Project Four
Airline Problem

Write a C++ program to assign passenger seats in an airplane. *Chesapeake Airlines* is a small commuter airline with seats for 36 passengers in 9 rows of 4 seats each. See the sample display below for a layout of the plane.

The user enters the row (1 – 9) and the seat (A – D). The program checks the array to see if the seat is available. An **X** in the array indicates the seat is not available.

If the seat is available, assign an **X** to that position in the array. If it's unavailable, display a message to the passenger.

There are 3 ticket classes:

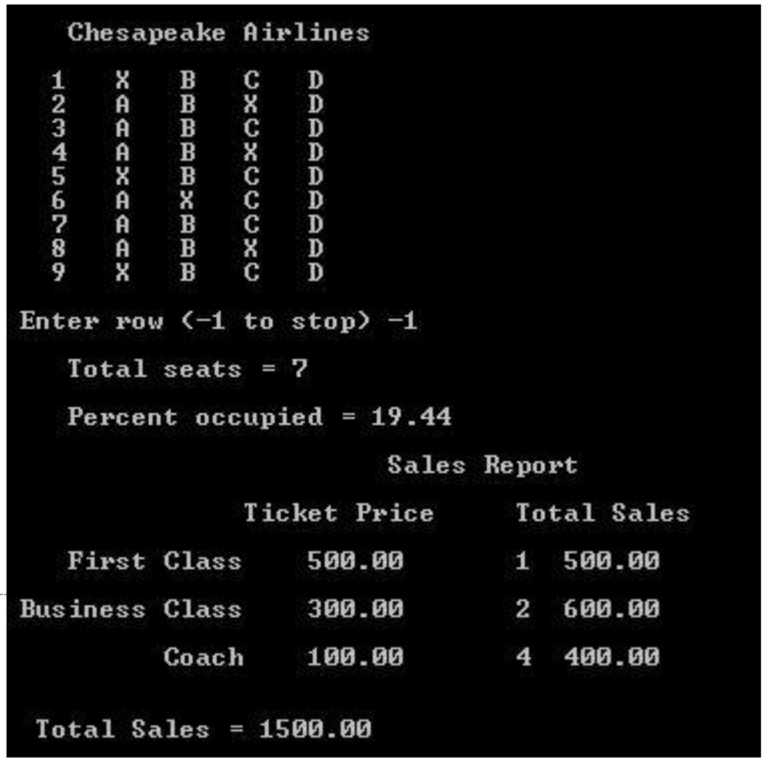
Row 1	= First Class
Rows 2-4	= Business Class
Rows 5-9	= Coach

*Note that the rows are 0-8 in the C++ program.

Use the *classCtr* array supplied in the header file to keep track of how many seats in each class are purchased.

Continue processing requests until the user enters -1 for row. After -1 is entered for the row, display

- number of seats sold.
- percentage occupied.
- sales report.



```
   Chesapeake Airlines

  1   X   B   C   D
  2   A   B   X   D
  3   A   B   C   D
  4   A   B   X   D
  5   X   B   C   D
  6   A   X   C   D
  7   A   B   C   D
  8   A   B   X   D
  9   X   B   C   D

Enter row (-1 to stop) -1

   Total seats = 7

   Percent occupied = 19.44

                        Sales Report

              Ticket Price      Total Sales

   First Class     500.00        1   500.00

Business Class     300.00        2   600.00

         Coach     100.00        4   400.00

 Total Sales = 1500.00
```

You need to include the following functions in your program.

- Function to get the row and seat.

This function displays a prompt requesting the value for row. If **-1** is not entered for the row, the function displays a prompt requesting the seat letter. The seat can be an upper or lowercase letter. Check how the *toupper* function works in Chapter Six (Display 6.9).

This function is a **call-by-reference** since it needs to supply main with two values.
See the function declaration *getData* in the cpp file provided.

Check Display 5.4 getNumbers function and Display 5-9 getInput function.

These two functions are **call-by-value**.

- Function to display the plane layout using the *layout* array provided in the header file.
  See the function declaration *displayPlane* in the cpp file provided.

- Function to display the Sales Report, using the *classes*, *classCtr*, and *fares* arrays provided.

- Any additional functions that make the program more modular.

I have included 2 files to help you get started.

airlineArrays.cpp file - Use this file for your program statements.

```
#include<iostream>
#include<string>
#include <iomanip>
using namespace std;

#include "array.h"

void displayPlane(char [ROW][COL]);
void getData(int & , char & );

int main()
{

}
```

array.h file – This file contains the arrays needed for this project.

```
//arrays for airline problem

const int ROW = 9;
const int COL = 4;
const int CTR = 3;

//initial seats in the plane
        char layout[ROW][COL] = {  { 'A', 'B', 'C', 'D' },
                                   { 'A', 'B', 'C', 'D' },
                                   { 'A', 'B', 'C', 'D' },
                                   { 'A', 'B', 'C', 'D' },
                                   { 'A', 'B', 'C', 'D' },
                                   { 'A', 'B', 'C', 'D' },
                                   { 'A', 'B', 'C', 'D' },
                                   { 'A', 'B', 'C', 'D' },
                                   { 'A', 'B', 'C', 'D' }};

        int classCtr[CTR] = {0,0,0};

        string classes[] = {"First Class", "Business Class", "Coach"};

        double fare [] = {500, 300, 100};
```

Header files

User-defined header files are useful for reducing redundant code. The array.h file contains all the arrays you need for this project. To incorporate these arrays into your cpp file, use a statement like this:

```
#include "array.h"
```

The program has to know where this file is located. To add the header file: right-click on *Header Files* in the Solution Explorer of the IDE and add the header file to the project so you don't need to specify a drive.

The #include statement is basically like a copy/paste operation. The compiler will *replace* the #include line with the actual contents of the file you're including when it compiles the file. This way, any program that needs these arrays can include them instead of keying the array data in each program.

Develop C++ program with the following new features:

- Arrays
- Arrays and functions
- User defined header files.

Advice from the Instructor:

- As usual, code this problem in small steps, one function at a time.

- In main, the row and seat values need to be converted to indices for the plane layout array.

  The row value needs to be converted to a legal index from 0 to 8.
  You can use one statement to adjust the row index.

  The seat letter needs to be converted to a legal index from 0 to 3. Consider subtracting the ASCII value *65* from the value of the seat entered to change the index. It only takes one statement to adjust the column index using this technique.

  To produce the correct column results:
  'A' – 65 = 0
  'B' – 65 = 1
  'C' – 65 = 2
  'D' – 65 = 3

- The advantage of arrays is the same in C++ as in Java - often reducing the amount of code you need. First make sure your program works correctly and then go back and see if you can write the code more efficiently. For example the function to write the sales report can contain one *for loop* that produces the 3 line items in the report as well totaling total sales.

Do **not** use code like this:

total = classCtr[0] + classCtr[1] + classCtr[2];

Use the power of indexing through an array with a *for* loop.

Use your Java array experience to code this problem.