The TheaterSeatingProgram.cpp C++ console application achieves its goal with a procedural programming approach. The idea is to provide the user with a console application for selling tickets for a small theater. The system will save the seats sold and available to disk with the help of two text files: “theaterseats.txt”, and a backup text file “backuptheaterseats.txt”.

The program will start by looking for the file “theaterseats.txt”, if it does not find the file, it will create it and it will save 100% availability of seats from information stored in ten char arrays named row1…row0. If the file exists it will read the file into these arrays. At the exit of the program a module will save the content of these arrays to the text file “theaterseats.txt”, and “backuptheaterseats.txt” for backup. If something goes wrong, and it cannot save the values of the arrays to disk, the backup file will be available for making a copy with the name of the file “theaterseats.txt”.

The main menu has five choices and an exit choice which can be selected by pressing “0” or the key “ESC”. The other choices are:

1. Selling tickets,
2. Total of all tickets sales,
3. List of sold tickets,
4. Available seats per row,
5. Available seats in auditorium
6. Exit the program [ Esc ].

There is a sentinel boolean variable named “soldOut” which is checked for its status every time the user enters an option from the menu. If there are no available seats at all in the theater for selling, the value of the variable should be true and the program will display a message saying “All the seats in the theater are sold out.” Giving the user the chance to press any key and come back to the main menu.

Also, there is an array for storing available seats per row in the auditorium with the name “auditoriumSeats[]”, type int. Another array type double with ten elements keeps the prices per row, “seatPricesByRow[]”.

At the beginning of the program there is a welcome screen with the name of the program and the names of the Learning Team C members. After the user press any key to start the program, an intro() module will give the effect of theater opening curtain, and it will stay open until the end of the program.

On the first screen, on the top left half screen will appear “M A I N M E N U” inside a box, and on the same side at the bottom the two lines box for communication between the user and the program. At the same time on the left side of the screen it will appear the choices for the main menu, and on the right side of the screen it will be the seat availability chart. This chart which is updated at all time for its seat availability shows a matrix where the name of the rows are listed on the left, and the numbers of the columns or seat numbers on top. The possible symbols for this chart are a “#” for available , or “\*” for taken or sold.

The first entry of the menu allows the user to sell any available seat in the theater. It starts asking for the row number from 0-10 and if the user wants to finish, he/she can do it by entering a “0” and pressing [Enter]. The program is built to allow only a number from 0 to 10. It will not accept a number out of that range. If the number entered is different from zero, the program will ask for entering a seat number always from 1-10. The same policy in this case, valid numbers only from 1-10.

The program will check the availability of that seat, and if it is available , it will show a message with the price of the location. It will also ask for confirmation of the sale. There is a confirmation module “char confirm()” that will ask for a response from the user, allowing only the following characters “YnNn”. If the user confirms the sale by pressing “Y” or “y”, then a module will update the chart (array) and refresh the availability chart on screen.

The user will remain in the loop of entering a row number while there are still available seats and he/she does not press “0” for returning to the main menu.

The second entry will display row by row the amount in $ for sales, and with the total at the end of the last row.

The third entry will display by row the number of the seats sold.

The fourth entry will show row by row the number of the seats available.

The fifth entry will show number of seats available per row, and the sum of available seats at the end of the rows.