20);

if ( ch == 1 )

Add ();

else if ( ch == 2)

Modify ();

else if ( ch == 3 )

Delete ();

else if ( ch == 0 )

system ( "pause" );

else

{

cout << "\t Enter a VALID character.";

goto again;

}

}

void Admin :: Add ()

{

clrscr ();

Cassettes C;

ofstream F;

char ch = 'Y';

F.open ( "Cas.txt", ios::out| ios::app );

while ( ch != 'N' )

{

C.GetData ();

F.write ( ( char \* ) & C, sizeof ( C ) );

cout << "\n\tContinue (Y/N) ... ";

cin >> ch;

ch = toupper ( ch );

}

F.close ();

AdminMenu ();

}

void Admin :: Modify ()

{

clrscr ();

char ch;

Cassettes C;

fstream F;

F.open ( "Cas.txt", ios::in| ios::out );

int TCasCode;

long Pos;

char found = 'F';

again:

cout << "\n\tEnter Cassettes Code whose record is to be Modified : ";

cin >> TCasCode;

while ( !F.eof () )

{

Pos = F.tellg ();

F.read ( ( char \* ) & C, sizeof ( C ) );

if ( C.GetCasCode () == TCasCode )

{

C.Modify ();

F.seekg ( Pos );

F.write ( ( char \* ) & C, sizeof ( C ) );

found = 't';

break;

}

}

if ( found == 'f' )

cout << "\n\tCassettes not found. Try Again.";

cout << "\n\tContinue (Y/N) ... ";

cin >> ch;

ch = toupper ( ch );

if ( ch == 'Y')

goto again;

F.close ();

AdminMenu ();

}

void Admin :: Delete ()

{

clrscr ();

Cassettes C;

char ch;

char found = 'f';

ifstream F;

ofstream File;

F.open ( "Cas.txt", ios::in );

File.open ( "Temp.txt", ios::out );

int TCasCode;

cout << "\n\tEnter Cassette Code whose record is to be deleted : ";

cin >> TCasCode;

while ( !F.eof () )

{

F.read ( ( char \* ) & C, sizeof ( C ) );

if ( C.GetCasCode () == TCasCode )

{

found = 't';

C.Display ( TCasCode );

cout << "\n\tConfirm (Y/N) ... ";

cin >> ch;

ch = toupper ( ch );

if ( ch == 'N' )

{

File.write ( ( char \* ) & C, sizeof ( C ) );

break;

}

}

else

File.write ( ( char \* ) & C, sizeof ( C ) );

}

if ( found == 'f' )

cout << "\n\tRecord not found.";

F.close ();

File.close ();

remove ( "Cas.txt" );

rename ( "Temp.txt", "Cas.txt" );

AdminMenu ();

}

void main ()

{

clrscr ();

int ch;

Rent R;

Admin A;

cout << "\n\n\t\t\t Welcome to Our Music Library,\n\n";

cout << "\n\n\t\t\t Bass Boosted";

gotoxy (29,20);

system ( "pause" );

clrscr ();

cout << "\n\n\n\t\tMenu :\n";

cout << "\n\t1 - To Take a Cassette on Rent.\n";

cout << "\n\t2 - To Return a Cassette.\n";

cout << "\n\t3 - Open Administrator Menu.\n";

cout << "\n\t0 - To Exit.\n";

again5:

cout << "\n\tYour Choice ... ";

cin >> ch;

if ( ch == 1 )

R.Renting ();

else if ( ch == 2 )

R.Return ();

else if ( ch == 3 )

A.Password ();

else if ( ch == 0 )

{

gotoxy (36,15);

cout << "Thank You.";

gotoxy (33,17);

cout << "Please Visit Again.";

}

else

{

cout << "\tEnter a VALID character.";

goto again5;

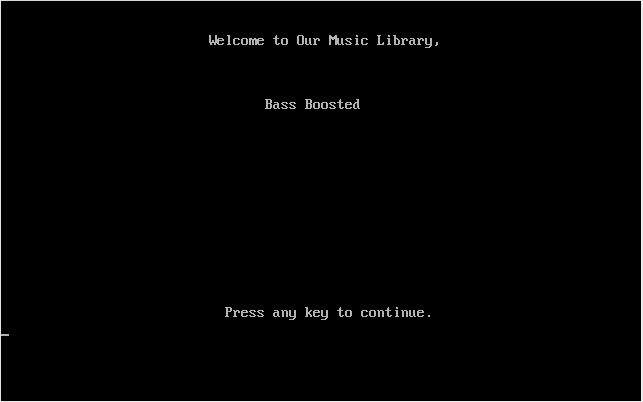
}

getch ();

