```
// ============================================================================
// TheaterSeatingProgram.cpp : Defines the entry point for the console application.
// Author : Learning Team C - University of Phoenix 2015 -
//          Timothy Fletcher, Justin Amescua, Elbio Iseas, Leif Rebuck, Michelle Patino
// ============================================================================
#include "stdafx.h"
#include <iostream>
#include <iomanip>
#include <conio.h>
#include <array>
#include <cstdlib>
#include <stdlib.h>
#include <ctime>
#include <string>
#include <windows.h>
#include <fstream>

#define ESC 27

using namespace std;

// ================================================
//   theater seat availability by row from row 1 to 10
// ================================================
char row1[]={'#','#','#','#','#','#','#','#','#','#'};
char row2[]={'#','#','#','#','#','#','#','#','#','#'};
char row3[]={'#','#','#','#','#','#','#','#','#','#'};
char row4[]={'#','#','#','#','#','#','#','#','#','#'};
char row5[]={'#','#','#','#','#','#','#','#','#','#'};
char row6[]={'#','#','#','#','#','#','#','#','#','#'};
char row7[]={'#','#','#','#','#','#','#','#','#','#'};
char row8[]={'#','#','#','#','#','#','#','#','#','#'};
char row9[]={'#','#','#','#','#','#','#','#','#','#'};
char row0[]={'#','#','#','#','#','#','#','#','#','#'};
// ============================================
//   theater seat availability in the Auditorium
// ============================================
int auditoriumSeats[] = { 10, 10, 10, 10, 10, 10, 10, 10, 10, 10 };
// ============================================================================================
//   Seat prices by row # :   1     2     3     4     5     6     7     8     9     10
// ============================================================================================
double seatPricesByRow[] = { 60.00, 55.00, 50.00, 45.00, 40.00, 35.00, 30.00, 25.00, 20.00, 15.00 };
// ========================================================
//   array to keep the dolar amount of tickets sold by row
// ========================================================
double soldPerRow[] = { 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00 };
// ========================================================
//   boolean soldOut sentinel for when all seats are sold out
// ========================================================
bool soldOut = false;
// ============================================================
//   function to delaying a loop in miliseconds as an argument
// ============================================================
void sleep (unsigned int);
// ============================================================================================================
//   https://msdn.microsoft.com/en-us/library/windows/desktop/ms686025(v=vs.85).aspx
//   result of the online research for the old C language function for positioning the cursor
//   The purspose of this function is to position the cursor on screen
//   so there is no need to use endl or leading blanks
// ============================================================================================================
void gotoxy(int x, int y) // function with the two int parameters x, and y for column, and row respectively
{
          HANDLE hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);
          COORD coord; // it creates an object COORD named coord for the screen coordinates
          coord.X = x; // it assigns the value of the parameter x (column) to the object coord.X
          coord.Y = y; // it assigns the value of the parameter y (row) to the object coord.Y
          SetConsoleCursorPosition(hStdOut, coord); // it sets the cursor position with the values of the object coord
}
// ============================================================================================
//   this function displays a message or character in the x, y coordinates of the console
//   x is the column argument to display the message
//   y is the line argument to display the message
//   message is the argument you want to display at the coordinates x, y
// ============================================================================================
void say (int x, int y, string message)
{
          gotoxy(x,y);
          cout<< message;
}
// ========================================================
//   this function when called it clear the console screen
//   it uses the system OS command CLS
// ========================================================
void clearscreen()
{
          system("CLS");
}
// ================================================================
//   this function makes the border of a box with five parameters :
//   top left screen column coordinate
//   top left screen line coordinate
//   bottom right screen column coordinate
//   bottom right screen line coordinate
//   the parameter symbol can have one of three possible values
//   0 for blank
//   1 for single line
//   2 for double line
// ================================================================================
void drawboxborder(int x1, int y1, int x2, int y2, int symbol)
{
           string topleftcorner,
                          toprightcorner,
```

```
                              bottomleftcorner,
                              bottomrightcorner,
                              topsymbol,
                              bottomsymbol,
                              leftsymbol,
                              rightsymbol;

          if (symbol == 0)
          {
                    topleftcorner     = " ";
                    toprightcorner    = " ";
                    bottomleftcorner  = " ";
                    bottomrightcorner = " ";
                    topsymbol        = " ";
                    bottomsymbol      = " ";
                    leftsymbol       = " ";
                    rightsymbol       = " ";
          }
          else if (symbol == 1 || symbol == 2)
          {
                    topleftcorner = ( symbol == 1 ) ? "Ú": "É";
                    toprightcorner = ( symbol == 1 ) ? "¿": "»";
                    bottomleftcorner = ( symbol == 1 ) ? "À": "È";
                    bottomrightcorner = ( symbol == 1 ) ? "Ù": "¼";
                    topsymbol = ( symbol == 1 ) ? "Ä" : "Í";
                    bottomsymbol = ( symbol == 1 ) ? "Ä" : "Í";
                    leftsymbol = ( symbol == 1 ) ? "³" : "º";
                    rightsymbol = ( symbol == 1 ) ? "³" : "º";
          }

          if (symbol >= 0 && symbol <= 2)
          {
                    int col, line;

                    say(x1,y1,topleftcorner);
                    say(x2,y1,toprightcorner);

                    for (col = x1+1; col <= x2-1; col++)
                              say(col,y1,topsymbol);
                    for (line = y1+1; line <= y2-1; line++)
                    {
                              say(x1,line,leftsymbol);
                              say(x2,line,rightsymbol);
                    }

                    say(x1,y2,bottomleftcorner);
                    say(x2,y2,bottomrightcorner);

                    for (col = x1+1; col <= x2-1; col++)
                              say(col,y2,bottomsymbol);
          }
}
// ========================================================================
//   this function paints a screen box with the symbol passed as an argument
//   x1 is the left column value of the top left corner
//   y1 is the top line value for the top left corner
//   x2 is the right column value of the bottom right corner
//   y2 is the bottom line value of the bottom right corner
//   symbol is the character you pass as argument to fill the box
// ========================================================================
void paintbox(int x1, int y1, int x2, int y2, string symbol)
{
          int col, line;
          for (line = y1; line <= y2; line++)
          {
                    for (col = x1; col <= x2; col++)
                              say(col,line,symbol);
          }

}
// ========================================================
//   module for reading file with available and taken seats
// ========================================================
void readFromFile()
{
          ifstream textFile;                    // it creates a input file named textFile

          textFile.open("theaterseats.txt");       // it opens the file with the physical name "theaterseats.txt"

          if (textFile.fail())                  // if it fails to  open the file executes the block
          {

                    ofstream textFile;                  // it creates an output file named textFile

