Description: OC_Masterbrand_CMYK.eps

**in partnership with**



**ICTPRG414 Apply Introductory programming skills in another language**

**Assessment**

**Assessment Number: 33126/06**

Student Name: Lance Flatman

Student Number: E0822930

Assessment Number: 33126/06

*© Open Colleges Pty Ltd, 2016*

*All rights reserved. No part of the material protected by this copyright may be reproduced or utilised in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without permission in writing from the copyright owner.*

*All terms mentioned in this text that are known to be trademarks or service marks have been appropriately capitalised. Use of a term in this text should not be regarded as affecting the validity of any trademark or service mark.*

## Introduction

When you feel confident that you have covered the learning materials for this unit, you are ready to attempt this assessment.

To complete this assessment, please create the code to each criteria or task including your comments in the code.

Your code must be created in Eclipse using Java. To help Open Colleges manage your assessment files, please use an appropriate file naming convention that is clear and easy to store.

## Assessment submission

Respond to all questions in this assessment. When you are ready to submit your assessment, place all your files in a zip folder. Name the folder:

ICTPRG414\_AssessmentNumber\_yourName\_studentNumber.

(For example, ICTPRG414\_3312606\_JohnSmith\_17756433).

Upload the file to OpenSpace using the assessment upload link in this module of your course.

If you need further assistance, the Student Lounge provides a ‘Quick Guide to Uploading Assessments’. Uploading assessments in OpenSpace enables Open Colleges to provide you with the fastest feedback and grading for your assessment.

Please ensure that you keep a copy of all assessments you submit to Open Colleges.

## Getting started

In the unit ICTPRG414 Apply introductory skills in another language there are two assessments you will need to complete satisfactorily to provide evidence that demonstrates your competence.

Assessment 33126/05 is a de-bugging exercise  
Assessment 33126/06 is a project where you will create a program in Java

## Your trainer will:

* answer any questions that you have about the assessment
* assess your competence as stipulated in the unit of competency by making judgments about the evidence you have presented in line with the rules of evidence – validity, authenticity, currency and sufficiency
* provide feedback on the outcomes of the assessment process.

## Scenario

### Part 1 – Create a program

Congratulations, you are now working for the software company Code Empire! For your first project you will be building a program named BirdArrayList that will allow the end user to sort birds by their name. The program will also allow users to add, delete, search, display and include a quit function.

Read through this document, then watch the video which will help you to formulate how you would code this task. [Click here to access video.](https://opencolleges-1.wistia.com/medias/z1rfvv2xvd)

Code Empire has provided you with a requirements file (BirdArrayRequirements.txt) which contains comments that will help you with coding your application. You will need to create a program to include an array of 5 birds. Structure your program using the step by step points below.

1. Create a Java project named Assessment and a package called Assessment.
2. Create a class called BirdArrayList with a main method.
3. In the Main method you are to:
4. Initialise an ArrayList
5. Add 5 bird names to the ArrayList (you can choose the names of the birds, for example Australian King Parrot)
6. Create an input dialog to ask the user to either:
   1. Add a bird to the ArrayList
   2. Remove an element in the ArrayList
   3. Search for an element in the ArrayList
   4. Display the ArrayList
   5. Quit the program.
7. Call the appropriate method for the entered selection
8. Keep asking the above selection until they quit the program.
9. You will need to create a method named addEntry() which will:
   1. Add a new bird name to the ArrayList that has been supplied from an input dialog
   2. Display the ArrayList when the new bird name has been added.

7

1. You will need to create a method named deleteEntry() that will:
   1. Use a sequential search (for loop) to find the bird name to be deleted that has been supplied from an input dialog
   2. If the bird name to be deleted is found in the ArrayList, confirm if it is to be deleted and if so, remove the bird name from the ArrayList
   3. If not found, display an error message
   4. Finally, display the array.
2. You will need to create a method to search the ArrayList.
   1. Use a binary search algorithm for the search method which includes a sort function. Do not use the Java.util.Arrays.binarySearch() method, but write the binary search algorithm code.
   2. Display the array element and the array index.
3. You will need to create a method to display the elements of an ArrayList.
4. All program code must contain:
5. comments in program code that contain details of the program
6. comments attributing code source where you are not the original code author
7. variables that are written in the lowerCamelCase naming convention, except where loop control variables are used
8. methods with Javadocs comments
9. other comments within methods that explain significant logic or points reached in the logic.
10. When you have completed coding your program, generate Javadoc documentation.

### Part 2 – Create testing documentation

You are required to test your program throughout the development cycle using de-bugging tools and complete the [testing template](https://app.box.com/s/jftxeki12rtofgrr7zpz7k0u486f9f6l/1/8600772509/71855153077/1) [p](https://share.tafensw.edu.au/share/file/14c5e8ee-980f-418c-b191-46eb22362135/1/ICTPRG414_Testing_template_v1.docx)rovided. Rename the testing template document to TestCase\_yourName\_studentNumber.docx. (For example, TestCase\_JohnSmith\_17756433.docx).

For the testing document, enter the name of the application, your name, date the test was carried out and a brief description of what you are going to test in the Test Specifications. For each test, assign a sequential number to the Requirement Number column, what you will input into the application and the result you expect to see. Signify if the result was correct. For example,

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. #** | **Input** | **Expected Output** | **Result** |
| 1 | Select A from the menu, enter Eagle. | Select D from the menu, the bird name Eagle should be listed last. | Correct |

Create at least four test cases using the testing template to test the outcomes of the following:

1. Add an element
2. Remove an element
3. Search for an element
4. Quit the program

You can do multiple test for each of the above.

When you have completed the project, zip all your project files (follow the naming convention suggested at the beginning of the assessment) and upload, including the Javadoc documentation (BirdArrayList.html) and testing documentation.

Note: The application testing is an important part of developing your program. Evidence that you have conducted testing needs to be completed. Remember to include Part 2 create application testing documentation as you are developing your project.