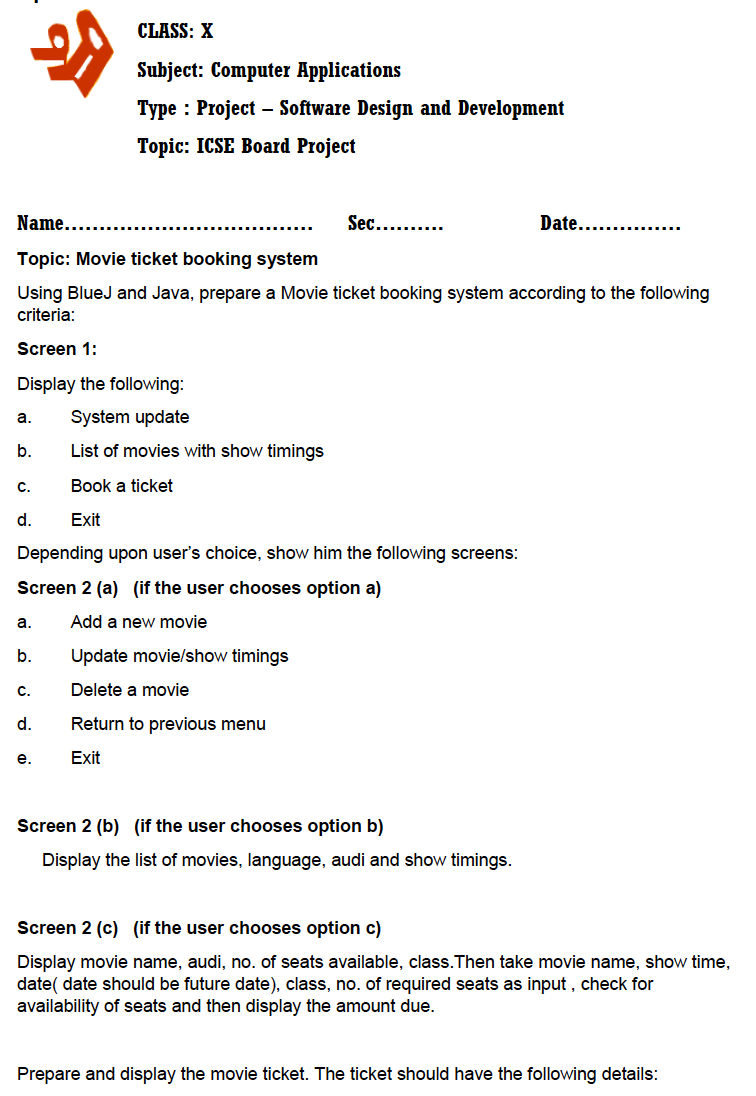
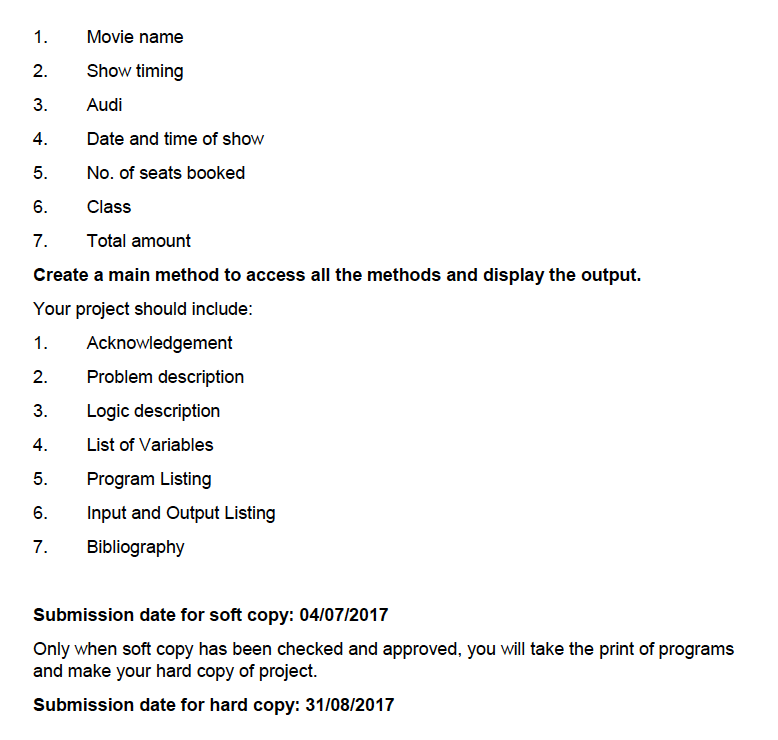
Acknowledgements

I

Project Description

LOGIC

DESCRIPTION

**Store**

The Store class is the core of this program. This class initialises all the important variables of the program. It uses java.io.Serializable to write all the variables to a file named “Data.txt”. In this class all the major variables are declared and initialised. This class also calls three different methods from two other classes. These methods act as the preset values. This class has only one output and that is “System ready to use”, if all the variables have been declared and initialised successfully or “Error”, if anything in the middle goes wrong.

This class ensures that all the changes made by the user ( Usually in System Update ) remain unchanged even if the program is closed and opened again.

**Preset**

This class has the method preset(), which is called from the store class.

This method adds default values to the various variables. This method enters ten movies, their language, duration and about 50 shows as a default value when starting the program.

**AudiFormat**

This class has the following two methods, Audi\_G () and Audi\_3 ().

Both of these methods are called from the Store class. These Methods gives default values to the variables formatAudi1 and formatAudi3 respectively.

These default values gives the structure to the Audis and helps in displaying the seats ( In booking during choose your seat option ).

NOTE : All of the above three classes work in the background and have negligible output. All of the three classes given above at as the constructor of this program as they declare, initialise and give default values to the variables just like a constructor.

**Main**

The software starts working when the main class is initiated. This class joins the whole software together. This class acts as a link between the Login, SystemUpdate, Show, Book and Screen classes. Objects for the above classes are created in the main method of the main class. Specific methods from the above given classes are called from inside a while loop. This while loop ensures the continuity of the program.

The main class calls all the methods from different classes. All these methods are designed to return a value to the main method of the main class so as to maintain the continuity of the program.

Entering “Exit”, “E”, or “D” in the main method causes the program to terminate.

If the user chooses the option to update the system, then the main method first initiates the login class. If the Login is successful, then the login class sends back a true boolean to the main class, otherwise it sends a false boolean to the main class. The main class receives the boolean value and lets the user access the System Update only if the returned boolean is true.

Therefore, this main method also protects the software from unauthorised persons.

**Login**

This class contains the method login (). This method verifies if the user is an authorised person or not by asking the user for the pre-shared user ID and password and hence maintains the security of the program.

This method asks for the user ID first which is not case sensitive. Over here it uses the .eualsIgnoreCae() method inside an if-else ladder. On entering wrong, it prompts the user to enter the ID again. Three successive false IDs freezes the program for ten seconds hence increasing the security level of the program.

This method asks for the password after the user could enter the correct user ID. The password is case sensitive, which increases the security even more. Over here it uses the .euals() method inside an if-else ladder. On entering wrong, it prompts the user to enter the password again. Three successive false passwords freezes the program for ten seconds hence increasing the security level of the program.

This method is called by the main method in the main class. This method is made to return a boolean value. If the login is successful, this method returns a true value. On the other hand, if the login is unsuccessful due to some reason, this method returns a false value.

**Screen**

This class contains the output screen for the home screen and the System Update screen. This separation makes this program more modular and efficient.

**SystemUpdate**

This class joins the System Update section of the program. It calls AddMovie, UpdateMovie, DeleteMovie and Screen according to the users choice. It also returns the program to the main class once the user has completed the desired task and wants to exit. This maintains the continuity of the program.

**AddMovie**

This class has only one method namely the add() method. In this method, the user is asked for the name of the movie that is to be added.

Then this method searches the movieList variable to find if the entered movie is already playing. This is a linear search using the .equals() method. If the movie is already playing ( i.e. the search is successful ), then it gives the user the option to overwrite the existing movie ( to change language or duration ), to add it anyway ( two copies of the same movie will be playing ), or to exit this section. Then the program asks for the language and duration of the movie. If the result of the search is not successful the program moves forward and asks for the language and duration of the movie.

After all the input has been taken ( In String value ), the program assigns the given movie a code and afterwards it increments the value of this code for future use. Now the program stores all these details in the movieList variable.

**UpdateMovie**

In this class the program first displays all the movies playing by printing the movieList array and then asks the user which movie does it want to update.The program here extracts the code of that particular movie. Now the program asks the user what does it want to update in that movie. According to the users choice, the program calls related methods from the class Update to handle the function and passes the code ( extracted earlier ) to these methods.

**Update**

This class contains four methods. All of the four methods are designed to handle and perform specific and distinct functions. All of these four methods are discussed below:

**1) Edit (int code):** This method accepts one integer parameter as the code of the movie that needs to be updated. Here, in a while loop the program asks the user if ti want stop update the name, language or the duration of the movie. The program takes an input and then according to the choice of the user it overwrites that particular cell in the movieList array.

**2) AddShow (int code):** This method accepts one integer parameter as the code of the movie that needs to be updated. Now the program displays all the days of the week and asks the user its choice. Then the program asks the user for the Audi and show time that needs to be added. According to day entered by the user, the program searches either of the seven storing arrays ( D1, D2, D3, D4, D5, D6 and D7 ). If the show time entered by that user is a null, the program writes the movie’s code followed by a dot and a capital S in that cell, to symbolise that that slot is full.Next, by using the code of the movie, the program extracts the duration of the movie. Now the program runs a loop of length equal to the movies duration and fills the cells adjacent to the newly added cells with the movie code. The last cell is written with the movie code followed by a dot and a capital E to symbolise the end of that show. Here the procedure of adding a show ends.

**3) DeleteShow (int code):** This method accepts one integer parameter as the code of the movie that needs to be updated. The program now asks the user the day on which the show needs to be deleted. According to the day entered by the user, the program runs a linear search in the appropriate array ( D1, D2, D3, D4, D5, D6 and D7 ) and prints all the row numbers as time and column numbers as the Audi number which contain the entered code followed by a dot and followed by a capital S. The user is now given the choice of which show to delete. The user enters the timing of the that needs to be deleted.Now the program extracts the duration of the movie using the code of the movie. Now the program overwrites the respective cells and makes all those cells null by running a loop.

**4) ShowTime (int code):** This method accepts one integer parameter as the code of the movie that needs to be updated. The program now asks the user the day on which it needs to see the show timings for. According to the day entered by the user, the program runs a linear search in the appropriate array ( D1, D2, D3, D4, D5, D6 and D7 ) and prints all the row numbers as time and column numbers as the Audi number which contain the entered code followed by a dot and followed by a capital S.

**Show**

This class is called from the main method of the main class. It works on the same principle as Showtime method in the Update class only with an improved user interface. This class helps in showing non authorised persons the list of show timings that he/she can choose from.

**Book**

This class acts as the backbone of the book-a-ticket option of the software. Here it calls all the different classes to perform their specific functions.

**Movie**

This class contains of only one method which is used to print the list of movies, their languages and durations in a systematic form. It accepts the movie for which the user wants to book the tickets for. It extracts the code of the entered movie and sends it back to the class that called it, that is the Book class.

**Day**

This class displays the days of the week. Current day is followed by current date and the rest are followed by their upcoming dates. For this the program uses simpleDateFormat and calendar methods. This is done so that the user has no other option but to choose a future date to book its tickets.

This class returns the choice of the day back to the Book class.

**Show\_Book**

This class also has only one method which accepts the code of the movie and the day as parameters and just like the method showTime in Update class, it performs a liner search for the timings of that particular movie on that particular day and prints it. It also asks the user for its choice and returns the value to the Book class.

**Print**

Depending upon the Audi selected by the user, this class displays that particular Audi along with the booked and available seats of that Audi for the user to choose their preferred seats.

**DeCode**

This class has only one method which accepts the Audi number as a parameter and the seat choosen as another parameter. This method contains the price values of all the seats in both the Audi types. This method therefore increases the flexibility of the program as the user may choose seats of any price value. This method returns the price to the Price class for computation.

**Price**

This class contains only one method which accepts the Audi number, number of seats booked by the user and an array of all the seats booked by the user. This class in collaboration with the DeCode class, compute the total cost of the seats booked by the user.

**Seat**

This class acts as the second backbone of the book-a-ticket option of this software. It takes in the code of the movie, time of the show, day of the show and Audi in which the show is showing as the input parameters. It combines and handles the process of selecting seats, ordering food, printing the tickets and works internally to compute the grand total of the user.

**Food**

The class food is an added privilege in this software. It enables the user to order food online and beforehand to avoid standing in quest and missing parts of the movie during interval. It consists of four methods. One method is responsible for printing the menu on the screen. The other method is responsible for storing the food items name entered by the user to print it in the bill. Another method is responsible to keep in account of the cost of the food items ordered by the user. The last method is responsible for taking in input of the food item wanted by the user, work systematically and in an organised way with all the other three methods and return values appropriately.

**Bill**

This class has three methods. The first method displays all the important data along with the price and asks the user for confirmation.

The second method deals with the payment process and takes the necessary details like the account number, date of expiry and the cvv number of the card from which the payment has to made. The last method deals with the printing of the tickets.