                    textFile.open("theaterseats.txt");   // it opens the file with the physical name "theaterseats.txt"

        textFile << row1;                           // saves the array row1 to disk
                    textFile << row2;                           // saves the array row2 to disk
                    textFile << row3;                   // saves the array row3 to disk
                    textFile << row4;                           // saves the array row4 to disk
                    textFile << row5;                           // saves the array row5 to disk
                    textFile << row6;                           // saves the array row6 to disk
                    textFile << row7;                           // saves the array row7 to disk
                    textFile << row8;                           // saves the array row8 to disk
                    textFile << row9;                           // saves the array row9 to disk
                    textFile << row0;                           // saves the array row0 to disk

                    textFile.close();                   // it closes the file
          }
          else                                      // if the files exists it executes the next block
          {
```

```
                    char line[100];                     // it creates a char array of 100 elements, named line
                    char aChar = ' ';                   // it creates a char variable named aChar

                    ifstream textFile;                  // it creates an input file name textFile

                    textFile.open("theaterseats.txt");  // it opens a file named "theaterseats.txt"

                    for (int i = 0; i < 100; i++)       // if runs a for loop of 100 cycles
                    {
                              textFile.get(aChar);            // it gets a char from the file into the variable aChar
                        line[i] = aChar;                     // it assigns that char to the char array line
                    }
                    for (int i = 0; i < 10; i++)        // it runs a for loop of 10 cycles
                    {
                        row1[i] = line[i];              // it loads a specific char from the line array into the row arrays
                              row2[i] = line[i+10];           //        it loads a specific char from the line array into the row arrays
                              row3[i] = line[i+20];           //        it loads a specific char from the line array into the row arrays
                              row4[i] = line[i+30];           //        it loads a specific char from the line array into the row arrays
                              row5[i] = line[i+40];           //        it loads a specific char from the line array into the row arrays
                              row6[i] = line[i+50];           //        it loads a specific char from the line array into the row arrays
                              row7[i] = line[i+60];           //        it loads a specific char from the line array into the row arrays
                              row8[i] = line[i+70];           //        it loads a specific char from the line array into the row arrays
                              row9[i] = line[i+80];           //        it loads a specific char from the line array into the row arrays
                              row0[i] = line[i+90];           //        it loads a specific char from the line array into the row arrays
                    }

                    textFile.close();                   // it closes the file textFile
          }

          textFile.close();                       // it closes the file textFile

}
// ================================
//   module to save seats to file
// ================================
void saveToFile()
{
          ofstream textFile;                                                                  // it creates an output file named textFile

          textFile.open("theaterseats.txt");                                  // it opens the file with the physical name "theaterseats.txt"

    textFile << row1;                                                                 // saves the array row1 to disk
          textFile << row2;                                                           // saves the array row2 to disk
          textFile << row3;                                                           // saves the array row3 to disk
          textFile << row4;                                                           // saves the array row4 to disk
          textFile << row5;                                                           // saves the array row5 to disk
          textFile << row6;                                                           // saves the array row6 to disk
          textFile << row7;                                                           // saves the array row7 to disk
          textFile << row8;                                                           // saves the array row8 to disk
          textFile << row9;                                                           // saves the array row9 to disk
          textFile << row0;                                                           // saves the array row0 to disk

          textFile.close();                                                           // it closes the file textFile

          ofstream backupFile;                          // it creates an output file named backupFile

          backupFile.open("backuptheaterseats.txt");    // it opens this file for writing and it gives the name "backuptheaterseats.txt"

    backupFile << row1;                                             // saves the array row1 to disk
          backupFile << row2;                                       // saves the array row2 to disk
          backupFile << row3;                                 // saves the array row3 to disk
          backupFile << row4;                                 // saves the array row4 to disk
          backupFile << row5;                                 // saves the array row5 to disk
          backupFile << row6;                                 // saves the array row6 to disk
          backupFile << row7;                                 // saves the array row7 to disk
          backupFile << row8;                                 // saves the array row8 to disk
          backupFile << row9;                                 // saves the array row9 to disk
          backupFile << row0;                                 // saves the array row0 to disk

          backupFile.close();                           // it closes the file backupFile
}
// ====================================
//   module for confirming an operation
//   it returns a char Y/y/N/n
// ====================================
char confirm()
{
          char resp = ' ';     // it creates a char variable and initializes with blank

          while (resp!='Y'&& resp!='y'&& resp!='N'&& resp!='n') // it will keep in the while loop until resp is Y/y/N/n
          {
                    say(42,22," ");  // it displays a blank at 10,18
                    gotoxy(42,22);   // it moves the cursor to the coordinates specified
                    resp = getch();  // it gets a character from them keyboard
                    cout << resp;    // display that character on screen
          }

          return resp;
}
// ==========================================================
//   module for updating the state of the seat to '*' Taken
// ==========================================================
void updateLocation(int rowNum, int seatNum)
{
          switch (rowNum)                 // starts evaluating rowNum
          {
                    case 1 :                        // when row is 1
                              row1[seatNum-1] = '*';   // it updates the availability of the seat
                              break;                   // it exits the switch statement
                    case 2  :                       // when row is 2
                              row2[seatNum-1] = '*';   // it updates the availability of the seat
                              break;                   // it exits the switch statement
```

```
                    case 3 :                        // when row is 3
                                    row3[seatNum-1] = '*';   // it updates the availability of the seat
                                    break;                   // it exits the switch statement
