**C/C++ RESERVATION SYSTEM LAB REPORT**

**1) Enter your name, student ID, platform (Mac or PC) and date**

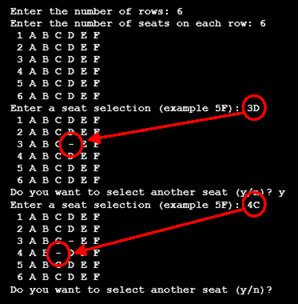
Name: Samuel Indurkar, 0888068

Class: CIS054 C/C++ Programming

Platform (Mac or PC): gcc eclipse MAC  
Date: 7/13/2017

**DESCRIPTION:**

Write a program that reserves seats for a theater, airline, etc. using a two-dimensional array. The program should start by asking for the number of rows and seats on each row and display the available seats. The rows are to be identified by a number starting at '1'. The seats on each row are to be identified by a letter, 'A', 'B', 'C', etc. For example, 3C, 4A and 2D are sample seat identifiers.

Because the total number of rows and seats is unknown when the program first starts, it will be necessary for the program to dynamically allocate memory for the seating array and then release that memory when the program is ready to end.

The program should ask the user to select row and seat. The program needs to verify and process only legal row and seat requests. If the seat is already reserved, the program should notify the user that the seat is taken. If the seat is available, the program should replace the seat identifier with a dash character '-'. The program should contain a loop to ask if additional seats are to be reserved. The program should also give a message when all the seats are taken. A counter can be used to keep track of the total number of seats purchased.

When running the program, you must use a minimum  
of 3 rows and 4 seats. The program output part of the  
lab report must show these minimums. The program output must also show a sample display of when more than one seat is taken, and another display showing a message when all seats are taken.

**LAB REPORT:**

**2) Fill in the HIPO chart using English or pseudo-code. Do NOT paste your program or any part of it in the PROCESSING section.**

|  |  |  |
| --- | --- | --- |
| **INPUTS** | **PROCESSING** | **OUTPUTS** |
| Ask user the max rows and max seats. Then ask user for seat in the form of 1a, 2b, 3c etc | allocate memory for the seating array based on rows and seats.  process legal row and seat requests. If the seat is already reserved, notify the user that the seat is taken. If the seat is available, decrement the “availableSeatCount” and replace the seat identifier with a dash character '-'. | display seating map. |

**DISCUSSION:**

**3) Complete the DISCUSSION section. It does not need to be long, but it needs to be complete.**3a) What did you do to develop the program? ("Followed the Directions" is not a complete description)

Most of the code was provided. I only validated the user input for valid seat by comparing it with allowed rows and seats.

Also, I created a new variable to keep track of available seats. Each time a seat gets taken I decrement the available seat count.

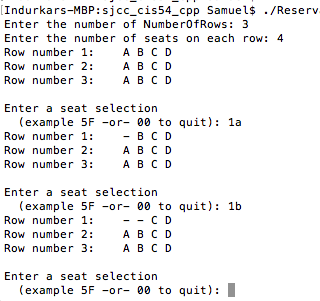
3b) What problems did you have and how did you overcome the problems?

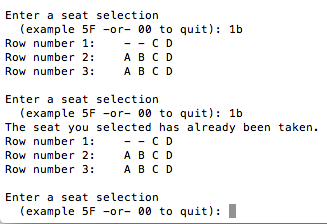
The problem I had was for a 4 by 4 array, even when user entered row 5 it accepted that illegal input. I realized the array goes from index 0 not 1, so subtracting 1 fixed the problem.

**PROGRAM OUTPUT:**

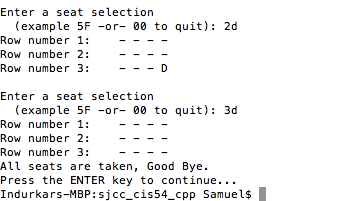
**4) Show a screen shot for purchases of at least TWO seats and another screen shot with a message that shows what happens when ALL seats have been purchased..**

Refer to previous lab assignments for instructions on how to capture a screen or portions of a screen for either the PC or a Mac



**That seat is already taken**

**ALL Seats are full:**

****

**PROGRAM LISTING:**

**5) Copy and paste the code that YOU typed to make the program work. Your program should include a comment block at the top that shows the name of the program, date, version and your name.**

do

{

cout << endl << "Enter a seat selection" << endl << " (example 5F -or- 00 to quit): ";

cin >> rowSelection; // get row from the user

cin >> seatSelection; // get the seat from the user

if (rowSelection=='0')

continue; // skip the rest of the loop

seatSelection = toupper(seatSelection); // convert to upper case

row = rowSelection - '1'; // count from zero to work with the array

seat = seatSelection - 'A'; // convert 'A' to 0, 'B' to 1, 'C' to 2, etc.

if ( (row > NumberOfRows-1) || (seat > NumberOfSeats-1))

{

cout << "INVALID Seat. Please enter a valid seat number\n";

DisplayArrayOfSeats (ArrayOfSeats, NumberOfRows, NumberOfSeats);

continue;

}

if (ArrayOfSeats[row][seat] == '-')

{

cout << "The seat you selected has already been taken." << endl;

DisplayArrayOfSeats (ArrayOfSeats, NumberOfRows, NumberOfSeats);

continue;

}

else

ArrayOfSeats[row][seat] = '-';

DisplayArrayOfSeats (ArrayOfSeats, NumberOfRows, NumberOfSeats);

seatCount--;

if (seatCount == 0) {

cout << "All seats are taken, Good Bye." << endl;

}

} while (rowSelection != '0' && seatCount);