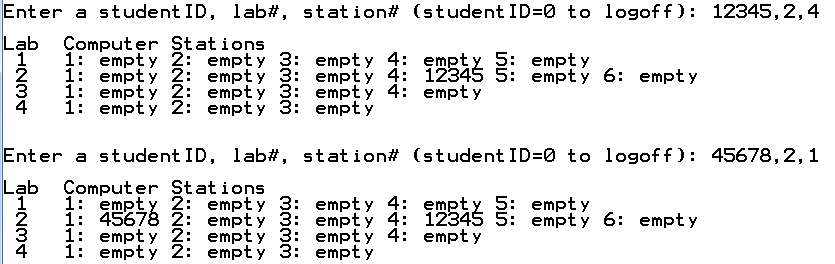
**C/C++ COMPUTER STATION LOGIN LAB REPORT**

**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): MAC  
Date: 7/14/2017

**DESCRIPTION:**

This project contains fairly advanced topics using pointers to create an array of dynamic arrays. It uses a multi-dimensional jagged array. It is called jagged because the rows in the array are not necessarily the same size.

Refer to the textbook, end of chapter problems for Chapter 9

- Eight Edition – Chapter 9 problem 5, pages 520-521

- Ninth Edition – Chapter 9 problem 5, pages 535-536

**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** |
| userID, stationNumber, labNumber | first create a two dimension array of labs[4] and stations{ 5,6,4,3}  then initialize all array locations with 0  Receive user input.  validate user input.  assign userID into the multi-dimension array. | display current status of labs and stations. |

**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)

first create a two dimension array of labs[4] and stations{ 5,6,4,3}

then initialize all array locations with 0

Then ask the user for input as "Enter a studentID lab# station# "

Example: To logon: '12345 2 3' To logoff: '0 2 3'

Receive user input.

validate user input.

subtract 1 to convert to base 0 because array index starts from 0 not 1.

assign userID into the multi-dimension array.

display current status of all labs and stations.

loop back and ask the user for next input

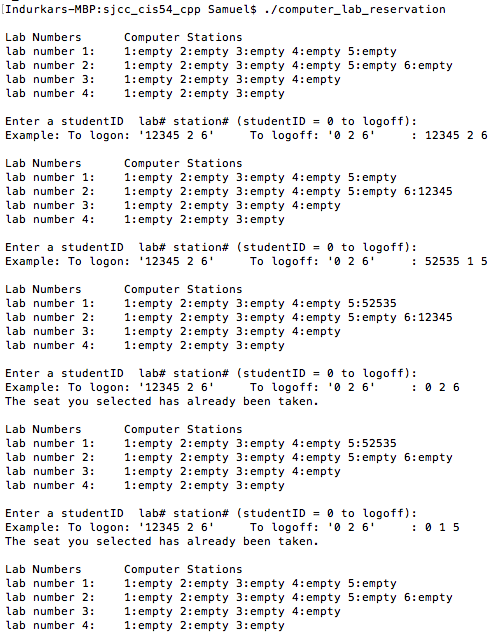
3b) What problems did you have and how did you overcome the problems?

one problem I faced was, the user enters input starting from 1, but the array index starts from 0. during first run, instead of lab1, station1, it reserved lab2, station2 because of index miscalculation. I overcame this problem by subtracting 1.

**PROGRAM OUTPUT:**

**4) Show screen shots for at least TWO sample purchases of the program execution.**

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

****

**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.**

/\*

\* computer\_lab\_reservation.cpp

\*

\* Created on: Jul 14, 2017

\* Author: Samuel

\*/

#include <iostream>

#include <cctype>

using namespace std;

// Function declarations (prototypes)

int \*\* CreateArrayOfSeats (int NumberOfLabs, int \*NumberOfSeats);

void InitializeSeats (int \*\*ArrayOfSeats, int NumberOfLabs, int \*NumberOfSeats);

void DisplayArrayOfSeats (int \*\*ArrayOfSeats, int NumberOfLabs, int \*NumberOfSeats);

void MemoryCleanup (int \*\*ArrayOfSeats, int NumberOfLabs, int \*NumberOfSeats);

int main(int argc, char\* argv[])