                    case 4 :                        // when row is 4
                                    row4[seatNum-1] = '*';   // it updates the availability of the seat
                                    break;                   // it exits the switch statement
                    case 5 :                        // when row is 5
                                    row5[seatNum-1] = '*';   // it updates the availability of the seat
                                    break;                   // it exits the switch statement
                    case 6 :                        // when row is 6
                                    row6[seatNum-1] = '*';   // it updates the availability of the seat
                                    break;                   // it exits the switch statement
                    case 7 :                        // when row is 7
                                    row7[seatNum-1] = '*';   // it updates the availability of the seat
                                    break;                   // it exits the switch statement
                    case 8 :                        // when row is 8
                                    row8[seatNum-1] = '*';   // it updates the availability of the seat
                                    break;                   // it exits the switch statement
                    case 9 :                        // when row is 9
                                    row9[seatNum-1] = '*';   // it updates the availability of the seat
                                    break;                   // it exits the switch statement
                    case 10 :                       // when row is 10
                                    row0[seatNum-1] = '*';   // it updates the availability of the seat
                                    break;                   // it exits the switch statement
                    default :                       // exception condition
                                    {
                                say(4,21,"Exception condition found.");      // it displays the exception condition message
                                    say(4,22,"Press any key to continue ... ");  // it displays the message to continue
                                    getch();                                     // it waits for the user to press any key
                                }
          }
}
// ================================================================================
//   this module displays the seats availability of a specific row received as an argument
// ================================================================================
void displaySeats(int rownum)
{
          int row;
          switch (rownum)
          {
                    case 1:
                              row = 3;
                              gotoxy(58,row);
                              cout << row1[0];
                              gotoxy(60,row);
                              cout << row1[1];
                              gotoxy(62,row);
                              cout << row1[2];
                              gotoxy(64,row);
                              cout << row1[3];
                              gotoxy(66,row);
                              cout << row1[4];
                              gotoxy(68,row);
                              cout << row1[5];
                              gotoxy(70,row);
                              cout << row1[6];
                              gotoxy(72,row);
                              cout << row1[7];
                              gotoxy(74,row);
                              cout << row1[8];
                              gotoxy(76,row);
                              cout << row1[9];
                              break;
                    case 2:
                              row = 5;
                              gotoxy(58,row);
                              cout << row2[0];
                              gotoxy(60,row);
                              cout << row2[1];
                              gotoxy(62,row);
                              cout << row2[2];
                              gotoxy(64,row);
                              cout << row2[3];
                              gotoxy(66,row);
                              cout << row2[4];
                              gotoxy(68,row);
                              cout << row2[5];
                              gotoxy(70,row);
                              cout << row2[6];
                              gotoxy(72,row);
                              cout << row2[7];
                              gotoxy(74,row);
                              cout << row2[8];
                              gotoxy(76,row);
                              cout << row2[9];
                              break;
                    case 3:
                              row = 7;
                              gotoxy(58,row);
                              cout << row3[0];
                              gotoxy(60,row);
                              cout << row3[1];
                              gotoxy(62,row);
                              cout << row3[2];
                              gotoxy(64,row);
                              cout << row3[3];
                              gotoxy(66,row);
                              cout << row3[4];
                              gotoxy(68,row);
                              cout << row3[5];
                              gotoxy(70,row);
                              cout << row3[6];
```

```
            gotoxy(72,row);
            cout << row3[7];
            gotoxy(74,row);
            cout << row3[8];
            gotoxy(76,row);
            cout << row3[9];
            break;
case 4:
            row = 9;
            gotoxy(58,row);
            cout << row4[0];
            gotoxy(60,row);
            cout << row4[1];
            gotoxy(62,row);
            cout << row4[2];
            gotoxy(64,row);
            cout << row4[3];
            gotoxy(66,row);
            cout << row4[4];
            gotoxy(68,row);
            cout << row4[5];
            gotoxy(70,row);
            cout << row4[6];
            gotoxy(72,row);
            cout << row4[7];
            gotoxy(74,row);
            cout << row4[8];
            gotoxy(76,row);
            cout << row4[9];
            break;
case 5:
            row = 11;
            gotoxy(58,row);
            cout << row5[0];
            gotoxy(60,row);
            cout << row5[1];
            gotoxy(62,row);
            cout << row5[2];
            gotoxy(64,row);
            cout << row5[3];
            gotoxy(66,row);
            cout << row5[4];
            gotoxy(68,row);
            cout << row5[5];
            gotoxy(70,row);
            cout << row5[6];
            gotoxy(72,row);
            cout << row5[7];
            gotoxy(74,row);
            cout << row5[8];
            gotoxy(76,row);
            cout << row5[9];
            break;
case 6:
            row = 13;
            gotoxy(58,row);
            cout << row6[0];
            gotoxy(60,row);
            cout << row6[1];
            gotoxy(62,row);
            cout << row6[2];
            gotoxy(64,row);
            cout << row6[3];
            gotoxy(66,row);
            cout << row6[4];
            gotoxy(68,row);
            cout << row6[5];
            gotoxy(70,row);
            cout << row6[6];
            gotoxy(72,row);
            cout << row6[7];
            gotoxy(74,row);
            cout << row6[8];
            gotoxy(76,row);
            cout << row6[9];
            break;
case 7:
            row = 15;
            gotoxy(58,row);
            cout << row7[0];
            gotoxy(60,row);
            cout << row7[1];
            gotoxy(62,row);
            cout << row7[2];
            gotoxy(64,row);
            cout << row7[3];
            gotoxy(66,row);
            cout << row7[4];
            gotoxy(68,row);
            cout << row7[5];
            gotoxy(70,row);
            cout << row7[6];
            gotoxy(72,row);
            cout << row7[7];
            gotoxy(74,row);
            cout << row7[8];
            gotoxy(76,row);
            cout << row7[9];
            break;
case 8:
            row = 17;
            gotoxy(58,row);
```

```
                                cout << row8[0];
                                gotoxy(60,row);
                                cout << row8[1];
                                gotoxy(62,row);
                                cout << row8[2];
                                gotoxy(64,row);
