COMPX101 Practice Test 2 2018a

First Name/Given Name	Username
Surname/Family Name	ID Number

Getting Started

Download the template project for the practice test from Moodle and then unzip the file. Then load the project in Visual Studio.

Assessment

This practical test will be assessed as follows:

10	Pseudo-code completed for the Draw Seats button.
5	Interface matches specifications.
5	Appropriate names for important controls.
5	Variables and constants have appropriate data types, names and scope.
5	Clear and Exit buttons work correctly.
5	Try…Catch used. Catch part displays error, clears textbox and sets focus.
10	Checks if number of rows is valid. Displays error message if not valid, clears textbox and gives it the focus.
10	Draws the A seat in each row of the plane based on user input.
5	Draws correct sized gap between each seat and aisle gap.
5	Draws correct number of first class rows in red.
10	Draws seats B–D in each row in the plane (-5 if does not use nested loops).
5	Creates variables with appropriate names, data types and scope.
5	Created required constants
5	Used constants appropriately.
10	Appropriate documentation as described in Task 8.

Complete Pseudo-code here for the Draw Seats button:

```
Declare variables
TRY




CATCH
   Display Error message
   Clear the textbox
   Set focus to the textbox
ENDTRY
```

Instructions

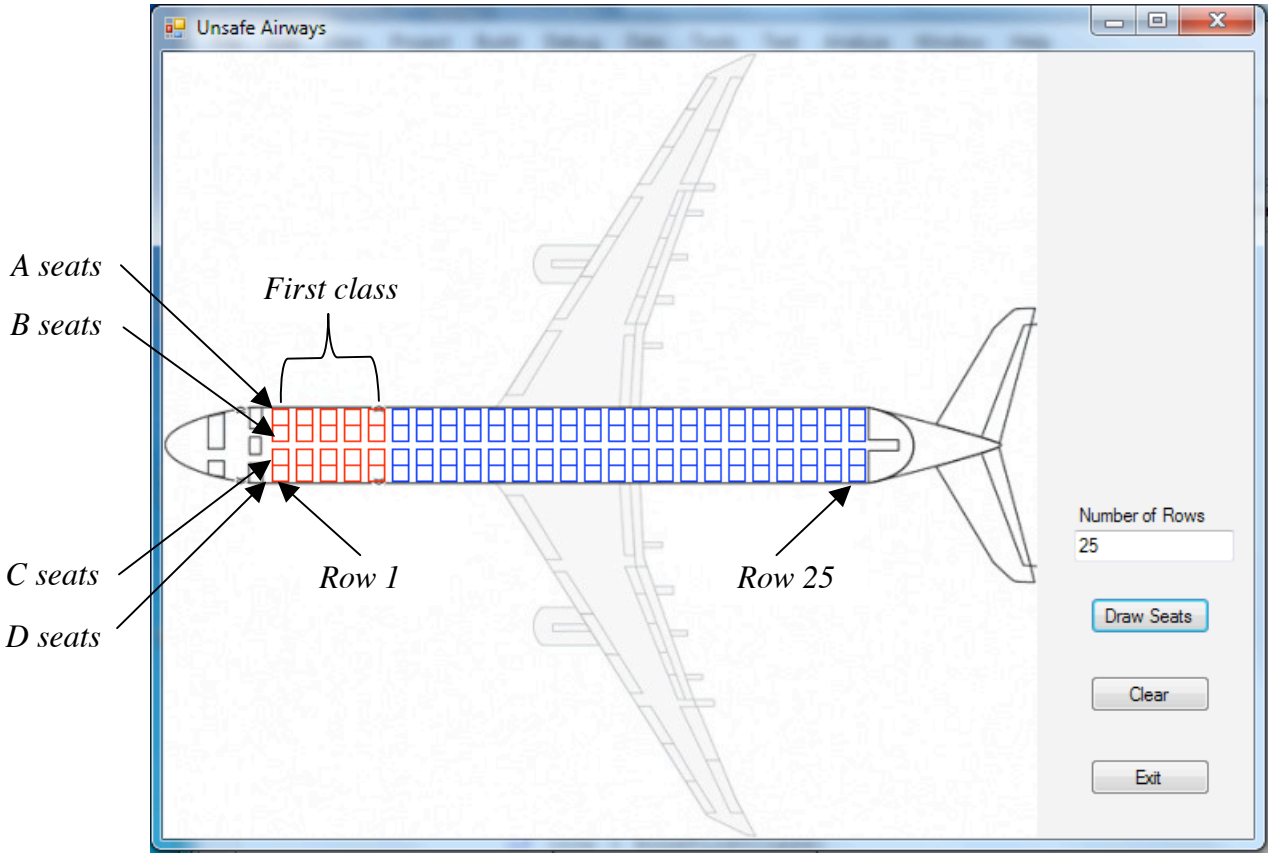
This is an open book test. You have **10** minutes to write the pseudo-code for the application, which must be written in the space provided. If you finish your pseudo-code early then you are allowed to write notes about the problem but you cannot start using Visual Studio until you are told to start. You will have a further **90** minutes to complete as much of the application as you can.