Code

**Class: Store**

import java.io.\*;

public class Store implements java.io.Serializable

{

public String formatAudi1 [][];

public String formatAudi3 [][];

public String D1 [][];

public String D2 [][];

public String D3 [][];

public String D4 [][];

public String D5 [][];

public String D6 [][];

public String D7 [][];

public String movieList [][];

public int code;

public static void main(String [] args)

{

Store s = new Store();

s.code = 1;

s.formatAudi1 = new String [17][14];

s.formatAudi3 = new String [12][12];

s.D1 = new String [2359][4];

s.D2 = new String [2359][4];

s.D3 = new String [2359][4];

s.D4 = new String [2359][4];

s.D5 = new String [2359][4];

s.D6 = new String [2359][4];

s.D7 = new String [2359][4];

s.movieList = new String [50][4];

for ( int i = 0,j=0,c=0;c<9436;++c)

{ s.D1 [i][j] = "";s.D7 [i][j] = "";s.D6 [i][j] = "";

s.D2 [i][j] = "";s.D5 [i][j] = "";

s.D3 [i][j] = "";s.D4 [i][j] = "";

if(j==3){++i;j=0;}else{++j;}}

for(int i = 0, j = 0 , c = 0; c < 200 ; ++c )

{ s.movieList [i][j] = "";

if(i==49)

{i=0;++j;}else{ ++i;}}

s.movieList [0][0] = "Movie Name ";

s.movieList [0][1] = "Lang";

s.movieList [0][2] = "Mins";

try {

FileOutputStream fileOut = new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s); out.close();fileOut.close(); System.out.print("\u000C");

}catch(IOException i) { i.printStackTrace();System.out.printf("ERROR 000”);}

Preset P = new Preset (); AudiFormat Af= new AudiFormat();

int ran = Af.Audi\_G(); ran= Af.Audi\_3(); ran = P.preset();

System.out.println("System ready to use");}

}

**Class: Preset**

import java.io.\*;

public class Preset

{

public int preset ()

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

s.movieList [1][0] = "Avenger:Infinity War";s.movieList [6][0] = "The Golden Circle";

s.movieList [1][1] = "Eng 3D";s.movieList [6][1] = "Eng";

s.movieList [1][2] = "107";s.movieList [6][2] = "115";

s.movieList [1][3] = "1";s.movieList [6][3] = "8";

s.movieList [2][0] = "Jab Harry Met Sejal";s.movieList [7][0] = "Secret Superstar";

s.movieList [2][1] = "Hindi";s.movieList [7][1] = "Hindi";

s.movieList [2][2] = "121";s.movieList [7][2] = "105";

s.movieList [2][3] = "2";s.movieList [7][3] = "4";

s.movieList [3][0] = "Spiderman Homecoming";s.movieList [8][0] = "Lego:Ninjago Movie";

s.movieList [3][1] = "Eng 3D";s.movieList [8][1] = "Eng 3D";

s.movieList [3][2] = "120";s.movieList [8][2] = "125";

s.movieList [3][3] = "5";s.movieList [8][3] = "3";

s.movieList [4][0] = "Toilet Ek Prem Katha";s.movieList [9][0] = "The Emoji Movie";

s.movieList [4][1] = "Hindi";s.movieList [9][1] = "Eng 3D";

s.movieList [4][2] = "133";s.movieList [9][2] = "95";

s.movieList [4][3] = "7";s.movieList [9][3] = "9";

s.movieList [5][0] = "The Death Cure";s.movieList [10][0] = "Annabelle Creation";

s.movieList [5][1] = "Eng";s.movieList [10][1] = "Eng 3D";

s.movieList [5][2] = "95";s.movieList [10][2] = "97";

s.movieList [5][3] = "6";s.movieList [10][3] = "10";

s.D1 [830][1] = "1.S.";

s.D1 [1017][1] = "1.E";

s.D1 [1215][1] = "7.S.";

s.D1 [1428][1] = "7.E";

s.D1 [1500][1] = "5.S.";

s.D1 [1700][1] = "5.E";

s.D1 [1820][1] = "2.S.";

s.D1 [2021][1] = "2.E";

s.D1 [2140][1] = "10.S.";

s.D1 [2317][1] = "10.E";

s.D2 [830][1] = "8.S.";

s.D2 [1025][1] = "8.E";

s.D2 [1500][1] = "5.S.";

s.D2 [1800][1] = "5.E";

s.D2 [1820][1] = "2.S.";

s.D2 [2021][1] = "2.E";

s.D2 [2140][1] = "5.S.";

s.D2 [2340][1] = "5.E";

s.D3 [830][1] = "1.S.";

s.D3 [1017][1] = "1.E";

s.D3 [1215][1] = "7.S.";

s.D3 [1428][1] = "7.E";

s.D3 [1820][1] = "9.S.";

s.D3 [1955][1] = "9.E";

s.D4 [830][1] = "1.S.";

s.D4 [1017][1] = "1.E";

s.D4 [1215][1] = "7.S.";

s.D4 [1448][1] = "7.E";

s.D4 [1500][1] = "4.S.";

s.D4 [1645][1] = "4.E";

s.D4 [2140][1] = "10.S.";

s.D4 [2317][1] = "10.E";

s.D5 [830][1] = "8.S.";

s.D5 [1025][1] = "8.E";

s.D5 [1820][1] = "4.S.";

s.D5 [2005][1] = "4.E";

s.D5 [2140][1] = "10.S.";

s.D5 [2317][1] = "10.E";

s.D6 [830][1] = "6.S.";

s.D6 [1005][1] = "6.E";

s.D6 [1215][1] = "3.S.";

s.D6 [1420][1] = "3.E";

s.D6 [1500][1] = "2.S.";

s.D6 [1701][1] = "2.E";

s.D6 [1820][1] = "4.S.";

s.D6 [2005][1] = "4.E";

s.D7 [830][1] = "6.S.";

s.D7 [1005][1] = "6.E";

s.D7 [1820][1] = "4.S.";

s.D7 [2005][1] = "4.E";

s.D7 [2140][1] = "5.S.";

s.D7 [2340][1] = "5.E";

s.D1 [1000][2] = "3.S.";

s.D1 [1205][2] = "3.E";

s.D1 [1330][2] = "8.S.";

s.D1 [1525][2] = "8.E";

s.D1 [1615][2] = "4.S.";

s.D1 [1700][2] = "4.E";

s.D1 [1850][2] = "6.S.";

s.D1 [2025][2] = "6.E";

s.D1 [2130][2] = "5.S.";

s.D1 [2330][2] = "5.E";

s.D2 [1000][2] = "1.S.";

s.D2 [1147][2] = "1.E";

s.D2 [1330][2] = "9.S.";

s.D2 [1505][2] = "9.E";

s.D2 [1615][2] = "6.S.";

s.D2 [1750][2] = "6.E";

s.D2 [1850][2] = "4.S.";

s.D2 [2005][2] = "4.E";

s.D2 [2130][2] = "10.S.";

s.D2 [2307][2] = "10.E";

s.D3 [1000][2] = "3.S.";

s.D3 [1205][2] = "3.E";

s.D3 [1330][2] = "8.S.";

s.D3 [1525][2] = "8.E";

s.D3 [1615][2] = "5.S.";

s.D3 [1815][2] = "5.E";

s.D3 [1850][2] = "2.S.";

s.D3 [2051][2] = "2.E";

s.D3 [2130][2] = "10.S.";

s.D3 [2307][2] = "10.E";

s.D4 [1000][2] = "3.S.";

s.D4 [1205][2] = "3.E";

s.D4 [1330][2] = "8.S.";

s.D4 [1525][2] = "8.E";

s.D4 [1615][2] = "5.S.";

s.D4 [1815][2] = "5.E";

s.D4 [1850][2] = "9.S.";

s.D4 [2025][2] = "9.E";

s.D4 [2130][2] = "6.S.";

s.D4 [2305][2] = "6.E";

s.D5 [1000][2] = "1.S.";

s.D5 [1147][2] = "1.E";

s.D5 [1330][2] = "9.S.";

s.D5 [1505][2] = "9.E";

s.D5 [1615][2] = "7.S.";

s.D5 [1828][2] = "7.E";

s.D5 [1850][2] = "2.S.";

s.D5 [2051][2] = "2.E";

s.D5 [2130][2] = "6.S.";

s.D5 [2305][2] = "6.E";

s.D6 [1330][2] = "9.S.";

s.D6 [1505][2] = "9.E";

s.D6 [1615][2] = "7.S.";

s.D6 [1828][2] = "7.E";

s.D6 [1850][2] = "5.S.";

s.D6 [2050][2] = "5.E";

s.D6 [2130][2] = "10.S.";

s.D6 [2307][2] = "10.E";

s.D7 [1330][2] = "9.S.";

s.D7 [1505][2] = "9.E";

s.D7 [1615][2] = "2.S.";

s.D7 [1816][2] = "2.E";

s.D7 [2130][2] = "10.S.";

s.D7 [2307][2] = "10.E";

s.D1 [900][3] = "5.S.";

s.D1 [1100][3] = "5.E";

s.D1 [1230][3] = "5.S.";

s.D1 [1430][3] = "5.E";

s.D1 [1600][3] = "4.S.";

s.D1 [1745][3] = "4.E";

s.D1 [2100][3] = "10.S.";

s.D1 [2237][3] = "10.E";

s.D2 [900][3] = "10.S.";

s.D2 [1037][3] = "10.E";

s.D2 [1230][3] = "4.S.";

s.D2 [1415][3] = "4.E";

s.D2 [1600][3] = "6.S.";

s.D2 [1735][3] = "6.E";

s.D2 [2100][3] = "2.S.";

s.D2 [2301][3] = "2.E";

s.D3 [900][3] = "2.S.";

s.D3 [1101][3] = "2.E";

s.D3 [1230][3] = "4.S.";

s.D3 [1415][3] = "4.E";

s.D3 [1600][3] = "5.S.";

s.D3 [1800][3] = "5.E";

s.D3 [2100][3] = "9.S.";

s.D3 [2235][3] = "9.E";

s.D4 [900][3] = "10.S.";

s.D4 [1037][3] = "10.E";

s.D4 [1230][3] = "4.S.";

s.D4 [1415][3] = "4.E";

s.D4 [1600][3] = "5.S.";

s.D4 [1800][3] = "5.E";

s.D4 [2100][3] = "9.S.";

s.D4 [2235][3] = "9.E";

s.D5 [900][3] = "9.S.";

s.D5 [1035][3] = "9.E";

s.D5 [1230][3] = "6.S.";

s.D5 [1405][3] = "6.E";

s.D5 [1600][3] = "2.S.";

s.D5 [1801][3] = "2.E";

s.D5 [2100][3] = "5.S.";

s.D5 [2300][3] = "5.E";

s.D6 [900][3] = "7.S.";

s.D6 [1113][3] = "7.E";

s.D6 [1230][3] = "2.S.";

s.D6 [1431][3] = "2.E";

s.D6 [1600][3] = "6.S.";

s.D6 [1735][3] = "6.E";

s.D6 [2100][3] = "10.S.";

s.D6 [2237][3] = "10.E";

s.D7 [900][3] = "9.S.";

s.D7 [1035][3] = "9.E";

s.D7 [1230][3] = "6.S.";

s.D7 [1405][3] = "6.E";

s.D7 [1600][3] = "7.S.";

s.D7 [1813][3] = "7.E";

s.D7 [2100][3] = "5.S.";

s.D7 [2300][3] = "5.E";

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

return 1;

}

}

**Class: AudiFormat**

import java.io.\*;

**public class AudiFormat**

**{**

**public int Audi\_G ()**

**{**

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

for(int i = 0, j = 0 , c = 0;c<238;++c)

{

s.formatAudi1 [i][j] = "[]";

if(i==16)

{

i=0;

++j;

}else{

++i;}

}

s.formatAudi1 [0][0] = " "; s.formatAudi1 [0][1] = " "; s.formatAudi1 [0][2] = "A ";

s.formatAudi1 [0][3] = "B "; s.formatAudi1 [0][4] = "C "; s.formatAudi1 [0][5] = " ";

s.formatAudi1 [0][6] = "D "; s.formatAudi1 [0][7] = "E "; s.formatAudi1 [0][8] = "F ";

s.formatAudi1 [0][9] = "G "; s.formatAudi1 [0][10] = "H ";s.formatAudi1 [0][11] = " ";

s.formatAudi1 [0][12] = "\*\*";s.formatAudi1 [0][13] = "\*\*";

s.formatAudi1 [2][2] = "# ";s.formatAudi1 [3][2] = "# ";s.formatAudi1 [4][2] = "# ";

s.formatAudi1 [5][2] = "# ";

s.formatAudi1 [6][2]= "# ";s.formatAudi1 [10][3]= "# ";s.formatAudi1 [5][4]= "# ";

s.formatAudi1 [7][2]= "# ";s.formatAudi1 [11][3]= "# ";s.formatAudi1 [6][4]= "# ";

s.formatAudi1 [12][2]= "# ";s.formatAudi1 [12][3]= "# ";s.formatAudi1 [7][4]= "# ";

s.formatAudi1 [13][2]= "# ";s.formatAudi1 [14][3]= "# ";s.formatAudi1 [13][4]= "# ";

s.formatAudi1 [4][3]= "# ";s.formatAudi1 [15][3]= "# ";s.formatAudi1 [14][4]= "# ";

s.formatAudi1 [6][3]= "# ";s.formatAudi1 [4][4]= "# ";s.formatAudi1 [15][4]= "# ";

s.formatAudi1 [7][3]= "# ";

s.formatAudi1 [3][6]= "# ";s.formatAudi1 [3][7]= "# ";s.formatAudi1 [3][8]= "# ";

s.formatAudi1 [4][6]= "# ";s.formatAudi1 [6][7]= "# ";s.formatAudi1 [4][8]= "# ";

s.formatAudi1 [7][6]= "# ";s.formatAudi1 [8][7]= "# ";s.formatAudi1 [6][8]= "# ";

s.formatAudi1 [8][6]= "# ";s.formatAudi1 [9][7]= "# ";s.formatAudi1 [7][8]= "# ";

s.formatAudi1 [9][6]= "# ";s.formatAudi1 [10][7]= "# ";s.formatAudi1 [8][8]= "# ";

s.formatAudi1 [10][6]= "# ";s.formatAudi1 [11][7]= "# ";s.formatAudi1 [9][8]= "# ";

s.formatAudi1 [13][6]= "# ";s.formatAudi1 [13][7]= "# ";s.formatAudi1 [14][8]= "# ";

s.formatAudi1 [14][6]= "# ";s.formatAudi1 [14][7]= "# ";s.formatAudi1 [15][8]= "# ";

s.formatAudi1 [2][9]= "# ";s.formatAudi1 [7][9]= "# ";s.formatAudi1 [13][9]= "# ";

s.formatAudi1 [3][9]= "# ";s.formatAudi1 [8][9]= "# ";s.formatAudi1 [14][9]= "# ";

s.formatAudi1 [4][9]= "# ";s.formatAudi1 [9][9]= "# ";

s.formatAudi1 [1][1] = " "; s.formatAudi1 [7][1] = "6 "; s.formatAudi1 [12][1] = "11";

s.formatAudi1 [2][1] = "1 "; s.formatAudi1 [8][1] = "7 "; s.formatAudi1 [13][1] = "12";

s.formatAudi1 [3][1] = "2 "; s.formatAudi1 [9][1] = "8 "; s.formatAudi1 [14][1] = "13";