                                cout << row8[3];
                                gotoxy(66,row);
                                cout << row8[4];
                                gotoxy(68,row);
                                cout << row8[5];
                                gotoxy(70,row);
                                cout << row8[6];
                                gotoxy(72,row);
                                cout << row8[7];
                                gotoxy(74,row);
                                cout << row8[8];
                                gotoxy(76,row);
                                cout << row8[9];
                                break;
                        case 9:
                                row = 19;
                                gotoxy(58,row);
                                cout << row9[0];
                                gotoxy(60,row);
                                cout << row9[1];
                                gotoxy(62,row);
                                cout << row9[2];
                                gotoxy(64,row);
                                cout << row9[3];
                                gotoxy(66,row);
                                cout << row9[4];
                                gotoxy(68,row);
                                cout << row9[5];
                                gotoxy(70,row);
                                cout << row9[6];
                                gotoxy(72,row);
                                cout << row9[7];
                                gotoxy(74,row);
                                cout << row9[8];
                                gotoxy(76,row);
                                cout << row9[9];
                                break;
                        case 10:
                                row = 21;
                                gotoxy(58,row);
                                cout << row0[0];
                                gotoxy(60,row);
                                cout << row0[1];
                                gotoxy(62,row);
                                cout << row0[2];
                                gotoxy(64,row);
                                cout << row0[3];
                                gotoxy(66,row);
                                cout << row0[4];
                                gotoxy(68,row);
                                cout << row0[5];
                                gotoxy(70,row);
                                cout << row0[6];
                                gotoxy(72,row);
                                cout << row0[7];
                                gotoxy(74,row);
                                cout << row0[8];
                                gotoxy(76,row);
                                cout << row0[9];
                                break;
                        default :
                                {
                                        say(3,21,"Invalid row number. Press any key ... ");
                                        getch();
                                }
                }

}
// ============================================
//   module for updating the state of the seat
// ============================================
void refreshChart(int rowNumber)
{
        displaySeats(rowNumber);
}
// ============================================
//   module for updating the state of the seat
// ============================================
bool soldOutSeats()
{
        bool noMoreSeats = true;  // sets the bool variable to true

    for (int i = 0; i < 10; i++) // initiates the for loop to check for available seats in the theater
        {
                // evaluates the ten rows for available seats
                if (row1[i]=='#'||row2[i]=='#'||row3[i]=='#'||row4[i]=='#'||row5[i]||row6[i]=='#'||row7[i]=='#'||row8[i]=='#'||row9[i]=='#'||row0[i]=='#')
                {
                        noMoreSeats = false; // sets the bool variable to false
                        break;            // it exits the for loop because there is at least one seat available
                }
        }

        return noMoreSeats; // it returns a false or true value depending on if the theater seats are sold out or not
}
// ========================================
//   module for checking seat availability
```

```
// ========================================
bool available(int row, int seatnum)
{
          bool retvalue = false;
          int seat = 1;
          seat = seatnum - 1;

          switch (row)
          {
                    case 1 :
                              retvalue = (row1[seat] == '#') ? true : false;
                              break;
                    case 2 :
                              retvalue = (row2[seat] == '#') ? true : false;
                              break;
                    case 3 :
                              retvalue = (row3[seat] == '#') ? true : false;
                              break;
                    case 4 :
                              retvalue = (row4[seat] == '#') ? true : false;
                              break;
                    case 5 :
                              retvalue = (row5[seat] == '#') ? true : false;
                              break;
                    case 6 :
                              retvalue = (row6[seat] == '#') ? true : false;
                              break;
                    case 7 :
                              retvalue = (row7[seat] == '#') ? true : false;
                              break;
                    case 8:
                              retvalue = (row8[seat] == '#') ? true : false;
                              break;
                    case 9:
                              retvalue = (row9[seat] == '#') ? true : false;
                              break;
                    case 10 :
                              retvalue = (row0[seat] == '#') ? true : false;
                              break;
          }

          return retvalue;
}
// ============================
//    int chooseTask() module
// ============================
int chooseTask()
{
          char aKey = 54;      // it assigns a value out of range to the var aKey to push it in the while loop


          // it will stay in the loop as long as the key pressed is different from 0-5 or ESC

          while (aKey!=48 && aKey!=ESC && aKey!=49 && aKey!=50 && aKey!=51 && aKey!=52 && aKey!=53)
          {
                    gotoxy(39,22);   // it moves the cursor to the reading position
              aKey = getch();  // it waits for a key to be pressed and it assigns it to the variable aKey
          }
          if (aKey==ESC)       // if the key pressed was ESC then it enters the block
          {
                    aKey=48;         // it assigns the integer value of '0' to the variable aKey
          }
          return aKey;         // it returns the select key of the main menu
}
// ================================================
//   this functions accepts the user's input in a
//   specific range for selecting a seat number
// ================================================
int readSeatToSell()
{
          int seatNumber = -1; // initializes the var seatNumber in -1

          paintbox(3,21,46,22," "); // it clears any message in the message area
          say( 8,22,"Enter a seat number from 1 to 10"); // it displays the message to enter the seat number

          while (seatNumber < 1 || seatNumber > 10) // it stays in the loop while seatNumber is not in the range 0 - 10
          {
                    say(37,12,"     ");  // it clears the old rowNumber if it was out of range
                    gotoxy(37,12);       // it positions the cursor ready for input
              cin >> seatNumber;   // it prompts the user to enter the rowNumber
          }

          return seatNumber; // it returns the seatNumber
}
// ================================================
//   this functions accepts the user's input in a
//   specific range for selecting a row number
// ================================================
int readRow()
{
          int rowNumber = -1; // initializes the var rowNumber in -1

          paintbox(3,21,46,22," "); // it clears any message in the message area
          say( 6,22,"Enter a Row number [1-10], or 0 to exit"); // it displays a message asking the user to enter a row number

          while (rowNumber < 0 || rowNumber > 10) // it stays in the loop while rowNumber is not in the range 0 - 10
          {
                    say(37,10,"     ");  // it clears the old rowNumber if it was out of range.