When you have completed the test, make sure that you have a comment at the top of your code which has your name and id number. Document the rest of your code and save your work. Your work will be copied from your account and marked later. **Do Not** attempt to open your work until you have got the mark back for it.

Marks will be awarded based on the performance of your application, appropriate use of C# statements and the documentation of your methods. Thus even a partially completed application can receive some marks.

Problem Description

"Unsafe Airways" is redesigning their aircraft, and want you to create an application to help them design the seating arrangements in their aircraft. The application will draw an aircraft outline, and then allow the user to draw seats inside the aircraft.



The form already contains a picturebox with the plane image. **Do not resize the picturebox.** You will need to add a textbox to enter the number of rows and buttons for **`Draw Seats`**, **`Clear`** and **`Exit`**.

Tasks

1. In the space provided on page 2, complete the pseudo-code for drawing all of the seats in the airplane. The application will get the number of rows from a textbox. It then checks to see if the value entered is between 2 and 25 inclusive. If the value is valid then all of the seats are drawn based on the user's input with the first 5 rows shown in red (first class seats) and the rest shown in blue (economy seats). If the value isn't valid then an error message is shown, the textbox is cleared and the focus is set to the textbox. Use a try..catch structure to deal with any input errors by dispaying an error message, clearing the textbox and setting the focus to the textbox.

2. Create the user interface to match the specifications above.

3. Create the click event method for the **`Exit`** button, which exits the form.

4. Create the click event method for the **`Clear`** button, which will clear the picturebox of any graphics, clears the textbox and sets the focus to the textbox.

5. Create two new constants to store the minimum number of rows with the value 2 and maximum number of rows with the value 25.

6. Create the click event method for the **`Draw Seats`** to draw all of the seats in the airplane using your pseudo-code in Task 1. Don't forget to give the very first seat leg room as well.

Hint: Constants have been provided for you so use them where appropriate.

Hint: You may find it easier to just draw all of the A seats first and then draw the rest of the seats.

Hint: You may find it easier to draw all the seats in blue first and test to make sure all the seats are drawn and then add the code to draw the first class seats in red.

The first seat starts at position `SeatStartX, SeatStartY` and is of size `SeatWidth, SeatLength` and each following seat should be towards the tail, i.e. immediately to the right. This represents all the A seats in the aircraft, i.e. the first (window) seat in each row of the plane. **At this stage don't worry if the number of rows don't fill up the airplane or some rows are drawn out side of the airplane, you will fix this in the next task.**

7. Calculate the correct gap size for the given number of rows by dividing the `PlaneLength` by the number of rows entered by the user first and then subtracting the `SeatLength`. Alter your code from task 6 to use this new gap size.

8. Document your application as stated below:

   - You should have your name and ID as a comment at the top of your code.
   - Each click event method should have a comment header describing what the method does. Type in 3 slashes '///' and C# will automatically put in a template comment header for the click event method. Then type in a description of the click event method in the summary section.
   - Each variable should have a comment describing what it stores.
   - Appropriate inline comments in the code describing what the code is doing.