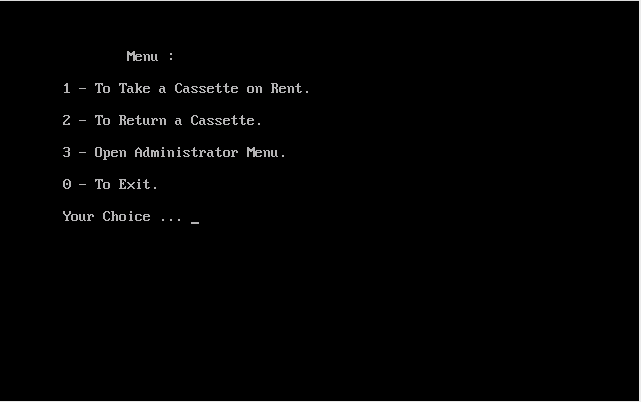
}

Glimpses of Our Programme

* Welcome Page



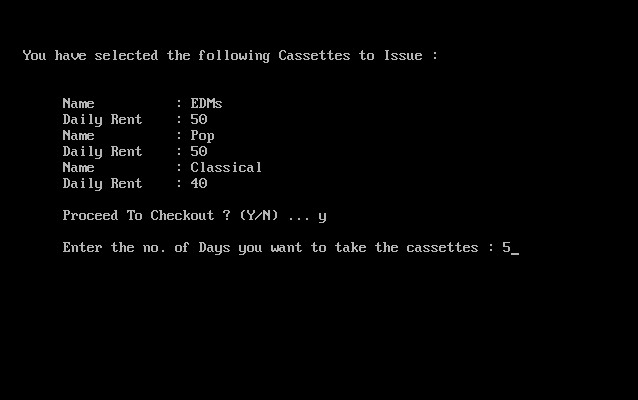
* Menu List



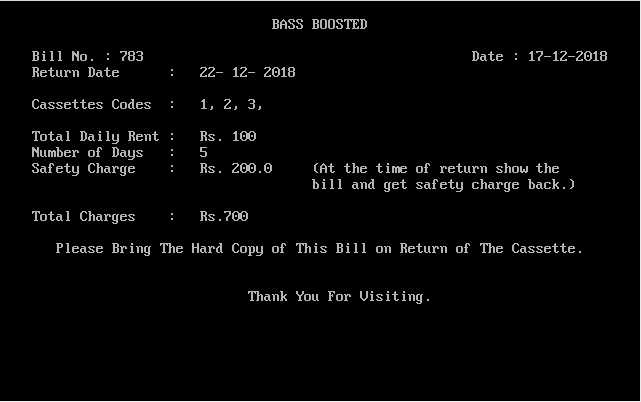
* Cassettes List



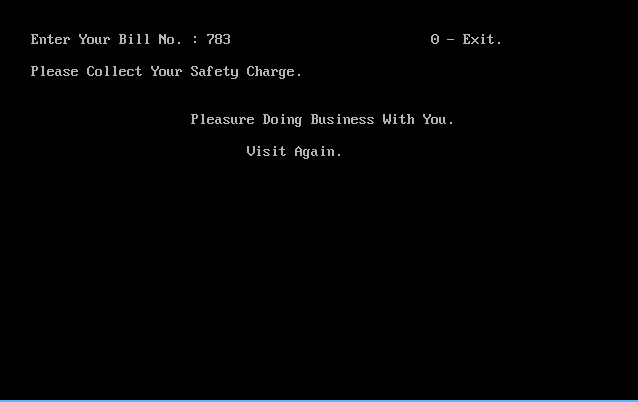
* Confirmation for Rent



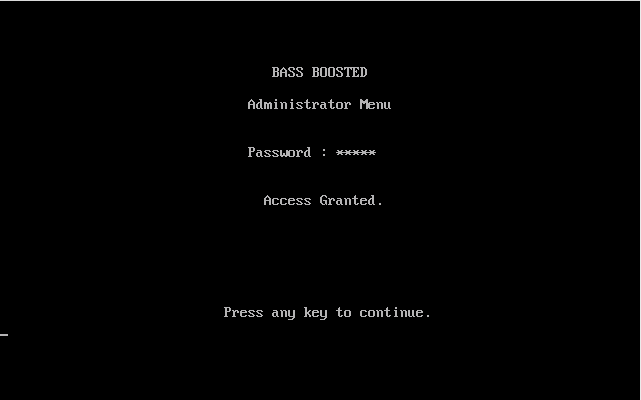
* Final Bill



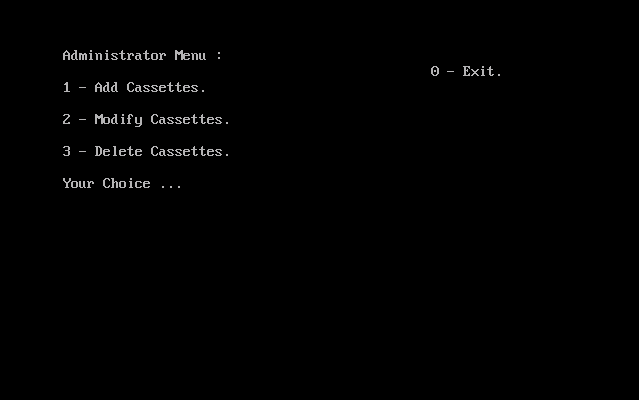
* Returning of Cassettes



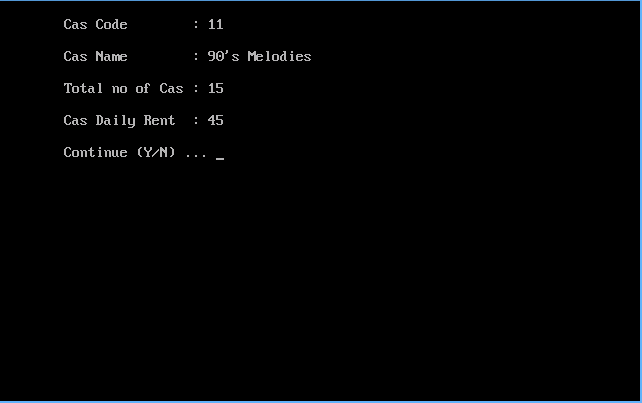
* Administrator Login Page