                    gotoxy(37,10);    // it positions the cursor ready for input
              cin >> rowNumber; // it prompts the user to enter the rowNumber
          }
```

```
          return rowNumber; // it returns the rowNumber
}
// ==========================
//      Tickets module
// ==========================
void tickets()
{
          int row = 1;                // it creates an int variable row and it initializes in 1
          int seatToSell = 1;         // it creates an int variable setToSell and initializes in 1
          bool seatAvailable = false;  // it creates a boolean variable seatAvailable and it initializes in false
          char answer = ' ';          // it creates a char variable answer and initializes in blank
          double price = 0.00;        // it creates a double variable price and initializes in 0.00
    say(3,1,"                                          ");              // it erases last screen title
          say(8,1,    "T I C K E T S   F O R   S A L E");                    // it displays the title of the task to do
          paintbox( 2, 3, 46, 19," ");                                      // it clears just the place of the menu items
          paintbox(3,21,46,22," ");                                         // it clears the messaging box
          drawboxborder( 2,5,47,19,1);                                      // it draws a box border with single line
          if (!soldOut)                                                     // it the theater seats are not sold out
          {
                    say(12,10,"Choose a Row  [1...10] : ");                         // it displays a message for choosing a row
                    say(12,12,"Choose a Seat [1...10] : ");                         // it displays a message for choosing a seat
                    say( 6,22,"Enter a Row number [1-10], or 0 to exit");           // it displays a message asking the user to enter a row number

                    while (row != 0 && !soldOut)                                    // it will stay in the loop until no more seats available or finish
selling seats
                    {
                              say(37,10,"     ");                                         // it clears the last row number
                              say(37,12,"     ");                                         // it clears the last seat number
                              gotoxy(37,10);                                              // it moves the cursor to the place where the user enters the row
number
                              seatToSell = 0;                                             // it initializes seatToSell in zero
                        row = 0;                                                  // it initializes the variable row in zero
                              row = readRow();                                            // it calls the module to read a row and it assigns it to the
variable row
                              if (row != 0)
                              {
                                        say(4,16,"                                    ");       // it clears the message saying the price of the seat
location in $
                                        say( 6,22,"                                      ");    // it clears the message asking the user to enter a row
number
                                        say( 5,22,"                                    ");       // it clears the message for confirming the sale of the seat
                                        say( 4,22,"Enter a Seat number from 1 to 10");           // it displays a message to enter a seat number
                                        gotoxy(37,12);                                           // it moves the cursor to the place to read the seat number
                                        seatToSell = 0;                                          // it initializes seatToSell in zero
                                        seatToSell = readSeatToSell();                           // it calls the module readSeatToSell and it assigns the seat
number to seatToSell
                                        seatAvailable = available(row, seatToSell);              // it checks for seat availability and assigns true or false
depending on availability
                                        if (seatAvailable)                                       // if the seat is available it enters the block
                                        {
                                                  price = seatPricesByRow[row-1];                      // it assigns the price of the seat to the variable
price
                                                  say(4,16,"The price for this seat location is $ ");  // it displays the message saying the price of the
seat location in $
                                                  cout << setw(2) << price;                            // it displays the price of the seat in a three digits
format
                                                  paintbox(3,21,46,22," ");                            // it clears the messaging box
                                                  say( 5,22,"Confirm the sale of this seat? (Y/N) ");  // it displays a message for confirming the sale of
the seat
                                                  gotoxy(42,22);                                       // it moves the cursor to the confirming position on
screen
                                                  answer = confirm();                                  // it calls the module confirm() to get an response
Y/y/N/n and assigns it to answer
                                                  if (answer == 'Y' || answer == 'y')                  // if the answer was 'Y' or 'y' then it enters the
block
                                                  {
                                                            updateLocation(row, seatToSell);                  // it updates the location of the seat to taken
'*'
                                                            refreshChart(row);                                // it refreshes the row of the seat sold
                                                            soldOut = soldOutSeats();                         // it calls the module soldOutSeats() and it
assigns the result (true/false) to the variable soldOut
                                                  }
                                                  say( 5,22,"                                  ");    // it erases a message for confirming the sale of the
seat
                                                  say(4,16,"                                     ");  // it erases the message for selling the seat
                                        }
                                        else                                                     // if not, display a message saying Seat sold
                                        {
                                                  paintbox(3,21,46,22," ");                             // it clears the message box
                                                  say( 4,22,"Seat sold. Press any key to continue... "); // it displays the inavailability message
                                                  getch();                                              // it waits for the user to press any key to continue
                                                  say( 4,22,"                                        "); // it clears the inavailability message
                                        }
                                        say( 6,22,"Enter a Row number [1-10], or 0 to exit");      // it displays a message asking the user to enter a row
number
                              }
                              if (soldOut)                                                    // if the reason for exiting was there were no more seats available
                              {
                                        paintbox(3,21,46,22," ");                                // it clears the messages box
                                        say(4,21,"All the seats in the theater are sold out.");  // it displays the message for sold out seats
                                        say(4,22,"Press any key to return to Main Menu ... ");   // it displays a message to press any key to return to Main
Menu
                                        getch();                                                 // it waits for the user to press any key
                              }
                    }
          }
          else                                                              // the theater seats are sold out
          {
                    say(4,21,"All the seats in the theater are sold out.");           // it dispalys a message for this reason
                    say(4,22,"Press any key to return to Main Menu ... ");            // it displays a message to press any key to return to Main Menu