s.formatAudi1 [4][1] = "3 "; s.formatAudi1 [10][1] = "9 "; s.formatAudi1 [15][1] = "14";

s.formatAudi1 [5][1] = "4 "; s.formatAudi1 [11][1] = "10"; s.formatAudi1 [16][1] = " ";

s.formatAudi1 [6][1] = "5 ";

s.formatAudi1 [1][11] = " ";s.formatAudi1 [7][11] = " ";s.formatAudi1 [12][11] = " ";

s.formatAudi1 [2][11] = " ";s.formatAudi1 [8][11] = " ";s.formatAudi1 [13][11] = " ";

s.formatAudi1 [3][11] = " ";s.formatAudi1 [9][11] = " ";s.formatAudi1 [14][11] = " ";

s.formatAudi1 [4][11] = " ";s.formatAudi1 [10][11] = " ";s.formatAudi1 [15][11] = " ";

s.formatAudi1 [5][11] = " ";s.formatAudi1 [11][11] = " ";s.formatAudi1 [16][11] = " ";

s.formatAudi1 [6][11] = " ";

s.formatAudi1 [1][12] = "\*\*";s.formatAudi1 [7][12] = "\*\*";s.formatAudi1 [12][12] = "\*\*";

s.formatAudi1 [2][12] = "\*\*";s.formatAudi1 [8][12] = "\*\*";s.formatAudi1 [13][12] = "\*\*";

s.formatAudi1 [3][12] = "\*\*";s.formatAudi1 [9][12] = "\*\*";s.formatAudi1 [14][12] = "\*\*";

s.formatAudi1 [4][12] = "\*\*";s.formatAudi1 [10][12] = "\*\*";s.formatAudi1 [15][12] = "\*\*";

s.formatAudi1 [5][12] = "\*\*";s.formatAudi1 [11][12] = "\*\*";s.formatAudi1 [16][12] = "\*\*";

s.formatAudi1 [6][12] = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

s.formatAudi1 [1][0] = " ";s.formatAudi1 [7][0] = "";s.formatAudi1 [12][0] = "";

s.formatAudi1 [2][0] = " ";s.formatAudi1 [8][0] = "";s.formatAudi1 [13][0] = "";

s.formatAudi1 [3][0] = " ";s.formatAudi1 [9][0] = "";s.formatAudi1 [14][0] = "";

s.formatAudi1 [4][0] = " ";s.formatAudi1 [10][0] = "";s.formatAudi1 [15][0] = "";

s.formatAudi1 [5][0] = "";s.formatAudi1 [11][0] = "";s.formatAudi1 [16][0] = "";

s.formatAudi1 [6][0] = "PLATINUM ( 240 /- ) ";

s.formatAudi1 [1][5] = " ";s.formatAudi1 [7][5] = "";s.formatAudi1 [12][5] = "";

s.formatAudi1 [2][5] = " ";s.formatAudi1 [8][5] = "";s.formatAudi1 [13][5] = "";

s.formatAudi1 [3][5] = " ";s.formatAudi1 [9][5] = "";s.formatAudi1 [14][5] = "";

s.formatAudi1 [4][5] = " ";s.formatAudi1 [10][5] = "";s.formatAudi1 [15][5] = "";

s.formatAudi1 [5][5] = "";s.formatAudi1 [11][5] = "";s.formatAudi1 [16][5] = "";

s.formatAudi1 [6][5] = "EXECUTIVE ( 190 /- )";

s.formatAudi1 [1][13] = "\*\*";s.formatAudi1 [7][13] = "";s.formatAudi1 [12][13] = "";

s.formatAudi1 [2][13] = "\*\*\*";s.formatAudi1 [8][13] = "";s.formatAudi1 [13][13] = "";

s.formatAudi1 [3][13] = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

s.formatAudi1 [9][13] = "";s.formatAudi1 [14][13] = "\*";

s.formatAudi1 [4][13] = " SCREEN THIS WAY ";

s.formatAudi1 [10][13] = "";s.formatAudi1 [15][13] = "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

s.formatAudi1 [5][13] = "";s.formatAudi1 [11][13] = "";s.formatAudi1 [16][13] = "\*\*";

s.formatAudi1 [6][13] = "";

s.formatAudi1 [16][0] = ""; s.formatAudi1 [16][1] = ""; s.formatAudi1 [16][2] = "";

s.formatAudi1 [16][3] = ""; s.formatAudi1 [16][4] = ""; s.formatAudi1 [16][5] = "";

s.formatAudi1 [16][6] = ""; s.formatAudi1 [16][7] = ""; s.formatAudi1 [16][8] = "";

s.formatAudi1 [16][9] = ""; s.formatAudi1 [16][10] = "";s.formatAudi1 [16][11] = "";

s.formatAudi1 [16][12] = "";s.formatAudi1 [16][13] = "";

s.formatAudi1 [8][2] = " ";s.formatAudi1 [8][3] = " ";s.formatAudi1 [8][4] = " ";

s.formatAudi1 [9][2] = " ";s.formatAudi1 [9][3] = " ";s.formatAudi1 [9][4] = " ";

s.formatAudi1 [5][6] = " ";s.formatAudi1 [5][8] = " ";s.formatAudi1 [5][10] = " ";

s.formatAudi1 [5][7] = " ";s.formatAudi1 [5][9] = " ";

s.formatAudi1 [12][6] = " ";s.formatAudi1 [12][8] = " ";s.formatAudi1 [12][10] = " ";

s.formatAudi1 [12][7] = " ";s.formatAudi1 [12][9] = " ";

s.formatAudi1 [1][1] = " ";s.formatAudi1 [1][5] = " ";s.formatAudi1 [1][8] = " ";

s.formatAudi1 [1][2] = " ";s.formatAudi1 [1][6] = " ";s.formatAudi1 [1][9] = " ";

s.formatAudi1 [1][3] = " ";s.formatAudi1 [1][7] = " ";s.formatAudi1 [1][10] = " ";

s.formatAudi1 [1][4] = " ";

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

return 1;

**}**

**public int Audi\_3 ()**

**{**

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

for(int i = 0, j = 0 , c = 0;c<144;++c)

{ if(i%2==0)

s.formatAudi3 [i][j] = "[\_\_\_]";

else

s.formatAudi3 [i][j] = " ";

if(i==11) {i=0; ++j;}else{ ++i;} }

s.formatAudi3 [0][0]=" ";s.formatAudi3 [0][4]="C ";s.formatAudi3 [0][8]="F ";

s.formatAudi3 [0][1]=" ";s.formatAudi3 [0][5]=" ";s.formatAudi3 [0][9]=" ";

s.formatAudi3 [0][2]="A ";s.formatAudi3 [0][6]="D ";s.formatAudi3 [0][10]="\*\*\*\*\*";

s.formatAudi3 [0][3]="B ";s.formatAudi3 [0][7]="E ";s.formatAudi3 [0][11]="\*\*\*\*\*";

s.formatAudi3 [2][1]=" 1 ";s.formatAudi3 [6][1]=" 3 ";s.formatAudi3 [10][1]=" 5 ";

s.formatAudi3 [4][1]=" 2 ";s.formatAudi3 [8][1]=" 4 ";

s.formatAudi3 [2][9]=" ";s.formatAudi3 [6][9]=" ";s.formatAudi3 [10][9]=" ";

s.formatAudi3 [4][9]=" ";s.formatAudi3 [8][9]=" ";

s.formatAudi3 [1][10]="\*\*\*\*\*";s.formatAudi3 [5][10]="\*\*\*\*\*";s.formatAudi3 [9][10]="\*\*\*\*\*";

s.formatAudi3 [2][10]="\*\*\*\*\*";s.formatAudi3 [6][10]="\*\*\*\*\*";s.formatAudi3 [10][10]="\*\*\*\*\*";

s.formatAudi3 [3][10]="\*\*\*\*\*";s.formatAudi3 [7][10]="\*\*\*";s.formatAudi3 [11][10]="\*\*\*\*\*";

s.formatAudi3 [4][10]="\*\*\*\*\*\*\*";s.formatAudi3 [8][10]="\*\*\*\*\*\*\*";

s.formatAudi3 [1][11]="\*\*\*\*\*";s.formatAudi3 [5][11]="N THI";s.formatAudi3 [9][11]="\*\*\*\*\*";

s.formatAudi3 [2][11]="\*\*\*\*\*\*\*\*";s.formatAudi3 [6][11]="S WAY";s.formatAudi3 [10][11]="\*\*\*\*";

s.formatAudi3 [3][11]="\*\*\*\* ";s.formatAudi3 [7][11]=" \*\*\*\*\*";s.formatAudi3 [11][11]="\*\*";

s.formatAudi3 [4][11]="SCREE";s.formatAudi3 [8][11]="\*\*\*\*\*\*\*";

s.formatAudi3 [1][0]=" ";s.formatAudi3 [5][0]="CLASS";s.formatAudi3 [9][0]=" ";

s.formatAudi3 [2][0]=" ";s.formatAudi3 [6][0]="(1200/-)";s.formatAudi3 [10][0]=" ";

s.formatAudi3 [3][0]=" "; s.formatAudi3 [7][0]=" ";s.formatAudi3 [11][0]=" ";

s.formatAudi3 [4][0]="GOLD ";s.formatAudi3 [8][0]=" ";

s.formatAudi3 [1][5]=" ";s.formatAudi3 [5][5]="CLASS";s.formatAudi3 [9][5]=" ";

s.formatAudi3 [2][5]=" ";s.formatAudi3 [6][5]="(1000/-)";s.formatAudi3 [10][5]=" ";

s.formatAudi3 [3][5]=" "; s.formatAudi3 [7][5]=" ";s.formatAudi3 [11][5]=" ";

s.formatAudi3 [4][5]="GOLD ";s.formatAudi3 [8][5]=" ";

s.formatAudi3 [4][2]="[[#]]";s.formatAudi3 [10][6]="[[#]]";s.formatAudi3 [6][7]="[[#]]";

s.formatAudi3 [6][2]="[[#]]";s.formatAudi3 [8][6]="[[#]]";s.formatAudi3 [4][7]="[[#]]";

s.formatAudi3 [8][2]="[[#]]";s.formatAudi3 [6][6]="[[#]]";s.formatAudi3 [2][7]="[[#]]";

s.formatAudi3 [10][2]="[[#]]";s.formatAudi3 [4][6]="[[#]]";s.formatAudi3 [8][8]="[[#]]";

s.formatAudi3 [4][4]="[[#]]";s.formatAudi3 [6][4]="[[#]]";s.formatAudi3 [6][8]="[[#]]";

try { FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{ Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1; }

return 1; **}** }

**Class: Update**

import java.util.Scanner;

import java.io.\*;

**public class Update**

**{**

**public int Edit (int temp)**

**{**

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

Scanner Sc = new Scanner ( System.in );

String Choice;

outer :

while ( true )

{

System.out.print("\u000C");

System.out.println("A. Edit Movie Name");

System.out.println("B. Edit Movie Language");

System.out.println("C. Edit Movie Duration");

System.out.println("D. Exit");

Choice = Sc.nextLine();

System.out.print("\u000C");

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT"))||(Choice.equalsIgnoreCase("D")))

{

System.out.print("\u000C");

break outer;

}

else if (Choice.equalsIgnoreCase("A"))

{

System.out.println("Enter new movie name");

Choice = Sc.nextLine();

s.movieList[temp][0]=Choice;

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT")))

{

System.out.print("\u000C");

continue;

}

else

;

System.out.print("\u000C");

System.out.println("Changed movie name sucessfully");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

continue;

}

else if (Choice.equalsIgnoreCase("B"))

{

System.out.println("Enter new movie language");

Choice = Sc.nextLine();

s.movieList[temp][1]=Choice;

System.out.print("\u000C");

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT")))

{

System.out.print("\u000C");

continue;

}

else

;

System.out.println("Changed movie language sucessfully");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

continue;

}

else if (Choice.equalsIgnoreCase("C"))

{

System.out.println("Enter new movie duration");

Choice = Sc.nextLine();

s.movieList[temp][2]=Choice;

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT")))

{

System.out.print("\u000C");

continue;

}

else

;

System.out.print("\u000C");

System.out.println("Changed movie duration sucessfully");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

continue;

}

else

{

System.out.println("ERROR:506");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

continue;

}

}

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

return 1;

**}**

**public int DeleteShow (int code)**

**{**

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

Scanner Sc = new Scanner ( System.in );

int d4 = 0;

int d;

int d2;

int d3;

d3 = Integer.valueOf(s.movieList [code][2]);

for(int i = 60,i2 =0;i<1380;i=i+60,++i2)

{

if(d3<=i)

{;

}

else

{

d4 = 40\*i2;

break;

}

}

System.out.println("Please enter the day for which you want to delete the timings for");

System.out.println("1. Monday");

System.out.println("2. Tuesday");

System.out.println("3. Wednesday");

System.out.println("4. Thursday");

System.out.println("5. Friday");

System.out.println("6. Saturday");

System.out.println("7. Sunday");

System.out.println("8. Exit");

String Choice = Sc.nextLine();

System.out.print("\u000C");

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT"))||(Choice.equalsIgnoreCase("8")))

{

System.out.print("\u000C");

return 1;

}

if(Choice.equalsIgnoreCase("Monday")||Choice.equalsIgnoreCase("1"))

{

for(int i = 0;i<2359;++i)

{ if((s.D1 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D1 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D1 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Please enter Audi no. from which the show is to be deleted(Integer Value Only))");

d2 = Sc.nextInt();

System.out.println("Please enter time of the show which is to be deleted(Integer Value Only)");

d = Sc.nextInt();

s.D1 [d][d2] = "";

for(int i = d2+1;i<=(d+d3+d4);++i)

{ s.D6 [i][d2] = code+".";

}

System.out.println("Movie Deleted");

}

else if(Choice.equalsIgnoreCase("Tuesday")||Choice.equalsIgnoreCase("2"))

{

for(int i = 0;i<2359;++i)

{ if((s.D2 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D2 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D2 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Please enter Audi no. from which the show is to be deleted");

d2 = Sc.nextInt();

System.out.println("Please enter time of the show which is to be deleted");

d = Sc.nextInt();

for(int i = d2+1;i<=(d+d3+d4);++i)

{ s.D6 [i][d2] = code+".";

}

System.out.println("Movie Deleted");

