



Assessment Information

Qualification National Code and Title	ICT50715 Diploma of Software Development
Unit National Code and Title	ICTPRG418 Apply intermediate programming skills in another language
Semester & Year	Semester 1, 2018
Assessment number and title	Assessment 2: 2D Arrays program
Assessment Description	This assessment requires you to develop an program that meets the given requirements. You will be demonstrating using data structures, file i/o, sorting and searching algorithms, and the application development / build / test process.
Assessment Venue: (if applicable)	Classroom
Assessment Time/duration	Normal class time; sessions 8-9
Due Date	End of week 11
Assessment Instructions	<ol style="list-style-type: none">1. Complete all the assessment tasks below.2. Observation by your lecturer of you doing the assessment is considered part of the assessment process.3. Submit your documentation into the Blackboard assessments area.4. All skills must be demonstrated to achieve a satisfactory result.5. All work submitted must be your own individual effort.



Student Name and ID	
Qualification code and title	ICT50715 Diploma of Software Development
Unit code and title	ICTPRG418 Apply intermediate programming skills in another language
Assessment number and title	Assessment 2: 2D Arrays program

Assessment Task

Read the requirements specification for the Flight Seating Allocation (described in the Appendix) and then complete the following tasks.

1. Write a Java application that provides Seat Allocation functionality as per the requirements provided. Design and develop a solution that consists of multiple source code files. (Each class is to be in its own file.)
2. Define a class to store all the Customer details.
3. Use a 2 dimensional array to store seat allocations for the flight. Every seat in the array must indicate whether it is vacant or which customer it is assigned to.
4. The program must be designed to have a frontend or menu for the user to interact with. The user must be able to add or cancel seat allocation bookings. For a new booking, design the algorithm so that it looks for an available seat that matches the customer's preferences. To demonstrate pass-by-reference parameter passing, the code should invoke a method called `AllocateSeat()` that accepts a Customer object as a parameter and then updates the 2D array. Similarly, there should be a `CancelSeatAllocation()` method.
5. Each booking or cancelation update should generate a file containing the seat allocation information in the specified format. The file should be a fixed length format. Use a random access technique to write the change to the file.
6. Copy the customers to a single dimensional array. Sort the array data in alphabetical order of customer name.
7. Provide a facility for searching for a customer by name. Use a Binary Search algorithm to search through the previously sorted array.
8. Please adhere to the Java coding standard provided, including commenting your code.
9. Build the application. Use the debugging tool to debug your program. Interactively test your program to ensure that it works correctly.

Compress your program folder into Zip format and upload to the Blackboard Assessments area.



Appendix: Requirements for Flight Seating Allocation

A startup business has purchased a plane which they intend to use to provide a daily site-seeing flight to tourists. The airplane has 12 rows, with 6 seats in each row. Seats labelled A, B and C are on one side of the aisle and seats labelled D, E, F are on the other side. The first 2 rows are first class; rows 3 to 6 are business class; and remaining rows are economy class.

To make a booking, a customer needs to provide the following information:

- Their name
- Whether they are an Adult or a Child
- Their class type (ie. first, business or economy class)
- Their preferred seat type (window, aisle or middle).

For simplicity purposes, you can assume that only the flight for the next day can be booked. (ie. There is no need to keep records for multiple flights.)

A booking application is required to keep records of which seat is allocated to which customer. The application must allow the user to create new seat allocations and also to cancel an existing seat allocation. The application must also be able to search for a customer by name and show their seat allocation.

The program must output the seating plan in the following format (where '*' or '-' indicates that a seat is vacant, 'A' indicates that the seat is allocated to an adult, and 'C' indicates that the seat is allocated to a child).

	A	B	C	D	E	F
Row 1	*	*	A	*	A	A
Row 2	*	A	*	A	*	A
Row 3	*	*	A	C	*	A
Row 4	C	*	A	*	A	A
Row 5	*	A	*	A	*	*
Row 6	*	A	*	*	*	A
Row 7	A	*	*	*	A	C
Row 8	*	A	*	A	C	*
Row 9	A	*	A	A	*	A
Row 10	*	A	*	A	C	C
Row 11	*	*	A	*	A	*
Row 12	*	*	A	A	*	A

The above format is how the data should be presented to the user (either on screen or by writing it to a text file). In addition, the bookings data must be able to be saved to a file in fixed length format (which does not need to match the above output format). ie. File storage will be used instead of a database to save the data.