                    getch();                                                          // it waits for the any key to be pressed
          }
```

```
}
// ================================================
//   module for calculating the total ticket sales
// ================================================
void totalTicketSales()
{
          double totalSold = 0.00;
          int sold[] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

          for (int i=0; i < 10; i++)
          {
                    if ( row1[i] == '*' )
                    {
                              sold[0]++;
                    }
                    if ( row2[i] == '*' )
                    {
                              sold[1]++;
                    }
                    if ( row3[i] == '*' )
                    {
                              sold[2]++;
                    }
                    if ( row4[i] == '*' )
                    {
                              sold[3]++;
                    }
                    if ( row5[i] == '*' )
                    {
                              sold[4]++;
                    }
                    if ( row6[i] == '*' )
                    {
                              sold[5]++;
                    }
                    if ( row7[i] == '*' )
                    {
                              sold[6]++;
                    }
                    if ( row8[i] == '*' )
                    {
                              sold[7]++;
                    }
                    if ( row9[i] == '*' )
                    {
                              sold[8]++;
                    }
                    if ( row0[i] == '*' )
                    {
                              sold[9]++;
                    }
          }

          soldPerRow[0] = seatPricesByRow[0] * sold[0];
          soldPerRow[1] = seatPricesByRow[1] * sold[1];
          soldPerRow[2] = seatPricesByRow[2] * sold[2];
          soldPerRow[3] = seatPricesByRow[3] * sold[3];
          soldPerRow[4] = seatPricesByRow[4] * sold[4];
          soldPerRow[5] = seatPricesByRow[5] * sold[5];
          soldPerRow[6] = seatPricesByRow[6] * sold[6];
          soldPerRow[7] = seatPricesByRow[7] * sold[7];
          soldPerRow[8] = seatPricesByRow[8] * sold[8];
          soldPerRow[9] = seatPricesByRow[9] * sold[9];

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

                    soldPerRow[i] = totalSold + soldPerRow[i];

    gotoxy(40, 6);
          cout << setw(5) << soldPerRow[0];
          gotoxy(40, 7);
          cout << setw(5) << soldPerRow[1];
          gotoxy(40, 8);
          cout << setw(5) << soldPerRow[2];
          gotoxy(40, 9);
          cout << setw(5) << soldPerRow[3];
          gotoxy(40,10);
          cout << setw(5) << soldPerRow[4];
          gotoxy(40,11);
          cout << setw(5) << soldPerRow[5];
          gotoxy(40,12);
          cout << setw(5) << soldPerRow[6];
          gotoxy(40,13);
          cout << setw(5) << soldPerRow[7];
          gotoxy(40,14);
          cout << setw(5) << soldPerRow[8];
          gotoxy(40,15);
          cout << setw(5) << soldPerRow[9];
          for (int i = 0; i < 10; i++)
                    totalSold = totalSold + soldPerRow[i];
          gotoxy(40,17);
          cout << setw(5) << totalSold;
}
// ============================
//   total Tickets sale module
// ============================
void totalTickets()
{
    say(3,1,"                                          ");
          say(7,1,"T O T A L   T I C K E T S   S A L E");
          paintbox( 2, 3, 47, 19," ");
          paintbox(3,21,46,22," ");
```

```
          say( 5, 6, "Tickets at $ 60 each in Row  1 : $");
          say( 5, 7, "Tickets at $ 55 each in Row  2 : $");
          say( 5, 8, "Tickets at $ 50 each in Row  3 : $");
          say( 5, 9, "Tickets at $ 45 each in Row  4 : $");
          say( 5,10, "Tickets at $ 40 each in Row  5 : $");
          say( 5,11, "Tickets at $ 35 each in Row  6 : $");
          say( 5,12, "Tickets at $ 30 each in Row  7 : $");
          say( 5,13, "Tickets at $ 25 each in Row  8 : $");
          say( 5,14, "Tickets at $ 20 each in Row  9 : $");
          say( 5,15, "Tickets at $ 15 each in Row 10 : $");
          say( 5,17, "Total amount for sold tickets  : $");
          drawboxborder(39,16,45,18,1);
          totalTicketSales();
          say(4,22,"Press any key to return to Main Menu ... ");
          getch();
}
// ======================
//   Sold Tickets module
// ======================
void soldTickets()
{
    say(3,1,"                                          ");
          say(14,1,"S O L D   T I C K E T S");
          paintbox( 2, 3, 47, 19," ");
          paintbox(3,21,46,22," ");
          int col = 21;

          say(10, 7, "Row  1 : ");
          say(10, 8, "Row  2 : ");
          say(10, 9, "Row  3 : ");
          say(10,10, "Row  4 : ");
          say(10,11, "Row  5 : ");
          say(10,12, "Row  6 : ");
          say(10,13, "Row  7 : ");
          say(10,14, "Row  8 : ");
          say(10,15, "Row  9 : ");
          say(10,16, "Row 10 : ");

          for (int i = 0; i < 10; i++)
          {
                    gotoxy(col, 7);
                    if (row1[i]=='*')
                       cout << i + 1;
                    gotoxy(col, 8);
                    if (row2[i]=='*')
                       cout << i + 1;
                    gotoxy(col, 9);
                    if (row3[i]=='*')
                       cout << i + 1;
                    gotoxy(col,10);
                    if (row4[i]=='*')
                       cout << i + 1;
                    gotoxy(col,11);
                    if (row5[i]=='*')
                       cout << i + 1;
                    gotoxy(col,12);
                    if (row6[i]=='*')
                       cout << i + 1;
                    gotoxy(col,13);
                    if (row7[i]=='*')
                       cout << i + 1;
                    gotoxy(col,14);
                    if (row8[i]=='*')
                       cout << i + 1;
                    gotoxy(col,15);
                    if (row9[i]=='*')
                       cout << i + 1;
                    gotoxy(col,16);
                    if (row0[i]=='*')
                       cout << i + 1;
                    col = col + 2;
          }
          say(4,22,"Press any key to return to Main Menu ... ");
          getch();
}
// ================================
//   Available Seats Per Row module
// ================================
void seatsAvailabilityperRow()
{
    say(3,1,"                                          ");
          say(3,1,"A V A I L A B L E  S E A T S  P E R  R O W");
          paintbox( 2, 3, 47, 19," ");
          paintbox(3,21,46,22," ");
          int col = 21;
          if (!soldOut)
          {
                    say(10, 7, "Row  1 : ");
                    say(10, 8, "Row  2 : ");
                    say(10, 9, "Row  3 : ");
                    say(10,10, "Row  4 : ");
                    say(10,11, "Row  5 : ");
                    say(10,12, "Row  6 : ");
                    say(10,13, "Row  7 : ");
                    say(10,14, "Row  8 : ");
                    say(10,15, "Row  9 : ");
                    say(10,16, "Row 10 : ");

                    for (int i = 0; i < 10; i++)
                    {
                              gotoxy(col, 7);
                              if (row1[i]=='#')
```

```
                                    cout << i + 1;
                                gotoxy(col, 8);
                                if (row2[i]=='#')
                                    cout << i + 1;
                                gotoxy(col, 9);
                                if (row3[i]=='#')
                                    cout << i + 1;
                                gotoxy(col,10);
                                if (row4[i]=='#')
                                    cout << i + 1;
                                gotoxy(col,11);
                                if (row5[i]=='#')
                                    cout << i + 1;
                                gotoxy(col,12);
                                if (row6[i]=='#')
                                    cout << i + 1;
                                gotoxy(col,13);
                                if (row7[i]=='#')
                                    cout << i + 1;
                                gotoxy(col,14);
                                if (row8[i]=='#')
                                    cout << i + 1;
                                gotoxy(col,15);
                                if (row9[i]=='#')
                                    cout << i + 1;
                                gotoxy(col,16);
                                if (row0[i]=='#')
                                    cout << i + 1;
                                col = col + 2;
                        }

                        say(4,22,"Press any key to return to Main Menu ... ");
                        getch();
                }
                else
                {
                        say(4,21,"All the seats in the theater are sold out.");