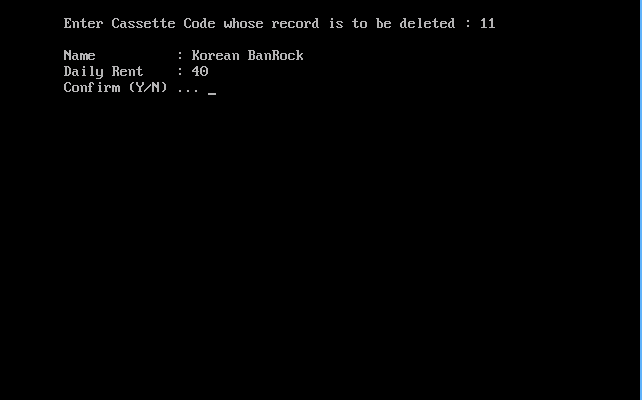
* Administrator Menu



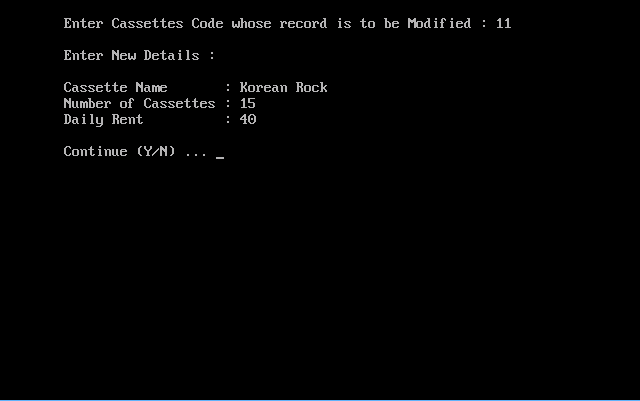
* Add Function



* Delete Function



* Modify Function



Bibliography

* For Background image in Introduction :

<http://mercercognitivepsychology.pbworks.com/w/page/70793859/Music%20and%20Memory>

* For References to Codes:

1. Computer Science with C++,

Class XII, Volume I by Sumita Arora

1. Computer Science with C++,

Class XI by Sumita Arora