}

else if(Choice.equalsIgnoreCase("Wednesday")||Choice.equalsIgnoreCase("3"))

{

for(int i = 0;i<2359;++i)

{ if((s.D3 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D3 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D3 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Please enter Audi no. from which the show is to be deleted");

d2 = Sc.nextInt();

System.out.println("Please enter time of the show which is to be deleted");

d = Sc.nextInt();

for(int i = d2+1;i<=(d+d3+d4);++i)

{ s.D6 [i][d2] = code+".";

}

System.out.println("Movie Deleted");

}

else if(Choice.equalsIgnoreCase("Thursday")||Choice.equalsIgnoreCase("4"))

{

for(int i = 0;i<2359;++i)

{ if((s.D4 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D4 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D4 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Please enter Audi no. from which the show is to be deleted");

d2 = Sc.nextInt();

System.out.println("Please enter time of the show which is to be deleted");

d = Sc.nextInt();

for(int i = d2+1;i<=(d+d3+d4);++i)

{ s.D6 [i][d2] = code+".";

}

System.out.println("Movie Deleted");

}

else if(Choice.equalsIgnoreCase("Friday")||Choice.equalsIgnoreCase("5"))

{

for(int i = 0;i<2359;++i)

{ if((s.D5 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D5 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D5 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Please enter Audi no. from which the show is to be deleted");

d2 = Sc.nextInt();

System.out.println("Please enter time of the show which is to be deleted");

d = Sc.nextInt();

for(int i = d2+1;i<=(d+d3+d4);++i)

{ s.D6 [i][d2] = code+".";

}

System.out.println("Movie Deleted");

}

else if(Choice.equalsIgnoreCase("Saturday")||Choice.equalsIgnoreCase("6"))

{

for(int i = 0;i<2359;++i)

{ if((s.D6 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D6 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D6 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Please enter Audi no. from which the show is to be deleted");

d2 = Sc.nextInt();

System.out.println("Please enter time of the show which is to be deleted");

d = Sc.nextInt();

for(int i = d2+1;i<=(d+d3+d4);++i)

{ s.D6 [i][d2] = code+".";

}

System.out.println("Movie Deleted");

}

else if(Choice.equalsIgnoreCase("Sunday")||Choice.equalsIgnoreCase("7"))

{

for(int i = 0;i<2359;++i)

{ if((s.D7 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D7 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D7 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Please enter Audi no. from which the show is to be deleted");

d2 = Sc.nextInt();

System.out.println("Please enter time of the show which is to be deleted");

d = Sc.nextInt();

for(int i = d2+1;i<(d+d3+d4);++i)

{ s.D6 [i][d2] = ".";

}

System.out.println("Show Deleted");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

}

else

{ System.out.println("ERROR");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

return 1;

**}**

**public int AddShow (int code, int duration)**

**{**

Scanner Sc = new Scanner ( System.in );

String Choice;

int Choice2;

int Choice3;

int d =0;

int Choice20;

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

int audi = 0;

int hr = 0;

int min = 0;

System.out.println("Please enter the day for which you want to add timings");

System.out.println("1. Monday");

System.out.println("2. Tuesday");

System.out.println("3. Wednesday");

System.out.println("4. Thursday");

System.out.println("5. Friday");

System.out.println("6. Saturday");

System.out.println("7. Sunday");

System.out.println("8. Exit");

Choice = Sc.nextLine();

System.out.print("\u000C");

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT"))||(Choice.equalsIgnoreCase("8")))

{

System.out.print("\u000C");

return 1;

}

System.out.println("Please enter the Audi number");

Choice3 = Sc.nextInt();

if(audi >= 3)

{

System.out.println("No Such Audi Exists");

System.out.println("Please try again");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

else

{ ;

}

System.out.print("\u000C");

System.out.println("Please enter the Start time in hrs ( Ex. 1420 for 2:20 pm )");

Choice2 = Sc.nextInt();

System.out.print("\u000C");

Choice20 = Integer.valueOf(Choice2);

if((Choice20 > 2200)||((Choice20%100)>59))

{

System.out.println("No Movie After 10 pm allowed");

System.out.println("Please try again");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

else

{

;

}

d=(duration/60)\*40;

if(Choice.equalsIgnoreCase("Monday")||Choice.equalsIgnoreCase("1"))

{

if (s.D1 [Choice2][Choice3].equals("")&&s.D1 [Choice2+duration][Choice3].equals("")){

s.D1 [Choice2][Choice3] = code+".S.";

s.D1 [Choice2+duration+d][Choice3] = code+".E";

for(int i = Choice2+1;i<(Choice2+duration+d);++i)

{ s.D1 [i][Choice3] = code+".";

}

}

else

{ System.out.println("Time Slot already Booked");

System.out.println("Please Try Again");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

else if(Choice.equalsIgnoreCase("Tuesday")||Choice.equalsIgnoreCase("2"))

{

if (s.D2 [Choice2][Choice3].equals("")&&s.D2 [Choice2+duration][Choice3].equals("")){

s.D2 [Choice2][Choice3] = code+".S.";

s.D2 [Choice2+duration+d][Choice3] = code+".E";

for(int i = Choice2+1;i<(Choice2+duration+d);++i)

{ s.D2 [i][Choice3] = code+".";

}

}

else

{ System.out.println("Time Slot already Booked");

System.out.println("Please Try Again");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

else if(Choice.equalsIgnoreCase("Wednesday")||Choice.equalsIgnoreCase("3"))

{

if (s.D3 [Choice2][Choice3].equals("")&&s.D3 [Choice2+duration][Choice3].equals("")){

s.D3 [Choice2][Choice3] = code+".S.";

s.D3 [Choice2+duration+d][Choice3] = code+".E";

for(int i = Choice2+1;i<(Choice2+duration+d);++i)

{ s.D3 [i][Choice3] = code+".";

}

}

else

{ System.out.println("Time Slot already Booked");

System.out.println("Please Try Again");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

else if(Choice.equalsIgnoreCase("Thursday")||Choice.equalsIgnoreCase("4"))

{

if (s.D4 [Choice2][Choice3].equals("")&&s.D4 [Choice2+duration][Choice3].equals("")){

s.D4 [Choice2][Choice3] = code+".S.";

s.D4 [Choice2+duration+d][Choice3] = code+".E";

for(int i = Choice2+1;i<(Choice2+duration+d);++i)

{ s.D4 [i][Choice3] = code+".";

}

}

else

{ System.out.println("Time Slot already Booked");

System.out.println("Please Try Again");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

else if(Choice.equalsIgnoreCase("Friday")||Choice.equalsIgnoreCase("5"))

{

if (s.D5 [Choice2][Choice3].equals("")&&s.D5 [Choice2+duration][Choice3].equals("")){

s.D5 [Choice2][Choice3] = code+".S.";

s.D5 [Choice2+duration+d][Choice3] = code+".E";

for(int i = Choice2+1;i<(Choice2+duration+d);++i)

{ s.D5 [i][Choice3] = code+".";

}

}

else

{ System.out.println("Time Slot already Booked");

System.out.println("Please Try Again");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

else if(Choice.equalsIgnoreCase("Saturday")||Choice.equalsIgnoreCase("6"))

{

if (s.D6 [Choice2][Choice3].equals("")&&s.D6 [Choice2+duration][Choice3].equals("")){

s.D6 [Choice2][Choice3] = code+".S.";

s.D6 [Choice2+duration+d][Choice3] = code+".E";

for(int i = Choice2+1;i<(Choice2+duration+d);++i)

{ s.D6 [i][Choice3] = code+".";

}

}

else

{ System.out.println("Time Slot already Booked");

System.out.println("Please Try Again");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

else if(Choice.equalsIgnoreCase("Sunday")||Choice.equalsIgnoreCase("7"))

{

if (s.D7 [Choice2][Choice3].equals("")&&s.D7 [Choice2+duration][Choice3].equals("")){

s.D7 [Choice2][Choice3] = code+".S.";

s.D7 [Choice2+duration+d][Choice3] = code+".E";

for(int i = Choice2+1;i<(Choice2+duration+d);++i)

{ s.D7 [i][Choice3] = code+".";

}

}

else

{ System.out.println("Time Slot already Booked");

System.out.println("Please Try Again");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

else

{ System.out.println("ERROR");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

System.out.println("Timing stored");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

**}**

**public int ShowTime (int code)**

**{**

Scanner Sc = new Scanner ( System.in );

String Choice;

Outer :

while ( true )

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

System.out.println("Please enter the day for which you want to see the timings");

System.out.println("1. Monday");

System.out.println("2. Tuesday");

System.out.println("3. Wednesday");

System.out.println("4. Thursday");

System.out.println("5. Friday");

System.out.println("6. Saturday");

System.out.println("7. Sunday");

System.out.println("8. Exit");

Choice = Sc.nextLine();

System.out.print("\u000C");

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT"))||(Choice.equalsIgnoreCase("8")))

{

System.out.print("\u000C");

return 1;

}

if(Choice.equalsIgnoreCase("Monday")||Choice.equalsIgnoreCase("1"))

{

for(int i = 0;i<2359;++i)

{ if((s.D1 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D1 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D1 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Outer;

}

else if(Choice.equalsIgnoreCase("Tuesday")||Choice.equalsIgnoreCase("2"))

{

for(int i = 0;i<2359;++i)

{ if((s.D2 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D2 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D2 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Outer;

}

else if(Choice.equalsIgnoreCase("Wednesday")||Choice.equalsIgnoreCase("3"))

{

for(int i = 0;i<2359;++i)

{ if((s.D3 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else ;

if((s.D3 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else ;

if((s.D3 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else ; }

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Outer; }

else if(Choice.equalsIgnoreCase("Thursday")||Choice.equalsIgnoreCase("4"))

{

for(int i = 0;i<2359;++i)

{ if((s.D4 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else ;

if((s.D4 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else ;

if((s.D4 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else ; }

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Outer; }

else if(Choice.equalsIgnoreCase("Friday")||Choice.equalsIgnoreCase("5"))

{

for(int i = 0;i<2359;++i)

{ if((s.D5 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else ;

if((s.D5 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else ;

if((s.D5 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else ; }

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Outer;

}

else if(Choice.equalsIgnoreCase("Saturday")||Choice.equalsIgnoreCase("6"))

{

for(int i = 0;i<2359;++i)

{ if((s.D6 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D6 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else ;

if((s.D6 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else ;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Outer;

}

else if(Choice.equalsIgnoreCase("Sunday")||Choice.equalsIgnoreCase("7"))

{

for(int i = 0;i<2359;++i)

{ if((s.D7 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else ;

if((s.D7 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else ;

if((s.D7 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else ;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Outer;

}

else

{ System.out.println("ERROR");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

}

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

} **}**

**}**

**Class: Login**

import java.util.Scanner;

public class Login

{

public boolean login ()

{

String ID = "i";

String Password = "i";

String Choice;

System.out.print("\u000C");

Scanner Sc = new Scanner ( System.in );

boolean result = false;

String PasswordIN ;

String IDIN ;

System.out.println("Please enter USER ID:");

outer :

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

{

IDIN = Sc.nextLine();

System.out.print("\u000C");

if(IDIN.equalsIgnoreCase(ID))

{

System.out.print("\u000C");

System.out.println("Please enter password");

inner :

for(int i2 = 0;i2<=3;++i)

{

PasswordIN = Sc.nextLine();

System.out.print("\u000C");

if(PasswordIN.equals(Password))

{

result = true;

System.out.println("USER VERIFIED");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

break outer;

}

else if (PasswordIN.equalsIgnoreCase("EXIT"))

{

System.out.println("Verification Failed");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return result;

}

else if(PasswordIN.equalsIgnoreCase("E"))

{

System.out.print("\u000C");

break outer;

}

else

{

System.out.println("Wrong password");

}

System.out.println("Please enter password again");

if(i2==2)

{

System.out.print("\u000C");

System.out.println("Too many incorrect tries");

System.out.println("System locked ");

System.out.println("System will resume after a few moments");

System.out.println("Please do not enter anything, doing so will close the program");

try{

Thread.sleep(10000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

i2 = 0;

System.out.println("Enter 'E' to exit the program");

System.out.println("Enter any character to continue");

Choice = Sc.next();

System.out.print("\u000C");

if(Choice.equalsIgnoreCase("E"))

{

System.out.print("\u000C");

break outer;

}

else if (Choice.equalsIgnoreCase("EXIT"))

{

System.out.println("Verification Failed");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return result;

}

else

{

continue inner;

}

}

}

}

else if (IDIN.equalsIgnoreCase("EXIT"))

{

System.out.println("Verification Failed");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return result;

}

else if (IDIN.equalsIgnoreCase("E"))

{

System.out.println("Verification Failed");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return result;

}

else

{

{

System.out.println("Wrong USER ID");

}

System.out.println("Please enter USER ID again");

if(i==2)

{

System.out.print("\u000C");

System.out.println("Too many incorrect tries");

System.out.println("System locked ");

System.out.println("System will resume after a few moments");

System.out.println("Please do not enter anything, doing so will close the program");

try{

Thread.sleep(30000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

i = 0;

System.out.println("Enter 'E' to exit program");

System.out.println("Enter any character to continue");

Choice = Sc.next();

System.out.print("\u000C");

if(Choice.equalsIgnoreCase("E"))

{System.out.print("\u000C");

break outer;

}

else if (Choice.equalsIgnoreCase("EXIT"))

{

System.out.println("Verification Failed");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return result;

}

else

{

continue outer;

}

}

}

}

if(result==false)

{

System.out.println("Verification Failed");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

}

else if (result==true) {;}return result;}

}

**Class: SystemUpdate**

public class SystemUpdate

{

public int systemupdate ()

{

Screen S = new Screen ();

AddMovie AM = new AddMovie ();

UpdateMovie UM = new UpdateMovie();

DeleteMovie DM = new DeleteMovie();

String Choice;

int a = 0;

while(true)

{

Choice = S.Screen\_2();

if(Choice.equalsIgnoreCase("A"))

{

a = AM.add();

continue;

}

else if(Choice.equalsIgnoreCase("B"))

{

a = UM.update();

continue;

}

else if(Choice.equalsIgnoreCase("C"))

{

a = DM.delete();

continue;

}

else if(Choice.equalsIgnoreCase("D"))

{

break;

}

else if (Choice.equalsIgnoreCase("E")||Choice.equalsIgnoreCase("EXIT"))

