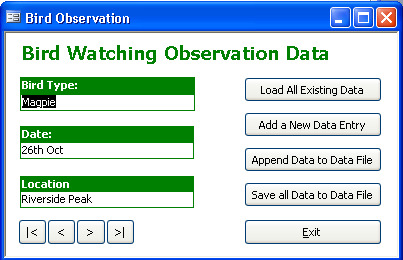
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Student Name | Nathan Robertson | | Student Number | | 450380548 |
| Unit Code/s & Name/s | ICTPRG418 Apply intermediate programming skills in another language | | | | |
| Assessment Name | Class Test | | Assessment Task No. | | AT3 |
| Date of test/exam | 10/8/2018 | | | | |
| **Student Declaration:**  I declare that this assessment is my own work. I am aware of and understand the rules related to assessment as outlined in TAFE Queensland Student Rules and acknowledge that failure to comply with these rules will be regarded as misconduct and will be subject to disciplinary action as outlined. | | | | | |
| Student Signature | Nathan | | | Date | 10/09/2018 |
| Assessor Feedback:  Student provided with feedback *(check box when completed)* | | | | | |
| Attempt 1 | Satisfactory | Unsatisfactory | | Date | / / |
| Attempt 2 | Satisfactory | Unsatisfactory | | Date | / / |
| Assessor Name | David Hunt | Assessor Signature | |  | |
| Note to assessor: Please record any reasonable adjustment below that has occurred during this assessment. E.g. written assessment given orally; scribe provided. | | | | | |
|  | | | | | |
| **PRIVACY DISCLAIMER:** TAFE Queensland is collecting your personal information for assessment purposes. The information will only be accessed by authorised employees of TAFE Queensland. Some of this information may be given to the Australian Skills Quality Authority (ASQA) or its successor and/or TAFE Queensland for audit and/or reporting purposes. Your information will not be given to any other person or agency unless you have given us written permission or we are required by law. | | | | | |

|  |  |
| --- | --- |
| Assessment rules | Only students enrolled in the unit of competency, the Assessor / Supervisor, and other authorised personnel may enter or remain in the room during a written test / exam. The Assessor/ Supervisor may ask you to produce photographic identification (e.g. student ID card, driver’s licence).  Unless approved by the Assessor / Supervisor prior to the written test / exam (e.g. for open-book exams) you may not bring any devices capable of conveying information relevant to the content (e.g. text books, course notes, mobile phones, pagers, notebook computers, and other devices). You must ensure mobile phones and other electronic devices are turned off prior to the commencement of the written test / exam.  You are required to comply with all directions:   1. Detailed in assessment material supplied; 2. Set out on any notice displayed in the room; and 3. Given by the supervisor.   During a written test / exam session you may not:   1. Communicate with any person other than the supervisor; 2. Assist another person to communicate with another person; and 3. Willingly receive communication from any person except with the approval of the supervisor.   Unless permitted by the supervisor, you may not take from the room any papers or other materials provided for use during the written test / exam.  You are expected to be considerate of other students when entering or leaving the room or when in the vicinity of the room.  If you consider that your performance in the written test / exam has been adversely affected by illness, disability, bereavement or other exceptional circumstances you may apply for special consideration. **For more information, refer to the Student Rules.** |
| Instructions to Student | **Number of Questions:** 15  **Time Allowed:** 4 hours  **Examination Conditions:**  This is a closed book examination; All questions must be attempted. |

|  |  |
| --- | --- |
|  | **Materials to be supplied:**  Examination paper  **Materials to be supplied by the Student:**  Paper for recording answers  **General Instructions:**  You are required to answer each of the questions provided. You must use a black or blue pen to provide answers, not pencil. Sketches, however, may be in pencil.  **Calculators:**  Calculators may be used during this examination. Before the examination commences, all memories must be fully cleared and programs erased.  **Number of Attempts:**  You will receive up to two (2) attempts at this assessment task. Should your 1st attempt be unsatisfactory (U), your teacher will provide feedback and discuss the relevant questions with you and will arrange a date your 2nd attempt. If your 2nd attempt is unsatisfactory (U), or you fail to attend the scheduled date for a 2nd attempt, you will receive an overall unsatisfactory result for this assessment task. Only one re-assessment attempt may be granted for each assessment task, with the exception of Apprentices or Trainees who are permitted an additional supplementary assessment. **For more information, refer to the Student Rules.** |
| Instructions for the Assessor | This is a closed book examination.Remind students of TAFE Queensland Student Rules assessment requirements. |
| Submission details  (if relevant) | Students are to submit written answers to the teacher at the end of the exam. |
| Note to Student | An overview of all Assessment Tasks relevant to this unit is located in the Unit Study Guide. |

