

**Trent University**  
COIS1020H  
Winter 2019  
**Assignment 4**  
Due: March 22, 2019

Air Canada is looking to replace their ailing airline reservation system and have hired you to make a prototype that incorporates a few of the features. Using a menu-driven interface as in Assignment 3, write a C# program that uses an array of strings to represent the seats assignments on a airplane. Assuming for the prototype there are 10 seats (numbered 0-9), use the indices of the array to represent the seat (i.e., array location 0 would be Seat 0, array location 1 would be Seat 1, etc. To make a seat assignment, find an empty seat in the array (represented with the null string "") and store the customer's name in the array location. Each location in the array should be initialized to "".

The seating array is to be declared in the `Main()` method which is to use a *do-while* loop to repeatedly prompt the user to enter a command (*char*). The possible commands are 'b' or 'B' for book a seat, 'c' or 'C' cancel a seat, 'p' or 'P' for print the seating assignments, and 'q' or 'Q' for quit. Using a *switch* statement in the *do-while* loop, `Main()` is to select the appropriate action and call the appropriate user-defined method. The `Main()` method should continue accepting commands until the user enters a 'Q' or 'q'. The program is to have five user-defined methods:

- (i) An *int* method call **FindEmptySeat** which takes one formal parameter representing the array of strings that stores the seating assignments and returns the array index of the first available seat on the airplane (or -1 if the airplane is full). The first empty seat will be the first array location found that contains the null string "". The method header should look like:

```
public static int FindEmptySeat(string [] SeatAssign)
```

- (ii) An *int* method call **FindCustomerSeat** which takes two formal parameters representing the array of strings that stores the seating assignments and a string for the customer to whom you are looking. This method is to return the array index of where the customer's name is found in the array (or -1 if the customer does not have a seat). The method header should look like:

```
public static int FindCustomerSeat(string [] SeatAssign, string cName)
```

- (iii) A *void* function called **Booking** which takes one formal parameter representing the array of strings that stores the seating assignments. This method should prompt the user for a customer's name and then use the method **FindEmptySeat** to find an available seat. If a seat can be found (**FindEmptySeat** return a number between 0 and 9), the customer's name is to be stored in the array at the available seat location and a message printed indicating giving the customer's name and seat location. If a seat cannot be found (**FindEmptySeat** returns a -1), a message should be printed with the customer's name and the fact that the plane is full. The method header should look like:

```
public static void Booking(string [] SeatAssign)
```

- (iv) A *void* function called **Cancel** which takes one formal parameter representing the array of strings that stores the seating assignments. This method should prompt the user for a customer's name and then use the method **FindCustomerSeat** to find the seat number for this customer. If **FindCustomerSeat** return a valid seat number, simply mark the seat as empty (store the null string) and print a message giving the customer's name and the fact that the seat has been cancelled. If **FindCustomerSeat** returns a -1, print a message stating that the customer name does not have seat on this plane. The method header should look like:

```
public static void Cancel(string [] SeatAssign)
```

- (v) A *void* function called **PrintSeats** which takes one formal parameter representing the array of strings that stores the seating assignments. This method should simply print out the Seat Number and associated Customer Name for every occupied seat on the plane. If a seat is empty, the information about that seat should not be printed out. The method header should look like:
- ```
public static void PrintSeat(string [] SeatAssign)
```

For this assignment, you are to submit a properly documented source code file (i.e. .cs file). This source code file is to include comments at the top containing your name, student number, a description of the class, and comments within the body of your class (inline comments). You are also required to submit a PDF of your testing documentation using the template that is provided on BlackBoard. Failure to use the testing template or submit as a PDF could result in a 0 for the testing component of the assignment. These 2 files are to be attached to the Assignment 4 DropBox by the due date.