{

break;

}

else

{

System.out.println("Please enter a valid choice");

continue;

}

}

return a;

}

}

**Class: AddMovie**

import java.util.Scanner;

import java.io.\*;

public class AddMovie

{

public int add ()

{

Scanner Sc = new Scanner ( System.in );

String Choice;

String Movie;

String Language;

String Duration;

int temp = -1;

int temp2 = -1;

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

outer :

while(true)

{

System.out.println("Please enter the name of the movie, to be added");

Movie = Sc.nextLine();

System.out.print("\u000C");

if((Movie.equalsIgnoreCase("E"))||(Movie.equalsIgnoreCase("EXIT")))

{

return 1;

}

else

;

inner :

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

{

if((Movie).equalsIgnoreCase(s.movieList[i][0]))

{

System.out.println("This Movie is already playing");

System.out.println("A. Add Anyway");

System.out.println("B. Overwrite Movie");

System.out.println("c. Cancel");

Choice = Sc.nextLine();

if(Choice.equalsIgnoreCase("a"))

{

break inner;

}

else if (Choice.equalsIgnoreCase("B"))

{

temp2 = i;

break inner;

}

else if (Choice.equalsIgnoreCase("C"))

{

return 1;

}

else

{

System.out.println("Key not found");

}

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

continue outer;

}

else

;

}

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

{

if("".equals((s.movieList[i][0])))

{temp = i;

break;

}

else

;

}

if(temp == -1 )

{

System.out.println("No space available");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

else

;

System.out.print("\u000C");

System.out.println("Please enter the language");

Language = Sc.nextLine();

System.out.print("\u000C");

if(Language.equalsIgnoreCase("E")||Language.equalsIgnoreCase("EXIT"))

{

return 1;

}

else

;

System.out.println("Please enter the duration of the movie in minutes");

Duration = Sc.nextLine();

System.out.print("\u000C");

if(Duration.equalsIgnoreCase("E")||Duration.equalsIgnoreCase("EXIT"))

{

return 1;

}

else

break outer;

}

if(temp2 == -1 )

{

s.movieList[temp][0] = Movie;

}

else

{s.movieList[temp2][0] = Movie;}

s.movieList[temp][1] = Language;

s.movieList[temp][2] = Duration;

s.movieList[temp][3] = Integer.toString(s.code);

s.code = s.code+1;

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

System.out.println("Movie Sucessfully stored");

System.out.println("Please update/add movie show timings in Update Movie/Show Timings section");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

**Class: DeleteMovie**

import java.util.Scanner;

import java.io.\*;

public class DeleteMovie

{

public int delete ()

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

Scanner Sc = new Scanner ( System.in );

String Choice;

char a = 'A';

String b = a+". " ;

int arr [] = new int [50];

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

{

arr[i]=-1;

}

for ( int i =0,j=0,c=0,d=0;c<49;++c)

{

if(!s.movieList[i][j].equals(""))

{

if((j==0) && (i>0))

{

System.out.print(b);

++a;

b=a+". ";

arr[d]=i;

++d;

}

else

{

}

System.out.print(s.movieList[i][j]+"\t");

System.out.print("\t|");

if(j==2)

{

++i;

j=0;

System.out.println();

}

else

{++j; }

}

else

{ ++i; }

}

System.out.println();

System.out.println("Please enter the movie you want to delete");

Choice = Sc.nextLine();

System.out.print("\u000C");

if(Choice.equalsIgnoreCase("EXIT"))

{

return 1;

}

else

;

a = 'A';

int temp = -1;

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

{

b = a+"";

if(Choice.equalsIgnoreCase(b))

{

temp = arr[i];

break;

}

else

{ ++a;}

}

if(temp != -1 )

{

System.out.println("Are you sure you want to delete "+(s.movieList[temp][0])+" ?");

System.out.println("A. Yes");

System.out.println("B. No, exit this menu");

Choice = Sc.nextLine();

}

else

{

System.out.println("ERROR:200");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

if(Choice.equalsIgnoreCase("A")||Choice.equalsIgnoreCase("YES"))

{ ; }

else if(Choice.equalsIgnoreCase("b")||Choice.equalsIgnoreCase("No"))

{

System.out.println("Movie Not deleted");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

else if(Choice.equalsIgnoreCase("e")||Choice.equalsIgnoreCase("exit"))

{

System.out.println("Movie Not deleted");

try{ Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

else

{

System.out.println("Key not found");

System.out.println("Movie Not deleted");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

s.movieList[temp][1] = “"; s.movieList[temp][2] = “"; s.movieList[temp][0] = "";

s.movieList[temp][3] = "";

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

System.out.println("Movie sucessfully deleted");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

**Class: UpdateMovie**

import java.util.Scanner;

import java.io.\*;

public class UpdateMovie

{

public int update ()

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

Scanner Sc = new Scanner ( System.in );

String Choice;

int temp = -1;

char a = 'A';

String b = a+". ";

int arr [] = new int [50];

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

{

arr[i]=-1;

}

for ( int i =0,j=0,c=0,d=0;c<49;++c)

{

if(!s.movieList[i][j].equals(""))

{

if((j==0) && (i>0))

{

System.out.print(b);

++a;

b=a+". ";

arr[d]=i;

++d;

}

else

{

}

System.out.print(s.movieList[i][j]+"\t");

System.out.print("\t|");

if(j==2)

{

++i;

j=0;

System.out.println();

}

else

{

++j;

}

}

else

{

++i;

}

}

System.out.println();

System.out.println("Please enter the movie you want to Update ");

Choice = Sc.nextLine();

System.out.print("\u000C");

if((Choice.equalsIgnoreCase("EXIT")))

{

System.out.print("\u000C");

return 1;

}

else

;

a = 'A';

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

{

b = a+"";

if(Choice.equalsIgnoreCase(b))

{

temp = arr[i];

break;

}

else

{

++a;

}

}

if(temp == -1)

{

System.out.println("ERROR:200");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

else

{

System.out.print("\u000C");

}

int temp2 = Integer.valueOf(s.movieList[temp][3]);

int temp3 = Integer.valueOf(s.movieList[temp][2]);

System.out.print(" Mavie: "+s.movieList[temp][0]+" | Language: "+s.movieList[temp][1]);

System.out.print(" | Duration: "+s.movieList[temp][2]+" mins");

System.out.println();

System.out.println("A. Edit movie Name/ Language/ Duration");

System.out.println("B. Show existing movie timings");

System.out.println("C. Add Shows");

System.out.println("D. Delete Shows");

Choice = Sc.nextLine();

System.out.print("\u000C");

Update U = new Update ();

int hi;

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT")))

{

System.out.print("\u000C");

return 1;

}

else if (Choice.equalsIgnoreCase("A"))

{

hi = U.Edit(temp);

return 1;

}

else if (Choice.equalsIgnoreCase("B"))

{

hi = U.ShowTime(temp2);

return 1;

}

else if (Choice.equalsIgnoreCase("C"))

{

hi = U.AddShow(temp2,temp3);

return 1;

}

else if (Choice.equalsIgnoreCase("D"))

{

hi = U.DeleteShow(temp2);

return 1;

}

else

{

System.out.println("ERROR:506");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

}

**Class: Screen**

import java.util.Scanner;

public class Screen

{

String Choice;

public String Screen\_1 ()

{

Scanner Sc = new Scanner ( System.in);

System.out.print("\u000C");

System.out.println(" I CINEMAS");

System.out.println();

System.out.println(" A. System Update");

System.out.println(" B. List Of Movies With Show Timings");

System.out.println(" C. Book A Ticket");

System.out.println(" D. Exit");

System.out.println();

System.out.println("Please Enter Your Choice");

Choice = Sc.nextLine();

System.out.print("\u000C");

return Choice;

}

public String Screen\_2 ()

{

Scanner Sc = new Scanner ( System.in);

System.out.print("\u000C");

System.out.println(" System Update");

System.out.println();

System.out.println(" A. Add A New Movie");

System.out.println(" B. Update Movie / Show Timings");

System.out.println(" C. Delete A Movie");

System.out.println(" D. Exit");

System.out.println();

System.out.println("Please Enter Your Choice");

Choice = Sc.nextLine();

System.out.print("\u000C");

return Choice;

}

}

**Class: Book**

import java.util.\*;

import java.io.\*;

import java.text.SimpleDateFormat;

public class Book

{

static String Food [][] = new String [3][25];

public int book ()

{

Scanner Sc = new Scanner ( System.in);

Movie M = new Movie ();

Day D = new Day ();

Show\_Book S = new Show\_Book ();

Seat s = new Seat ();

Bill B = new Bill ();

int MovieCode = 0;

MovieCode = M.choice ();

if(MovieCode==0){return 1;}else{;}

int Day = 0;

Day = D.choice ();

int audi =0;

int time =0;

int re =0;

String AT = S.Show(MovieCode,Day);

if(AT.equals("NO")){return 1;}

audi = Integer.valueOf(AT.substring(0,1));

int l = AT.length();

time = Integer.valueOf(AT.substring(1,l));

String Price [][];

Price = s.SelectSeat(audi,Day,time,MovieCode);

Calendar calendar = Calendar.getInstance();

int day = calendar.get(Calendar.DAY\_OF\_WEEK);

int last = day - 1;

SimpleDateFormat sdf = new SimpleDateFormat("dd/MM");

Calendar c = Calendar.getInstance();

String Date = "01/01";

Date = sdf.format(c.getTime());

int diff = 0;

try{c.setTime(sdf.parse(Date));}catch(Exception E23){;}

if ( Day > last )

{

diff = Day - last;

}

else

{diff = Day + 7 - last;}

String dateToIncr = Date;

String dt="";

try { c.setTime(sdf.parse(dateToIncr));} catch ( Exception e) {}

c.add(Calendar.DAY\_OF\_MONTH, diff);

dt = sdf.format(c.getTime());

int i = B.Bill1(Price,Food,MovieCode,dt,time,audi,Day);

return 1;

}

}

**Class: Movie**

import java.util.Scanner;

import java.io.\*;

public class Movie

{

public int choice ()

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

Scanner Sc = new Scanner ( System.in );

String Choice;

char a = 'A';

String b = a+". " ;

int arr [] = new int [50];

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

{

arr[i]=-1;

}

for ( int i =0,j=0,c=0,d=0;c<49;++c)

{

if(!s.movieList[i][j].equals(""))

{

if((j==0) && (i>0))

{

System.out.print(b);

++a;

b=a+". ";

arr[d]=i;

++d;

}

else

{

}

System.out.print(s.movieList[i][j]+"\t");

System.out.print("\t|");

if(j==2)

{

++i;

j=0;

System.out.println();

}

else

{

++j;

}

}

else

{

++i;

}

}

System.out.println();

System.out.println("Enter \"exit\" to exit");

System.out.println("Please enter the movie ");

Choice = Sc.nextLine();

System.out.print("\u000C");

if(Choice.equalsIgnoreCase("EXIT"))

{

return 0;

}

else

;

a = 'A';

int temp = -1;

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

{

b = a+"";

if(Choice.equalsIgnoreCase(b))

{

temp = arr[i];

break;

}

else

{

++a;

}

}

//return Integer.valueOf(s.movieList[temp][3]);

return temp;

}

}

**Class: Show\_Book**

import java.io.\*;

import java.util.Scanner;

public class Show\_Book

{

public String Show ( int codeIn, int D )

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return "NO";

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return "NO";

}

Scanner Sc = new Scanner ( System.in);

int code = Integer.valueOf(s.movieList[codeIn][3]);

if(D==1)

{

for(int i = 0,d=0;i<2359;++i)

{ if((s.D1 [i][1]).startsWith(code+".S"))

{ System.out.println("-> Audi:1 Time:"+i+" Hrs.");}

else {;}

if((s.D1 [i][2]).startsWith(code+".S"))

{ System.out.println("-> Audi:2 Time:"+i+" Hrs.");}

else {;}

if((s.D1 [i][3]).startsWith(code+".S"))

{ System.out.println("-> Audi:3 Time:"+i+" Hrs.");}

else {;}

}

}

else if(D==2)

{

for(int i = 0,d=0;i<2359;++i)

{ if((s.D2 [i][1]).startsWith(code+".S"))

{ System.out.println("-> Audi:1 Time:"+i+" Hrs.");}

else {;}

if((s.D2 [i][2]).startsWith(code+".S"))

{ System.out.println("-> Audi:2 Time:"+i+" Hrs.");}

else {;}

if((s.D2 [i][3]).startsWith(code+".S"))

{ System.out.println("-> Audi:3 Time:"+i+" Hrs.");}

else {;}

}

}

else if(D==3)

{

for(int i = 0,d=0;i<2359;++i)

{ if((s.D3 [i][1]).startsWith(code+".S"))

{ System.out.println("-> Audi:1 Time:"+i+" Hrs.");}

else {;}

if((s.D3 [i][2]).startsWith(code+".S"))

{ System.out.println("-> Audi:2 Time:"+i+" Hrs.");}

else {;}

if((s.D3 [i][3]).startsWith(code+".S"))

{ System.out.println("-> Audi:3 Time:"+i+" Hrs.");}

else {;}

}

}

else if(D==4)

{

for(int i = 0,d=0;i<2359;++i)

{ if((s.D4 [i][1]).startsWith(code+".S"))

{ System.out.println("-> Audi:1 Time:"+i+" Hrs.");}

else {;}

if((s.D4 [i][2]).startsWith(code+".S"))

{ System.out.println("-> Audi:2 Time:"+i+" Hrs.");}

else {;}

if((s.D4 [i][3]).startsWith(code+".S"))

{ System.out.println("-> Audi:3 Time:"+i+" Hrs.");}

else {;}

}

}

else if(D==5)

{

for(int i = 0,d=0;i<2359;++i)

{ if((s.D5 [i][1]).startsWith(code+".S"))

{ System.out.println("-> Audi:1 Time:"+i+" Hrs.");}

else {;}

if((s.D5 [i][2]).startsWith(code+".S"))

{ System.out.println("-> Audi:2 Time:"+i+" Hrs.");}

else {;}

if((s.D5 [i][3]).startsWith(code+".S"))

{ System.out.println("-> Audi:3 Time:"+i+" Hrs.");}

else {;}

}

}

else if(D==6)

{

for(int i = 0,d=0;i<2359;++i)

{ if((s.D6 [i][1]).startsWith(code+".S"))