                        say(4,22,"Press any key to return to Main Menu ... ");
                        getch();
                }
}
// ================================================================
//   module for calculating the number of available seats per row
// ================================================================
int availableSeatsPerRow(int rownum)
{
        int totalAvailableSeats = 0;

        switch (rownum)
        {
                case 1:
                        for (int i = 0; i < 10; i++)
                        {
                    if (row1[i]=='#')
                                totalAvailableSeats++;
                }
                        break;
                    case 2:
                        for (int i = 0; i < 10; i++)
                        {
                                if (row2[i]=='#')
                                        totalAvailableSeats++;
                        }
                break;
                    case 3:
                        for (int i = 0; i < 10; i++)
                        {
                                if (row3[i]=='#')
                                        totalAvailableSeats++;
                        }
                break;
                    case 4:
                        for (int i = 0; i < 10; i++)
                        {
                                if (row4[i]=='#')
                                        totalAvailableSeats++;
                        }
                break;
                    case 5:
                        for (int i = 0; i < 10; i++)
                        {
                                if (row5[i]=='#')
                                        totalAvailableSeats++;
                        }
                break;
                    case 6:
                        for (int i = 0; i < 10; i++)
                        {
                                if (row6[i]=='#')
                                        totalAvailableSeats++;
                        }
                break;
                    case 7:
                        for (int i = 0; i < 10; i++)
                        {
                                if (row7[i]=='#')
                                        totalAvailableSeats++;
                        }
                break;
                    case 8:
                        for (int i = 0; i < 10; i++)
                        {
```

```
                                        if (row8[i]=='#')
                                                totalAvailableSeats++;
                                }
                        break;
                           case 9:
                                for (int i = 0; i < 10; i++)
                                {
                                        if (row9[i]=='#')
                                                totalAvailableSeats++;
                                }
                        break;
                           case 10:
                                for (int i = 0; i < 10; i++)
                                {
                                        if (row0[i]=='#')
                                                totalAvailableSeats++;
                                }
                        break;
                           default :
                                {
                            totalAvailableSeats=-1;
                                }
                }


  return totalAvailableSeats++;
}
// ====================================
//   Available Seats in the Auditorium
// ====================================
void seatsAvailinAuditorium()
{
    int totalInAuditorium = 0;
    say(3,1,"                                          ");
           say(3,1,"A V A I L A B L E  I N  A U D I T O R I U M");
           paintbox( 2, 3, 47, 19," ");
           paintbox(3,21,46,22," ");
           if (!soldOut)
           {
                    say( 9, 6, "Available seats in Row  1 : ");
                    say( 9, 7, "Available seats in Row  2 : ");
                    say( 9, 8, "Available seats in Row  3 : ");
                    say( 9, 9, "Available seats in Row  4 : ");
                    say( 9,10, "Available seats in Row  5 : ");
                    say( 9,11, "Available seats in Row  6 : ");
                    say( 9,12, "Available seats in Row  7 : ");
                    say( 9,13, "Available seats in Row  8 : ");
                    say( 9,14, "Available seats in Row  9 : ");
                    say( 9,15, "Available seats in Row 10 : ");
                    say( 9,17, "Available in Auditorium   : ");
                    drawboxborder(36,16,40,18,1);

                    auditoriumSeats[0]=availableSeatsPerRow(1);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[0];
                    gotoxy(37,6);
                    cout<< setw(3) << auditoriumSeats[0];

                    auditoriumSeats[1]=availableSeatsPerRow(2);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[1];
                    gotoxy(37,7);
                    cout<< setw(3) << auditoriumSeats[1];

                    auditoriumSeats[2]=availableSeatsPerRow(3);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[2];
                    gotoxy(37,8);
                    cout<< setw(3) << auditoriumSeats[2];

                    auditoriumSeats[3]=availableSeatsPerRow(4);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[3];
                    gotoxy(37,9);
                    cout<< setw(3) << auditoriumSeats[3];

                    auditoriumSeats[4]=availableSeatsPerRow(5);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[4];
                    gotoxy(37,10);
                    cout<< setw(3) << auditoriumSeats[4];

                    auditoriumSeats[5]=availableSeatsPerRow(6);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[5];
                    gotoxy(37,11);
                    cout<< setw(3) << auditoriumSeats[5];

                    auditoriumSeats[6]=availableSeatsPerRow(7);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[6];
                    gotoxy(37,12);
                    cout<< setw(3) << auditoriumSeats[6];

                    auditoriumSeats[7]=availableSeatsPerRow(8);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[7];
                    gotoxy(37,13);
                    cout<< setw(3) << auditoriumSeats[7];

                    auditoriumSeats[8]=availableSeatsPerRow(9);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[8];
                    gotoxy(37,14);
                    cout<< setw(3) << auditoriumSeats[8];

                    auditoriumSeats[9]=availableSeatsPerRow(10);
                    totalInAuditorium = totalInAuditorium + auditoriumSeats[9];
                    gotoxy(37,15);
                    cout<< setw(3) << auditoriumSeats[9];
```

```
                    gotoxy(37,17);
                    cout<< setw(3) << totalInAuditorium;

                    say(4,22,"Press any key to return to Main Menu ... ");
                    getch();
          }
          else
          {
                    say(4,21,"All the seats in the theater are sold out.");
                    say(4,22,"Press any key to return to Main Menu ... ");
                    getch();
          }
}
// ====================================================
//   module for displaying the items of the mani menu
// ====================================================
void displayMenu()
{
          say(3,1,"                                                ");
          say(16,1,"M A I N    M E N U");
          paintbox( 2, 3, 47, 19," ");
          paintbox(3,21,46,22," ");
          say(9, 6,"1 - Selling tickets");
          say(9, 8,"2 - Total of all ticket sales");
          say(9,10,"3 - List of sold tickets");
          say(9,12,"4 - Available Seats per Row");
          say(9,14,"5 - Available Seats in Auditorium");
          say(9,16,"0 - Exit the program [ Esc ]");
          say(13,22,"Select a Task to Perform. ");
          gotoxy(39,22);
}
// ======================================================
//   module for displaying the intro screen to the program
// ======================================================
void intro()
{
          int col1, col2, i, j, k, line;
          col1 = 39;
          col2 = 40;
          i = 0;
          j = 39;
          k = 40;
          line = 0;
          clearscreen();
    paintbox(0,0,79,23,"±");
          for ( i = 0; i < 39; i++ )
          {
              say(j,line," ");
                    say(k,line," ");
                    say(j,line+1," ");
                    say(k,line+1," ");
        say(j,line+2," ");
                    say(k,line+2," ");
                    say(j,line+3," ");
                    say(k,line+3," ");
                    say(j,line+4," ");
                    say(k,line+4," ");
                    say(j,line+5," ");
                    say(k,line+5," ");
              say(j,line+6," ");
                    say(k,line+6," ");
                    say(j,line+7," ");
                    say(k,line+7," ");
