**IFN501 Assessment 2 – Deliverable 2 (Individual)**

**Project (Programming) - Total Weighting 30%**

**Online Submission of Code (25%)**

**Due Date: Sunday 26th May 2019 by 11.59 pm (Week 12)**

**Code Demonstration (5%)**

**Done during your Practical Sessions in Week 13**

**Note 1**

**You are advised to use only the C# concepts taught in this Unit through Lectures and Tutorials. Otherwise, you will incur a penalty and could get zero.**

**Problem Specification:**

Your programming task is to design and develop a **console application** for an airline company that issues boarding passes for an aircraft with 40 seats. You have to design and develop a **C# object-oriented solution** for this task.

The application requires a **class** called **Passenger** that contains passenger information and allocates a seat number based on seat availability to each passenger. The data attributes for the passenger class includes, but not limited to, a security number, departure time, departure gate and other attributes that are required to issue a boarding pass.

You need to construct methods to obtain all information necessary from the passengers.

Your application must meet **all** the following requirements and handle **all run time errors**. 50% of the marks allocated for each requirement will be deducted if your program cannot handle the errors**.**

1. **Passenger names** should not be longer than 5 characters.

The passenger name (i.e. first and last) should be trimmed or truncated to 5 characters if the name provided is greater than 5 characters.

(3 marks)

1. **Security number** on boarding pass should be a random number generated between 30,000 and 999,999.

(3 marks)

1. Each **seat number** must only be allocated to one passenger. For example, if the seat number 14 is allocated to a passenger, then the seat number 14 should not be available for the any other passenger.

(3 marks)

1. The application will continually accept passenger requests; the passenger attributes may vary from passenger to passenger (e.g. each passenger is a new object). Your application should also be able to handle a number of the passengers in a single invoice. For example, Mr John Smith may have bought two tickets for his wife and himself in single invoice. This means the application should be able to accept and allocate adjacent seats for the two passengers. (4 marks)
2. If all seats have been allocated to passengers, the application should not accept any more requests. Notify the Airline Assistant.

(2 marks)

1. **Date and Time** should be printed when passenger entries are completed.

(e.g., system time 24 April 2019).

(1 mark)

1. You should have a method (part of the Passenger class) to list all the passenger names followed by their seat numbers and the security numbers once the flight is closed.

(3 marks)

1. After check-in is completed, **boarding passes** must be displayed in a tabular format for each passenger.

(3 marks)

1. Create a method to display the **available seats** to the Airline Assistant on request anytime. This method should be part of the Passenger class and will be invoked from the Main() method.

(3 marks)

**Note 2**

1. **You are required to use the Object-Oriented concepts from C# taught in Lectures and Tutorials.**

**You are required to create only one class – Passenger, a number of data fields, corresponding properties, the necessary constructors and the required methods.**

**All display methods should be *Read Only*.**

1. **Your program must generate an appropriate error message if any input does not follow the required criteria and then request the user to re-enter the input.**

**Project Deliverables:**

1. **Program Code (25%)**

**Zip** your project file and upload this zipped file onto BB using Assessment 2 Part 2 link under **Assessment**. The link will be available only from **Friday 24 May 2019.** **You can only upload** **once.**

* Your **program** **code** will be evaluated by:

1. Assessing the conformance of the submitted program to the **functional requirements** outlined above. Does the program meet **all** these requirements, and how well?
2. Assessing the "means" of accomplishing the task by examining the **program structure**, **style**, **clarity** and **conciseness**, and if your **code** is written in a way that is **clear** and **maintainable**.
3. **Code Demonstration (5%) – during scheduled practical sessions in Week 13**
4. You will be asked to **execute your code** by **inputting** data provided by us during the practical sessions.
5. You will be asked to **do modifications** to your original code to perform the same functionality.
6. Non-attendance to this session for the code demonstration will result in **zero** mark for this part.

**Note 3**

1. Assessment submitted after the due date without an approved extension will not be marked and will receive a grade of 1 or 0%. If special circumstances prevent you from meeting the assessment due date, you can [apply for an extension](https://qutvirtual4.qut.edu.au/group/student/study/assignments/submitting/late-assignments-and-extensions) student services in the Faculty. If you do not have an approved extension, it is recommended that you submit the work you have completed by the due date, and it will be marked against the assessment criteria.
2. You are allowed to bring **only the softcopy of the code** you submitted online to your practical session; **no notes** are allowed.

-----------------------------------