{ System.out.println("-> Audi:1 Time:"+i+" Hrs.");}

else {;}

if((s.D6 [i][2]).startsWith(code+".S"))

{ System.out.println("-> Audi:2 Time:"+i+" Hrs.");}

else {;}

if((s.D6 [i][3]).startsWith(code+".S"))

{ System.out.println("-> Audi:3 Time:"+i+" Hrs.");}

else {;}

}

}

else if(D==7)

{

for(int i = 0,d=0;i<2359;++i)

{ if((s.D7 [i][1]).startsWith(code+".S"))

{ System.out.println("-> Audi:1 Time:"+i+" Hrs.");}

else {;}

if((s.D7 [i][2]).startsWith(code+".S"))

{ System.out.println("-> Audi:2 Time:"+i+" Hrs.");}

else {;}

if((s.D7 [i][3]).startsWith(code+".S"))

{ System.out.println("-> Audi:3 Time:"+i+" Hrs.");}

else {;}

}

}

System.out.println("Please select audi number");

System.out.println("Please enter E to exit");

String Choice = Sc.nextLine();

if ( Choice.equalsIgnoreCase("E")||Choice.equalsIgnoreCase("Exit"))

{ return "NO";}

else{;}

System.out.println("Please select show timing in the format HHmm");

System.out.println("Please enter E to exit");

String Choice2 = Sc.nextLine();

if ( Choice.equalsIgnoreCase("E")||Choice.equalsIgnoreCase("Exit"))

{ return "NO";}

else{;}

String Choice3 = Choice + Choice2;

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return "NO";

}

return Choice3;

}

}

**Class: Day**

import java.util.\*;

import java.text.\*;

public class Day

{

public int choice ()

{

SimpleDateFormat df = new SimpleDateFormat("dd/MM");

Calendar cal = Calendar.getInstance();

int day = cal.get(Calendar.DAY\_OF\_WEEK);

day = day - 1;

Date myDate = new Date();

System.out.println("Please select the day you want to see the movie");

System.out.println();

if( day == 1 )

{

System.out.println("1. Monday " + df.format(myDate));myDate = addDays(myDate, 1);

System.out.println("2. Tuesday "+ df.format(myDate));myDate = addDays(myDate, 1);

System.out.println("3. Wednesday "+ df.format(myDate));myDate = addDays(myDate, 1);

System.out.println("4. Thursday "+ df.format(myDate));myDate = addDays(myDate, 1);

System.out.println("5. Friday "+ df.format(myDate));myDate = addDays(myDate, 1);

System.out.println("6. Saturday "+ df.format(myDate));myDate = addDays(myDate, 1);

System.out.println("7. Sunday "+ df.format(myDate));

}

else if ( day == 2)

{

myDate = addDays(myDate, 6);

System.out.println("1. Monday (" +df.format(myDate)+")");myDate = new Date();

System.out.println("2. Tuesday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("3. Wednesday ("+ df.format(myDate)+")");myDate=addDays(myDate,1);

System.out.println("4. Thursday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("5. Friday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("6. Saturday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("7. Sunday ("+ df.format(myDate)+")");

}

else if ( day == 3)

{

myDate = addDays(myDate, 5);

System.out.println("1. Monday (" +df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("2. Tuesday ("+ df.format(myDate)+")");myDate = new Date();

System.out.println("3. Wednesday ("+ df.format(myDate)+")");myDate=addDays(myDate,1);

System.out.println("4. Thursday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("5. Friday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("6. Saturday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("7. Sunday ("+ df.format(myDate)+")");

}

else if ( day == 4)

{

myDate = addDays(myDate, 4);

System.out.println("1. Monday (" +df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("2. Tuesday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("3. Wednesday ("+ df.format(myDate)+")");myDate = new Date();

System.out.println("4. Thursday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("5. Friday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("6. Saturday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("7. Sunday ("+ df.format(myDate)+")");

}

else if ( day == 5)

{

myDate = addDays(myDate, 3);

System.out.println("1. Monday (" +df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("2. Tuesday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println(“3. Wednesday("+ df.format(myDate)+")");myDate=addDays(myDate,1);

System.out.println("4. Thursday ("+ df.format(myDate)+")");myDate = new Date();

System.out.println("5. Friday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("6. Saturday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("7. Sunday ("+ df.format(myDate)+")");

}

else if ( day == 6)

{

myDate = addDays(myDate, 2);

System.out.println("1. Monday (" +df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("2. Tuesday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("3. Wednesday ("+ df.format(myDate)+")");myDate=addDays(myDate,1);

System.out.println("4. Thursday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("5. Friday ("+ df.format(myDate)+")");myDate = new Date();

System.out.println("6. Saturday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("7. Sunday ("+ df.format(myDate)+")");

}

else if ( day == 7)

{

myDate = addDays(myDate, 1);

System.out.println("1. Monday (" +df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("2. Tuesday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("3. Wednesday ("+ df.format(myDate)+")");myDate=addDays(myDate,1);

System.out.println("4. Thursday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("5. Friday ("+ df.format(myDate)+")");myDate = addDays(myDate, 1);

System.out.println("6. Saturday ("+ df.format(myDate)+")");myDate = new Date();

System.out.println("7. Sunday ("+ df.format(myDate)+")");

}

Scanner Sc = new Scanner ( System.in );

System.out.println("8. Exit");

String Choice = Sc.nextLine();

System.out.print("\u000C");

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT"))||(Choice.equalsIgnoreCase("8")))

{

System.out.print("\u000C");

Book B = new Book();

B.book();

}

return (Integer.valueOf(Choice));

}

public static Date addDays(Date date, int days)

{

Calendar cal = Calendar.getInstance();

cal.setTime(date);

cal.add(Calendar.DATE, days); //minus number would decrement the days

return cal.getTime();

}

}

**Class: Seat**

import java.io.\*;

import java.util.Scanner;

public class Seat

{

public String [][] SelectSeat ( int Audi,int Day, int time, int code)

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close(); }catch(IOException i) { i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

String ar[][] = {};

return ar; }catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

String ar[][] = {};

return ar; }

Scanner Sc = new Scanner ( System.in );

System.out.println("Please enter the number of seats that you want");

int no = Sc.nextInt();

System.out.print("\u000C");

no = no+1;

Print P = new Print ();

Price Pr = new Price ();

int catchreturn;

catchreturn = P.print( Audi);

System.out.println("Please select your seats (Format:A1");

System.out.println(" A2");

System.out.println(" A3");

System.out.println(" e.t.c)");

String Ch ="";

String Store5 [] = new String [no];

for(int i = 0 ; i < no ; ++i )

{

Ch = Sc.nextLine();

if((Ch.equalsIgnoreCase("Exit"))||(Ch.equalsIgnoreCase("E")))

{

Book B = new Book(); B.book();

} else{ Store5 [i] = Ch; } }

String PriceList [][] = new String [2][no]; PriceList = Pr.price(Store5,Audi,no);

Food F = new Food (); Book B = new Book ();

try{B.Food = F.foodlist();}catch ( Exception E ) { ; }return (PriceList);}

}

**Class: Food**

import java.util.Scanner;

public class Food

{

public String [][] foodlist ( )

{

System.out.print("\u000C");

Scanner Sc = new Scanner ( System.in );

System.out.println("You can now order food which will be delivered at your seat");

System.out.println("1. Order Food");

System.out.println("2. Skip to Payment");

int in = Sc.nextInt();

String Hi [][] = new String [0][0];

if ( in == 1 ) { ; }

else if ( in == 2 ) { return Hi; }

else { Food f = new Food (); f.foodlist ();}System.out.print("\u000C");

String Order[][] = new String [3][25];

System.out.println("You can place a maximum of 25 orders only");

String x;

int x2;

outer:

for (int i = 0 ; i<25;)

{ x2 = Display();

System.out.println();

System.out.println("Order no:"+(i+1));

System.out.println("Please enter the item that you want");

x = Sc.next();

System.out.print("\u000C");

Order [0][i] = Calc (x);

Order [1][i] = Calc2 (x);

System.out.println("Please enter quantity");

Order [2][i] = Sc.next();

System.out.print("\u000C");

System.out.println("Oder Stored");

System.out.println("Enter 1 to exit, 2 to continue ordering");

x2 = Sc.nextInt();

if ( x2 == 1){break outer;}

else{i=i+1;}

System.out.print("\u000C"); }return Order;}

public static int Display()

{

String d[] = new String [25];

d[0]=" SNACK PRICE";

d[1]="A. Popcorn - Small Rs. 90";

d[2]="B. Popcorn - Medium Rs. 140";

d[3]="C. Popcorn - Tub Rs. 170";

d[4]="D. Caramel Popcorn - Small Rs. 150";

d[5]="E. Caramel Popcorn - Regular Rs. 210";

d[6]="F. Chilli Popcorn - Small Rs. 130";

d[7]="G. Chilli Popcorn - Regular Rs. 190";

d[8]="H. Nachos - Small Rs. 170";

d[9]="I. Nachos - Regular Rs. 250";

d[10]="J. Momos - veg ( 10 pc. ) Rs. 250";

d[11]="K. Momos - NonVeg ( 10 pc. ) Rs. 300";

d[12]="L. Chilli corn - Small Rs. 110";

d[13]="M. Chilli corn - Regular Rs. 160";

d[14]="N. Salsa Sauce ( Additional ) Rs. 50";

d[15]="O. Cheesy Dip ( Additional ) Rs. 50";

d[16]=" BEVERAGE";

d[17]="P. Tea Rs. 60";

d[18]="Q. Masala Tea Rs. 100";

d[19]="R. Hot Coffe Rs. 70";

d[20]="S. Cold Coffe Rs. 150";

d[21]="T. Soft Drink ( Coke, ThumbsUp, Pepsi, Fanta, Mirinda, Sprite, ) - Small Rs. 70";

d[22]="U. Soft Drink ( Coke, ThumbsUp, Pepsi, Fanta, Mirinda, Sprite, ) - Regular Rs. 120";

d[23]="V. Milkshake ( Vanilla, Strawberry, Chocolate, Butterscotch) - Regular Rs. 200";

d[24]="W. Milkshake ( Vanilla, Strawberry, Chocolate, Butterscotch) - Large Rs. 270";

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

{System.out.println(d[i]);} return 0;}

public static String Calc(String x)

{

if(x.equalsIgnoreCase("A")){return "Popcorn - Small";}

else if(x.equalsIgnoreCase("B")){return "Popcorn - Medium";}

else if(x.equalsIgnoreCase("C")){return "Popcorn - Tub";}

else if(x.equalsIgnoreCase("D")){return "Caramel Popcorn - Small";}

else if(x.equalsIgnoreCase("E")){return "Caramel Popcorn - Regular";}

else if(x.equalsIgnoreCase("F")){return "Chilli Popcorn - Small";}

else if(x.equalsIgnoreCase("G")){return "Chilli Popcorn - Regular";}

else if(x.equalsIgnoreCase("H")){return "Nachos - Small";}

else if(x.equalsIgnoreCase("I")){return "Nachos - Regular";}

else if(x.equalsIgnoreCase("J")){return "Momos - veg";}

else if(x.equalsIgnoreCase("K")){return "Momos - NonVeg";}

else if(x.equalsIgnoreCase("L")){return "Chilli corn - Small";}

else if(x.equalsIgnoreCase("M")){return "Chilli corn - Regular";}

else if(x.equalsIgnoreCase("N")){return "Salsa Sauce ( Additional )";}

else if(x.equalsIgnoreCase("O")){return "Cheesy Dip ( Additional )";}

else if(x.equalsIgnoreCase("P")){return "Tea";}

else if(x.equalsIgnoreCase("Q")){return "Masala Tea ";}

else if(x.equalsIgnoreCase("R")){return "Hot Coffe";}

else if(x.equalsIgnoreCase("S")){return "Cold Coffe";}

else if(x.equalsIgnoreCase("T")){return "Soft Drink - Small";}

else if(x.equalsIgnoreCase("U")){return "Soft Drink - Regular";}

else if(x.equalsIgnoreCase("V")){return "Milkshake - Regular";}

else if(x.equalsIgnoreCase("W")){return "Milkshake - Large";}

else { return “";} }

public static String Calc2(String x)

{

if(x.equalsIgnoreCase("A")){return “90";} else if(x.equalsIgnoreCase("B")){return "140";}

else if(x.equalsIgnoreCase("C")){return “170";} else if(x.equalsIgnoreCase("D")){return "150";}

else if(x.equalsIgnoreCase("E")){return “210";} else if(x.equalsIgnoreCase("F")){return "130";}

else if(x.equalsIgnoreCase("G")){return “190";} else if(x.equalsIgnoreCase("H")){return "170";}

else if(x.equalsIgnoreCase("I")){return “250";} else if(x.equalsIgnoreCase("J")){return "250";}

else if(x.equalsIgnoreCase("K")){return “300";} else if(x.equalsIgnoreCase("L")){return "110";}

else if(x.equalsIgnoreCase("M")){return "160";} else if(x.equalsIgnoreCase("N")){return "50";}

else if(x.equalsIgnoreCase("O")){return “50";} else if(x.equalsIgnoreCase("P")){return "60";}

else if(x.equalsIgnoreCase("Q")){return “100";} else if(x.equalsIgnoreCase("R")){return "70";}

else if(x.equalsIgnoreCase("S")){return “150";} else if(x.equalsIgnoreCase("T")){return "70";}

else if(x.equalsIgnoreCase("U")){return “120";} else if(x.equalsIgnoreCase("V")){return "200";}

else if(x.equalsIgnoreCase("W")){return “270";} else { return “";} }

}

**Class: Bill**

import java.util.Scanner;

import java.io.\*;

public class Bill

{

public int Bill1 ( String seat [][],String food [][], int code,String date,int time,int audi,int Day)

{

Scanner Sc = new Scanner ( System.in );

Store s = null;

System.out.print("\u000C");

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

System.out.println("Movie: "+s.movieList[code][0]);

System.out.println("Language: "+s.movieList[code][1]);

System.out.println("Duration: "+s.movieList[code][2]+"mins");

System.out.println("Date: "+date);

System.out.println("Time: "+time);

System.out.println("Audi: "+audi);

System.out.print("Seats: ");

int Price = 0;

for(int i = 0 ; i<96 ; ++i )

{

try{if(seat [0][i].equals("")||seat [0][i].equals(null)){;}

else{

System.out.print(seat [0][i]+" ");