        say(j,line+8," ");
                    say(k,line+8," ");
                    say(j,line+9," ");
                    say(k,line+9," ");
                    say(j,line+10," ");
                    say(k,line+10," ");
                    say(j,line+11," ");
                    say(k,line+11," ");
              say(j,line+12," ");
                    say(k,line+12," ");
                    say(j,line+13," ");
                    say(k,line+13," ");
        say(j,line+14," ");
                    say(k,line+14," ");
                    say(j,line+15," ");
                    say(k,line+15," ");
                    say(j,line+16," ");
                    say(k,line+16," ");
                    say(j,line+17," ");
                    say(k,line+17," ");
              say(j,line+18," ");
                    say(k,line+18," ");
                    say(j,line+19," ");
                    say(k,line+19," ");
        say(j,line+20," ");
                    say(k,line+20," ");
                    say(j,line+21," ");
                    say(k,line+21," ");
                    say(j,line+22," ");
                    say(k,line+22," ");
                    say(j,line+23," ");
                    say(k,line+23," ");

                    sleep(8000);

                    j = col1 - i;
                    k = col2 + i;
          }
          drawboxborder( 2,3,77,11,1);
```

```
            say( 18, 5, "- WELCOME TO THE THEATER SEATING PROGRAM -");
            say( 16, 7, "Learning Team C - University of Phoenix 2015");
            say(  5, 9, "Timothy Fletcher-Justin Amescua-Elbio Iseas-Leif Rebuck-Michelle Patino");
            drawboxborder( 2,21,77,23,1);
            say( 19,22, "Press any key to start the program ... ");
            getch();
            paintbox( 2,0,77,23," ");

}
// ========================================
//   module for delaying a loop in seconds
// ========================================
void sleep ( unsigned int secs )
{

}
// ================================================================
//   module to draw the availability chart and title for the screen
// ================================================================
void drawChart()
{
            drawboxborder( 2, 0,47, 2,1);
            say( 8, 1,"THEATER SEATING AVAILABILITY CHART");
            drawboxborder(48, 0,57,22,1);
            drawboxborder(57, 0,59,22,1);
    drawboxborder(59, 0,61,22,1);
            drawboxborder(61, 0,63,22,1);
            drawboxborder(63, 0,65,22,1);
            drawboxborder(65, 0,67,22,1);
            drawboxborder(67, 0,69,22,1);
            drawboxborder(69, 0,71,22,1);
            drawboxborder(71, 0,73,22,1);
            drawboxborder(73, 0,75,22,1);
            drawboxborder(75, 0,77,22,1);
    drawboxborder( 2,20,47,23,1);
            drawboxborder(48, 0,77, 2,1);
            drawboxborder(48, 2,77, 4,1);
    drawboxborder(48, 4,77, 6,1);
            drawboxborder(48, 6,77, 8,1);
            drawboxborder(48, 8,77,10,1);
            drawboxborder(48,10,77,12,1);
            drawboxborder(48,12,77,14,1);
            drawboxborder(48,14,77,16,1);
            drawboxborder(48,16,77,18,1);
            drawboxborder(48,18,77,20,1);
            drawboxborder(48,20,77,22,1);
            say(50, 1, "SEATS");
            say(58, 1, "1");
            say(60, 1, "2");
            say(62, 1, "3");
            say(64, 1, "4");
            say(66, 1, "5");
            say(68, 1, "6");
            say(70, 1, "7");
            say(72, 1, "8");
            say(74, 1, "9");
            say(76, 1, "0");
            say(50, 3, "ROW  1");
            say(50, 5, "ROW  2");
            say(50, 7, "ROW  3");
            say(50, 9, "ROW  4");
            say(50,11, "ROW  5");
            say(50,13, "ROW  6");
            say(50,15, "ROW  7");
            say(50,17, "ROW  8");
            say(50,19, "ROW  9");
            say(50,21, "ROW 10");
            displaySeats(1);
            displaySeats(2);
            displaySeats(3);
            displaySeats(4);
            displaySeats(5);
            displaySeats(6);
            displaySeats(7);
            displaySeats(8);
            displaySeats(9);
            displaySeats(10);
            say(50,23,"(#) Available (*) Taken");

}
// ============================================
//   main module point of entry to the program
// ============================================
int _tmain(int argc, _TCHAR* argv[])
{
            HANDLE hConsole = GetStdHandle(STD_OUTPUT_HANDLE);  // it creates a handle named hConsole to manage the standard output
            SMALL_RECT windowSize = { 0,0,80,24 };              // it defines the console's dimmensions by giving the top left corner
                                                                // coordinatesand the bottom right corner coordinates

            SetConsoleWindowInfo(hConsole, TRUE, &windowSize);  // it sets the console according to previous statement

            system("color 17");  // it sets the console forground color to White background color to Blue

            char option = ' ';   // it creates and initializes in blank the menu option char var
            bool stay = true;    // it creates a boolean variable to control the program exiting condition and sets it to true
            int i = 0;           // it creates an int variable i for various uses


            readFromFile();      // it reads the values of the text file into the row1..row0 arrays
                                 // if the text file does not exits it saves the values in the row1..row0 arrays to the theaterseats.txt file

            intro();             // it displays the opening theater curtain introduction and the name of the Learning Team C
```

```
        drawChart();         // it displays the theater seats availability and title of the menu
        displayMenu();       // it displays the main menu screen

        while (stay)         // while condition to stay is true it will stay in the loop
        {
           option = chooseTask();  // it accepts the user's menu choice and it assigns it to the variable option

           switch (option)         // it evaluates the menu choice variable
           {
                case '1' :                                      // Selling tickets
                          tickets();                            // it calls the module tickets()
                          displayMenu();                        // it displays the main menu
                          break;                                // it exits the switch statement
                case '2' :                                      // Total of all ticket sales
                          totalTickets();                       // it calls the module totaltickets()
                  displayMenu();                          // it displays the main menu
                          break;                                // it exits the switch statement
                case '3' :                                      // List of tickets sold
                          soldTickets();                        // it calls the module soldTickets()
                          displayMenu();                        // it displays the main menu
                          break;                                // it exits the switch statement
                case '4' :                                      // Available Seats per Row
                          seatsAvailabilityperRow();            // it calls the module seatAvailabilityperRow()
                          displayMenu();                        // it displays the main menu
                          break;                                // it exits the switch statement
                case '5' :                                      // Available Seats in Auditorium
                          seatsAvailinAuditorium();             // it calls the module setsAvailinAuditorium()
                          displayMenu();                        // it displays the main menu
                          break;                                // it exits the switch statement
                case '0' :                                      // Exit the program. [ Esc ]
                          stay = false;                         // it sets the condition to false for exiting the program
                          break;                                // it exits the switch statement
                default :
                                                {
                                          say(3,22,"Exception error. Press any key to exit... "); // it displays the exception error message
                                           getch();                                               // it waits for the user to press a key
                                          }
      }

        }

        saveToFile();  // it saves the theater seats availability to disk to a file theaterseats.txt and
                       // it also saves a backup file named backuptheaterseats.txt

        return 0;
}
```