{

int \*\*ArrayOfSeats;

int NumberOfLabs = 4;

int NumberOfSeats[] = {5, 6, 4, 3};

int userID;

int labNum; // index into ArrayOfSeats, 0 to NumberOfLabs-1

int station; // index into ArrayOfSeats, 0 to seats-1

int seatCount;

seatCount = 0;

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

{

seatCount += NumberOfSeats[i];

}

ArrayOfSeats = CreateArrayOfSeats(NumberOfLabs, NumberOfSeats);

InitializeSeats (ArrayOfSeats, NumberOfLabs, NumberOfSeats);

DisplayArrayOfSeats (ArrayOfSeats, NumberOfLabs, NumberOfSeats);

do

{

cout << endl << "Enter a studentID lab# station# (studentID = 0 to logoff): " << endl;

cout << "Example: To logon: '12345 2 6' To logoff: '0 2 6' : ";

cin >> userID;

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

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

labNum--; // convert to base 0;

station--; // convert to base 0;

if ( (labNum > NumberOfLabs-1) || (station > (NumberOfSeats[labNum])-1))

{

cout << "INVALID data. Please enter a valid number within range\n";

DisplayArrayOfSeats (ArrayOfSeats, NumberOfLabs, NumberOfSeats);

continue;

}

if (ArrayOfSeats[labNum][station])

{

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

ArrayOfSeats[labNum][station] = 0;

seatCount++;

DisplayArrayOfSeats (ArrayOfSeats, NumberOfLabs, NumberOfSeats);

continue;

}

else

{

ArrayOfSeats[labNum][station] = userID;

seatCount--;

DisplayArrayOfSeats (ArrayOfSeats, NumberOfLabs, NumberOfSeats);

}

if (seatCount == 0) {

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

}

} while ( seatCount);

MemoryCleanup (ArrayOfSeats, NumberOfLabs, NumberOfSeats); // return the memory that was allocated

cout << "Press the ENTER key to continue...\n";

char buff[100];

cin.getline (buff, 100);

return 0;

}

int \*\*CreateArrayOfSeats (int NumberOfLabs, int \*NumberOfSeats) // \*\* means pointer to pointers

{

int \*\*ArrayOfSeats;

ArrayOfSeats = new int\*[NumberOfLabs]; // create an array of pointers for the NumberOfLabs

for(int r = 0; r < NumberOfLabs; r++)

{

int seats = NumberOfSeats[r];

ArrayOfSeats[r] = new int[seats]; // create an array of seats for each row

}

return ArrayOfSeats; // return pointer to the array back to the main program

}

void InitializeSeats (int \*\*ArrayOfSeats, int NumberOfLabs, int \*NumberOfSeats)

{

for (int r=0; r<NumberOfLabs; r++) // initialize the data for each row

{

int seats = NumberOfSeats[r];

for (int s=0; s<seats; s++)

ArrayOfSeats[r][s] = 0; // put 0 in each row

}

}

void DisplayArrayOfSeats (int \*\*ArrayOfSeats, int NumberOfLabs, int \*NumberOfSeats)

{

cout << endl;

cout << "Lab Numbers Computer Stations\n";

for (int r=0; r<NumberOfLabs; r++) // for each row

{

cout.width(8);

cout << "lab number " << r+1 << ": "; // Display row numbers starting from 1

int seats = NumberOfSeats[r];

for (int s=0; s<seats; s++)

{

cout << s+1 << ":";

if (ArrayOfSeats[r][s] == 0)

{

cout << "empty ";

}

else

{

cout << ArrayOfSeats[r][s] << ' '; // Display info for each seat

}

}

cout << endl; // new line after each row

}

}

void MemoryCleanup (int \*\*ArrayOfSeats, int NumberOfLabs, int \*NumberOfSeats)

{

for (int r=0; r<NumberOfLabs; r++)

delete [] ArrayOfSeats[r]; // delete each row of seats individually

delete [] ArrayOfSeats; // delete the row array

}