Price = Price + Integer.valueOf(seat [1][i]);

}}catch(Exception E){;}

}

System.out.println();

try{

if(food [0][0].equals("")||food [0][0].equals(null))

{;}

else

{

System.out.println("Food:");

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

{

if(food [2][i].equalsIgnoreCase(null) && food [0][i].equalsIgnoreCase(null))

{

;

}

else

{

System.out.print(food [0][i]);

System.out.print(" : X"+food [2][i]);

Price = Price + (Integer.valueOf(food [1][i])\*Integer.valueOf(food [2][i]));

}

System.out.println();

}

}

}catch(Exception E){;}

System.out.println();

System.out.println("Internet Booking Charges : Rs.100");

Price = Price + 100;

System.out.println("Tax (28%) : Rs."+((Price/100)\*28));

Price = Price + ((Price/100)\*28);

System.out.println("Grand Total : Rs."+Price);

System.out.println();

System.out.println("1. Confirm");

System.out.println("2. Cancel");

int Choice = Sc.nextInt();

if ( Choice == 1 ) {

Choice = Payment(Price);

if ( Choice == 0 )

{

return 0;

}

else

{

Choice = Bill2(seat,food,code,date,time,audi,Day);

return 1;

}

}

else

{

return 0;

}

}

public static int Payment ( int Price )

{

outer:

while ( true )

{

System.out.print("\u000C");

Scanner Sc = new Scanner ( System.in ) ;

System.out.println("Please enter Payment option");

System.out.println("1. Credit Card");

System.out.println("2. Debit Card");

int x = Sc.nextInt();

if(x==1){;}else if (x==2){;} else{ continue outer;}

int l =0;

inner:

while(true)

{

System.out.print("\u000C");

System.out.println("Please enter yuor 16 digit account number");

String x2 = Sc.next();

l = x2.length();

if((x2.equalsIgnoreCase("Exit"))||(x2.equalsIgnoreCase("E"))){return 0;}

else if( (l < 16)||(l>16) ){

System.out.println("16 digit account number not valid");

System.out.println("Please try again");

System.out.println("Enter \"EXIT\" to exit");

try{Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

continue inner;}else{;}

inner2:

while(true)

{

System.out.print("\u000C");

System.out.println("Please enter expiry date of card in format MMYY");

x2 = Sc.next();

l = x2.length();

if((x2.equalsIgnoreCase("Exit"))||(x2.equalsIgnoreCase("E"))){return 0;}

else if( (l < 4)||(l>4) ){

System.out.println("Expiry date not valid");

System.out.println("Please try again");

System.out.println("Enter \"EXIT\" to exit");

try{Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

continue inner2;} else{;}

System.out.print("\u000C");

System.out.println("Please enter name on the card");

x2 = Sc.nextLine();

inner3:

while(true)

{

System.out.print("\u000C");

System.out.println("Please enter the three digit cvv number of your card");

x2 = Sc.next();

l = x2.length();

if((x2.equalsIgnoreCase("Exit"))||(x2.equalsIgnoreCase("E"))){return 0;}

else if( (l < 3)||(l>3) ){System.out.println("3 digit cvv number not valid");

System.out.println("Please try again");

System.out.println("Enter \"EXIT\" to exit");

try{Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

continue inner3;} else{;}

System.out.print("\u000C");

System.out.println("1. Confirm Payment of Rs."+Price);

System.out.println("2. Cancel Payment");

l = Sc.nextInt();

if( l == 2 ){return 0;} else {;}

inner4:

while(true)

{

System.out.print("\u000C");

double con = Math.random();

l = (int) (con\*1000000);

System.out.println(l);

System.out.println("Please enter the above numbers for security check");

x = Sc.nextInt();

if(l==x){break outer;}else{

System.out.println("Security code did not match");

System.out.println("Please try again");

try{Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

continue inner4;}

}

}

}

}

}

return 1;

}

public static int Bill2(String seat[][],String food[][],int code,String date,int time,int audi,int Day)

{

/\*\* try {\*/Scanner Sc = new Scanner ( System.in);

System.out.print("\u000C");

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

System.out.println("Please wait a moment while we prepare your tickets");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

int bk = (int) (Math.random()\*100000000);

int i = 0;

String Seeeeet;

outer:

while ( true )

{

try {

i=i+1;

System.out.println(" I Cinemas ");

Seeeeet = seat [0][i];

System.out.println("");

System.out.println("Movie: "+s.movieList[code][0]);

System.out.println("Language: "+s.movieList[code][1]);

System.out.println("Date: "+date);

System.out.println("Time: "+time);

System.out.println("Audi: "+audi);

System.out.println("Seat: "+Seeeeet);

System.out.println("Booking ID: "+bk);

System.out.println("");

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

bk=bk+1;

}

catch ( Exception E ) { break outer ;}

}

System.out.println("");

System.out.println("If you have odered food online, please show your");

System.out.println(" Booking ID at the food counter at the cinemas" );

System.out.println("");

System.out.println("Enter any character to exit");

String h = Sc.nextLine();

System.out.print("\u000C");

System.out.println(" Thank You For Using I Cinemas ");

System.out.println(" We Hope You Have A Great Cinematic Experience");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

System.out.println(" Thank You For Using I Cinemas ");

System.out.println(" We Hope You Have A Great Cinematic Experience");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

}

**Class: Print**

import java.io.\*;

public class Print

{

public int print ( int Audi)

{

int catchreturn;

if(Audi == 1){catchreturn = PrintA1();}

else {catchreturn = PrintA3();}

return 1; }

public int PrintA1 ()

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1; }

for(int i = 0, j = 0, c = 0; c < 238 ; ++c )

{

if(j <= 11 )

{

System.out.print(s.formatAudi1 [i][j]+" " );

if(i==16)

{

System.out.println("|");

i=0;

++j;

}else{

++i;} }

else

{

System.out.print(s.formatAudi1 [i][j] );

if(i==16)

{

System.out.println();

i=0;

++j;

}else{

++i;}} }

System.out.println();

System.out.println(" [] -> Available");

System.out.println(" # -> Booked");

return 1; }

public int PrintA3 ()

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1; }

for(int i = 0, j = 0, c = 0; c < 144 ; ++c )

{

System.out.print(s.formatAudi3 [i][j]);

if(i==11)

{

if(j == 11) {

System.out.println("|");

break;

}

else if ( j > 9 ) {

System.out.println("|");

System.out.println("\t\t\t\t\t\t\t |");

}

else {

System.out.println(" |");

System.out.println("\t\t\t\t\t\t\t |");

}

i=0;

++j; }else{

++i;} }

System.out.println(); System.out.println(" [ ] -> Available”);

System.out.println(" [[#]] -> Booked”); return 1; }

}

**Class: DeCode**

public class DeCode

{

public static String decode (int Audi, String Seat)

{

String A1 [][] = new String [17][14];

String A2 [][] = new String [12][12];

A1 [2][2]="A1";A1 [4][2]="A3";A1 [6][2]="A5";A1 [10][2]="A9";

A1 [3][2]="A2";A1 [5][2]="A4";A1 [7][2]="A6";A1 [11][2]="A10";

A1 [12][2]="A11";A1 [13][2]="A12";A1 [14][2]="A13";A1 [15][2]="A14";

A1 [2][3]="B1";A1 [4][2]="B3";A1 [6][3]="B5";A1 [10][3]="B9";

A1 [3][3]="B2";A1 [5][2]="B4";A1 [7][3]="B6";A1 [11][3]="B10";

A1 [12][3]="B11";A1 [13][2]="B12";A1 [14][3]="B13";A1 [15][3]="B14";

A1 [2][4]="C1"; A1 [4][4]="C3"; A1 [6][4]="C5"; A1 [10][4]="C9";

A1 [3][4]="C2"; A1 [5][4]="C4"; A1 [7][4]="C6"; A1 [11][4]="C10";

A1 [12][4]="C11";A1 [13][4]="C12";A1 [14][4]="C13";A1 [15][4]="C14";

A1 [2][6]="D1"; A1 [3][6]="D2"; A1 [4][6]="D3"; A1 [6][6]="D5";

A1 [7][6]="D6"; A1 [8][6]="D7"; A1 [9][6]="D8"; A1 [10][6]="D9";

A1 [11][6]="D10";A1 [13][6]="D12";A1 [14][6]="D13";A1 [15][6]="D14";

A1 [2][7]="E1"; A1 [3][7]="E2"; A1 [4][7]="E3"; A1 [6][7]="E5";

A1 [7][7]="E6"; A1 [8][7]="E7"; A1 [9][7]="E8"; A1 [10][7]="E9";

A1 [11][7]="E10";A1 [13][7]="E12";A1 [14][7]="E13";A1 [15][7]="E14";

A1 [2][8]="F1"; A1 [3][8]="F2"; A1 [4][8]="F3"; A1 [6][8]="F5";

A1 [7][8]="F6"; A1 [8][8]="F7"; A1 [9][8]="F8"; A1 [10][8]="F9";

A1 [11][8]="F10";A1 [13][8]="F12";A1 [14][8]="F13";A1 [15][8]="F14";

A1 [2][9]="G1"; A1 [3][9]="G2"; A1 [4][9]="G3"; A1 [6][9]="G5";

A1 [7][9]="G6"; A1 [8][9]="G7"; A1 [9][9]="G8"; A1 [10][9]="G9";

A1 [11][9]="G10";A1 [13][9]="G12";A1 [14][9]="G13";A1 [15][9]="G14";

A1 [2][10]="H1"; A1 [3][10]="H2"; A1 [4][10]="H3"; A1 [6][10]="H5";

A1 [7][10]="H6"; A1 [8][10]="H7"; A1 [9][10]="H8"; A1 [10][10]="H9";

A1 [11][10]="H10";A1 [13][10]="H12";A1 [14][10]="H13";A1 [15][10]="H14";

A2 [2][2] = "A1";A2 [4][2] = "A2";A2 [6][2] = "A3";A2 [8][2] = "A4";A2 [10][2] = "A5";

A2 [2][3] = "B1";A2 [4][3] = "B2";A2 [6][3] = "B3";A2 [8][3] = "B4";A2 [10][3] = "B5";

A2 [2][4] = "C1";A2 [4][4] = "C2";A2 [6][4] = "C3";A2 [8][4] = "C4";A2 [10][4] = "C5";

A2 [2][6] = "D1";A2 [4][6] = "D2";A2 [6][6] = "D3";A2 [8][6] = "D4";A2 [10][6] = "D5";

A2 [2][7] = "E1";A2 [4][7] = "E2";A2 [6][7] = "E3";A2 [8][7] = "E4";A2 [10][7] = "E5";

A2 [2][8] = "F1";A2 [4][8] = "F2";A2 [6][8] = "F3";A2 [8][8] = "F4";A2 [10][8] = "F5";

try{ if ( (Audi == 1 )||( Audi == 2 )) {

for ( int i = 0, j = 0,c = 0;c<238;++c)

{ if(A1[i][j].equalsIgnoreCase(Seat)){

return (String)(i+"."+j);

} else{

if(i==16){i=0;++j;}

else{++i;}} } }

else if ( Audi == 3 )

{ for ( int i = 0, j = 0,c = 0;c<144;++c)

{ try{if(A2[i][j].equalsIgnoreCase(Seat)){

return (String)(i+”."+j); }

else{ if(i==11){i=0;++j;}

else{++i;} }}catch(Exception E){;} }

}}catch(Exception E ){;}

return "NO";

} }

**Class: Price**

import java.io.\*;

public class Price

{

public String [][] price ( String in [] , int Audi , int no)

{

DeCode D = new DeCode ();

String out [][] = new String [2][no];

String temp = "";

int index;

int row =0;

for( int i = 1,j=0; i<no ; ++i )

{

if ( ( Audi == 1 ) || ( Audi == 2 )){

temp = D.decode(1,in[i]);}

else if ( Audi == 3){

temp = D.decode(3,in[i]);}

else { ; }

index = temp.indexOf(".");

try{row = Integer.valueOf(temp.substring((index+1),(temp.length())));}catch(Exception E){;}

if ( ( Audi == 1 ) || ( Audi == 2 )){

if((row==2)||(row==3)||(row==4)){

out [0][i] = in[i];

out [1][i] = "240";

}

else{

out [0][i] = in[i];

out [1][i] = "190";

}

}

else if ( Audi == 3 ){

if((row==2)||(row==3)||(row==4)){

out [0][i] = in[i];

out [1][i] = "1200";

}

else{

out [0][i] = in[i];

out [1][i] = "1000";

}

}

}

return out ;

}

}

**Class: Show**

import java.util.Scanner;

import java.io.\*;

public class Show

{

public int NowPlaying ( )

{

Scanner Sc = new Scanner ( System.in );

String Choice;

int temp = -1;

char a = 'A';

String b = a+". ";

int arr [] = new int [50];

String code;

Outer :

while ( true )

{

Store s = null;

try {

FileInputStream fileIn = new FileInputStream("Data.txt");

ObjectInputStream in = new ObjectInputStream(fileIn);

s = (Store) in.readObject();

in.close();

fileIn.close();

}catch(IOException i) {

i.printStackTrace();

System.out.println("Store not found : ERROR 002");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}catch(ClassNotFoundException c) {

System.out.println("Store not found : ERROR 002");

c.printStackTrace();

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

return 1;

}

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

{

arr[i]=-1;

}

for ( int i =0,j=0,c=0,d=0;c<49;++c)

{

if(!s.movieList[i][j].equals(""))

{

if((j==0) && (i>0))

{

System.out.print(b);

++a;

b=a+". ";

arr[d]=i;

++d;

}

else

{

}

System.out.print(s.movieList[i][j]+"\t");

System.out.print("\t|");

if(j==2)

{

++i;

j=0;

System.out.println();

}

else

{

++j;

}

}

else

{

++i;

}

}

System.out.println();

System.out.println("Please enter the movie you want to see the timings for");

System.out.println("Enter \"EXIT\" to exit");

Choice = Sc.nextLine();

System.out.print("\u000C");

if((Choice.equalsIgnoreCase("EXIT")))

{

System.out.print("\u000C");

return 1;

}

else

;

a = 'A';

in :

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

{

b = a+"";

if(Choice.equalsIgnoreCase(b))

{

temp = arr[i];

break in;

}

else

{

++a;

}

}

if(temp == -1)

{

System.out.println("ERROR:200");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

continue Outer;

}

else

{

System.out.print("\u000C");

}

code = s.movieList[temp][3];

Inner :

while ( true )

{

System.out.print(" Mavie: "+s.movieList[temp][0]+" | Language: "+s.movieList[temp][1]);

System.out.print(" | Duration: "+s.movieList[temp][2]+" mins");

System.out.println();

System.out.println("Please enter the day for which you want to see the timings");

System.out.println("1. Monday");

System.out.println("2. Tuesday");

System.out.println("3. Wednesday");

System.out.println("4. Thursday");

System.out.println("5. Friday");

System.out.println("6. Saturday");

System.out.println("7. Sunday");

System.out.println("8. Exit");

Choice = Sc.nextLine();

System.out.print("\u000C");

if((Choice.equalsIgnoreCase("E"))||(Choice.equalsIgnoreCase("EXIT"))||(Choice.equalsIgnoreCase("8")))

{

System.out.print("\u000C");

continue Outer;

}

if(Choice.equalsIgnoreCase("Monday")||Choice.equalsIgnoreCase("1"))

{

for(int i = 0;i<2359;++i)

{ if((s.D1 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D1 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D1 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Inner;

}

else if(Choice.equalsIgnoreCase("Tuesday")||Choice.equalsIgnoreCase("2"))

{

for(int i = 0;i<2359;++i)

{ if((s.D2 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D2 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D2 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Inner;

}

else if(Choice.equalsIgnoreCase("Wednesday")||Choice.equalsIgnoreCase("3"))

{

for(int i = 0;i<2359;++i)

{ if((s.D3 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D3 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D3 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Inner;

}

else if(Choice.equalsIgnoreCase("Thursday")||Choice.equalsIgnoreCase("4"))

{

for(int i = 0;i<2359;++i)

{ if((s.D4 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D4 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D4 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Inner;

}

else if(Choice.equalsIgnoreCase("Friday")||Choice.equalsIgnoreCase("5"))

{

for(int i = 0;i<2359;++i)

{ if((s.D5 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D5 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D5 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Inner;

}

else if(Choice.equalsIgnoreCase("Saturday")||Choice.equalsIgnoreCase("6"))

{

for(int i = 0;i<2359;++i)

{ if((s.D6 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D6 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D6 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Inner;

}

else if(Choice.equalsIgnoreCase("Sunday")||Choice.equalsIgnoreCase("7"))

{

for(int i = 0;i<2359;++i)

{ if((s.D7 [i][1]).startsWith(code+".S"))

System.out.println("-> Audi:1 Time:"+i+" Hrs.");

else

;

if((s.D7 [i][2]).startsWith(code+".S"))

System.out.println("-> Audi:2 Time:"+i+" Hrs.");

else

;

if((s.D7 [i][3]).startsWith(code+".S"))

System.out.println("-> Audi:3 Time:"+i+" Hrs.");

else

;

}

System.out.println("Enter any character to exit");

Choice = Sc.nextLine();

Choice = "ERROR";

System.out.print("\u000C");

continue Inner;

}

else

{ System.out.println("ERROR");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

}

try {

FileOutputStream fileOut =

new FileOutputStream("Data.txt");

ObjectOutputStream out = new ObjectOutputStream(fileOut);

out.writeObject(s);

out.close();

fileOut.close();

System.out.print("\u000C");

System.out.printf("");

}catch(IOException i) {

i.printStackTrace();

System.out.printf("ERROR 000");

try{

Thread.sleep(3000);}

catch ( InterruptedException ex){Thread.currentThread().interrupt();}

System.out.print("\u000C");

continue Outer;

} } } }

}

**Class: Main**

public class Main

{

public static void main ( String [] args )

{

String Choice;

boolean result = false;

Login l = new Login();

SystemUpdate SU = new SystemUpdate();

Show Sh = new Show ();

Book B = new Book ();

Screen S = new Screen ();

int a;

while ( true )

{

Choice = S.Screen\_1();

result = false;

if(Choice.equalsIgnoreCase("A"))

{

SystemUpdate S\_U = new SystemUpdate();

result = l.login();

if(result==true)

a = SU.systemupdate();

else

System.out.print("\u000C");

continue;

}

else if(Choice.equalsIgnoreCase("B"))

{

a = Sh.NowPlaying();

continue;

}

else if(Choice.equalsIgnoreCase("C"))

{

a = B.book();

continue;

}

else if(Choice.equalsIgnoreCase("D")||Choice.equalsIgnoreCase("E")||Choice.equalsIgnoreCase("EXIT"))

{

break;

}

else

{

System.out.println("Please enter a valid choice");

continue;

}

}

System.exit(0);

}

}

List Of Variables

|  |  |  |
| --- | --- | --- |
| **LIST OF VARIABLES ( Implements Serializable)** | | |
| **VARIABLE NAME** | **TYPE** | **PURPOSE** |
| formatAudi1 [][] | String [][] | Used to store the format of Audi 1 and Audi 2 and to print it while booking tickets. |
| formatAudi3 [][] | String [][] | Used to store the format of Audi 3 (gold class) and to print it while booking tickets. |
| D1 [][] | String [][] | Used to store the list of movies plying on Monday across all Audis |
| D2 [][] | String [][] | Used to store the list of movies plying on Tuesday across all Audis |
| D3 [][] | String [][] | Used to store the list of movies plying on Wednesday across all Audis |
| D4 [][] | String [][] | Used to store the list of movies plying on Thursday across all Audis |
| D5 [][] | String [][] | Used to store the list of movies plying on Friday across all Audis |
| D6 [][] | String [][] | Used to store the list of movies plying on Saturday across all Audis |
| D7 [][] | String [][] | Used to store the list of movies plying on Sunday across all Audis |
| movieList [][] | String [][] | Used to store the list of names, languages, durations and codes of all the movies |
| code | int | Each movie has a different code, used for searching in the above given arrays. |

The given list of variables are initialised when the user implements the store class. The following variables are the core of this software and is used throughout the program in each and every class.

These variables are in a class which implements java.io.Serializable and thus are stored in a file named “Data.txt”. These variables have been written in a file so that the changes made by the user ( in System Update ) are saved for the next use.

|  |  |  |
| --- | --- | --- |
| **LIST OF VARIABLES** | | |
| **VARIABLE NAME** | **TYPE** | **PURPOSE** |
| Pricelist [][] | String [][] | This array is used to store all the items ( seats and food ) bought by the user along with its price. This array is further used to compute the sum to be paid by the user and is also used to print the bill containing the seat numbers booked and the food ordered by the user. |
| Price [][] | String [][] | This has a similar function as Pricelist and is used to store the inputs of all the food ordered and seats booked by the user. |
| ID | String | Used to store the correct ID |
| Password | String | Used to store the correct password |
| PasswordIN | String | Takes input for the Password |
| IDIN | String | Takes input for the ID |

These variables do not implement java.io.Serializable and do not get written in the file; however, are essential for the functioning of the program.

A lot of other variables also exist; however, they are local in nature and numerous in number and thus all of them cannot be listed down. Never the less, all the variables in a program are important for the proper functioning of the program.

Output Screenshots

|  |  |  |
| --- | --- | --- |
| **S.No** | **Output Screenshot** | **Explantion** |
| **HOME SCREEN** | | |
| **1** |  | The starting screen asks the user to input his/her choice. Any option other than specified by the menu would produce no change. “Exit” or “E” will have the same effect as option “D”, i.e. to exit the program.  Choosing option “A” or “a” would  take the user to the login domain. The user must know the userID and password to proceed further. |
|  |
| **LOGIN** | | |
|  |  | Now the program asks for the user ID as System can only be updated by an authorised person.  On entering the wrong user ID the program displays an appropriate message and asks for the user ID again. Three incorrect tries freezes the system for 10 seconds for security purposes. Entering “EXIT” or “E” returns to the home page.  Exiting this page causes a Verification Failed message to pop up before returning to the home page.  On entering the correct User ID the program moves on to ask the password. |
|  |
|  |
|  |
|  |
|  |
|  |
|  | Now the program asks for the password for security purposes.  On entering the wrong password the program displays an appropriate message and asks for the password again. Three incorrect tries freezes the system for  10 seconds for security purposes. Entering “EXIT” or “E” returns to the home page.  Exiting this page causes a Verification Failed message to pop up before returning to the home page.  On entering the correct password the program moves on to the System Update page.  Password is case sensitive unlike the user ID. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **SYSTEM UPDATE** | | |
| **1A** |  | The System Update page has three options to add or delete a movie or to update an existing movie, its show timing. “E” or “EXIT” returns to the home page. |
| **HOME >> SYSTEM UPDATE >> Add a Movie** | | |
| **1A**  **A** |  | On entering “A”, the program now asks for the name, language and duration of the movie that needs to be added. Upon storing the details the program displays an appropriate message.  While entering the name of the movie, the program checks if that movie is already playing. If such a case occurs then the program gives the user the freedom of keeping both the movies or to overwrite the existing one (This feature can be used to modify language and duration of an existing movie). |
|  |
|  |
|  |
|  |
|  |
|  |
| **HOME >> SYSTEM UPDATE >> Update movie/show timings** | | |
| **1A**  **B** |  | On entering “B”, the program displays the list of movies and asks for the movie that needs to be updated.  After selecting the movie, the program gives the user the option to (i) edit the movie’s name, language or duration, (ii) to show the movie’s existing timings, (iii) to |
|  | add shows for that movie, and (iv) to delete existing shows of the movie. |
| **1A**  **B**  **A** |  | On entering "A", the user can choose to change the name, language or duration of the selected movie.  After updating any of the parameters of the movie, the program displays an appropriate message before moving to the System Update screen.  In case of an error, the program handles it by displaying an appropriate message before moving to the System Update screen without making any changes. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **1A**  **B**  **B** | A | On entering “B”, the program displays the list of movies and asks for the movie for which the user needs to see the existing show timings.  After selecting the movie the program asks for the day for which the user needs to see the timings.  Now, the program lists down all the shows of that particular movie across all audis for that day of the week.  After seeing the timings the user may  enter any character to exit. Entering “Exit” at any point, returns the user to the System Update screen. |
|  |
|  |
| **1A**  **B**  **C** |  | On entering “C”, the program displays the list of movies and asks for the movie for which the user needs to add shows.  After that the program asks the user the day on which the user needs to add the show.  After that the program asks for the Audi in which the show needs to be added and the timing of the show.  If no other movie is playing at that time in that Audi, then the program displays a message of timing stored. If another movie is playing in that Audi at that particular time then the program displays the message that the time slot is already booked and asks the user to please try again. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **1A**  **B**  **D** | A | On entering “D”, the program displays the list of movies and asks for the movie for which the user needs to delete shows.  After that the program asks the user the day on which the user needs to delete the show.  Now the program displays all the shows of the selected movie on that day and asks for the Audi number and show timing of the show that needs to be deleted.  After deleting the show the program displays an appropriate message.  Entering “Exit” at any point causes a message of ‘show not deleted’ to appear and then return to the System Update screen without making any changes. |
|  |  |
|  |  |
|  |  |
| **HOME >> SYSTEM UPDATE >> Delete a Movie** | | |
| **1A**  **C** |  | On entering "C", the program displays the list of movies from which the user can choose which one to delete. After selecting the movie, the program asks for confirmation before deleting that movie. It should be noted that deleting a movie also deletes all the shows associated with that movie.  Upon successfully deleting the movie, the program displays an appropriate message.  Entering “Exit” at any point causes a movie not deleted message to pop up before going back to the System Update screen. |
|  |
|  |
|
| **HOME >> LIST OF MOVIES WITH SHOW TIMINGS** | | |
| **1B** |  | On entering "B", on the home screen, the program displays the list of movies.  Now the user can choose the movie for which he/she wants to see the timings for.  Now the program asks for the day for which the user wants to see the timings for.  Now the program displays all the shows of that movie on that day across all Audis.  Exiting any page causes the program to precede to the previous page. |
|  |
|  |
| **HOME >> BOOK A TICKET** | | |
| **1C** |  | * On entering "C", the program displays the list of movies. * Here the user can choose the movie for which he/she wants to book the tickets. * Next the program asks for the day the user wants to book the tickets for. [NOTE: Dates are written in front of the days. This insures that the user does not chooses a past date] * Now the program displays all the shows of that movie on that day and asks the user to enter the     Audi no. and timing of  the movie.   * Now the program asks the user, how many seats do they require. * Now the program displays the seats in the Audi. Seats that are booked is represented by “#”. * The user may choose any seat that is not booked. [NOTE: Different Audis have different seating arrangements. All the seating arrangements of all the three Audis have been displayed later in this documentation] * Now the program gives the user the option to order food online. * If the user does not want to order food online, the user has the option to sip to payment. * If the user selects to order food online, then the program displays the list of menu from which the user may choose. * After selecting the item of food from the menu, the program asks for the quantity of that item. * After entering the quantity of that item, the program asks the user if he/she wants to order more or proceed to payment. * If the user chooses to order more, then the whole process of ordering and taking the quantity of the ordered food is repeated until the user chooses the option of proceed to payment. [NOTE: Here, it may be noted that the user can only order upto 25 different items ( there is no limit on the quantity). After exceeding the limit of twenty-five orders the program by itself moves on to the payment screen without the users confirmation. This feature is also for security purposes] * Now the program displays the movie name, language, duration, date, time, Audi, seats chosen, food ordered, internet booking charges, tax (28% of total) and the grand total and asks for confirmation before proceeding to payments. * In payments, the program asks the user whether it wants to pay by credit or debit card. * Now the program asks the user to enter the 16 digit acc number. Acc numbers less or more than 16 digits are rejected. * Likewise the program asks the user the expiry date and the cvv number of the card. Incorrect digits would not be accepted. * Another confirmation is required. [NOTE: Entering “EXIT” anywhere till now would have returned to the home page; however, after this confirmation, the user cannot go back] * Now the program displays a random number on the screen and asks the user to enter the exact same number for security reasons. * Failing to enter the   exact same number  the program displays  an error message and  prints a different  number to be entered.   * After security check, the program asks the user to wait till it prepares the tickets. * Now the program displays the tickets with different booking IDs. * After the user is satisfied with the tickets, he/she may enter any character to exit. * Before returning to the home page the program displays a token of thanks to the user to use ICinemas. |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
| **HOME >> EXIT** | | |
| **This causes the program to terminate. ( All changes are saved for the next use )** | | |

Audi Formats