You have been asked to create a simple program that allows the user to enter birds that they sight, the date and location. On pressing the button: Append Data to Data File, the program copies the three entries to a new line in the data file which is later read into a spreadsheet program for analysis.



Java screenshot used with permission from [Oracle](https://www.oracle.com/legal/copyright.html).

In relation to this program, you are required to answer the following questions:

**Introductory**

* 1. Write down a portion of code that would be used to save a line of Bird data.

To save to a file:

StringBuilder sb = new StringBuilder();

Sb.AppendLine(birdType + “,” + date + “,” + location);

File.AppendAllText(@”fileLocation”, sb.ToString());

To save in the application:

List<Bird> data = new List<Bird>();

Bird bird = new Bird(“Magpie”, “26 Oct”, “Riverside Peak”);

Data.add(bird);

* 1. Explain in detail where the principles of Modularisation (Cohesion and Coupling) and/or Object Oriented Programming could be applied in this program.

The principles of Modularisation can be applied by creating a Bird class that holds all of the observation data, this will allow us to store the observation data in an Array and will allow us to access or sort through the data efficiently.

1.3 In searching for an error in the saving routine, how might you use your IDE’s debugging tool?

By setting breakpoints and variable watches throughout the code, to help you step through your application and to set what the variable is set to, helping you locate the bug.

1.4 What Java utility could be used to create detailed technical program documentation?

JavaDoc can be used to create detailed technical program documentation.

1.5 Prepare a test procedure and/or plan for testing this program.

To test this application I would Input the test data making sure the data is saved correctly and that all data is inputted correctly potentially validating any data inputted into the application so not to cause error.

I would prepare a feedback survey and give the application to other people. Afterwards getting them to fill out the feedback survey.

I would then make any necessary changes to the application potentially resulting in another test.

**Intermediate**

2.1 In relation to objects and arrays:

* Draw a class diagram for a BirdWatchingObservation class that would hold one record (entry) of the data for the Bird Watching Observation program. Add at least 3 properties and 6 methods to this diagram.

Bird

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

string:birdType;

date:dateSeen;

string:location;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

getBirdType():string

setBirdType(string):void

getDateSeen():date

setDateSeen(date):void

getLocation():string

setLocation(string):void

* Show how you would declare an array of these Bird Watching Observation Records.

List observations = new List<Bird>();

2.2 In relation to the array of Bird Watching Observation records:

* Write full code or pseudo code for a Delete method that deletes a Bird Watching Observation record. This method should demonstrate appropriate Low Coupling and High Cohesion.

Public DeleteData(Bird bird){

Data.Remove(bird);

}

* Write a line of code for sorting a single array of Bird Types.

List birdTypes = new List<string>();

Birdtypes.Add(“Magpie”);

Birdtypes.Add(“Crow”);

Birdtypes.Add(“Parrot”);

Birdtypes.Sort();

* Explain or give example code for how a binary search could be applied to this sorted array of Bird Types.

A binary search could be applied to this sorted array with each node in the binary tree holding the bird type in alphabetical order.

* Explain or give example code for how this sorted array might be saved to a Random Access File.

WriteToFile(string filename, Bird bird, int pos, int size){

FileStream fs;

BinaryWriter bw;

fs = new FileStream(filename, FileMode.Append, FileAccess.Write);

bw = new BinaryWriter(fs);

fs.Position = pos \* size;

bw.Write(bird.birdType);

bw.Write(bird.dateSeen);

bw.Write(bird.location);

bw.Close();

fs.Close();

}

2.3. Write a line of code that declares and instantiates a 2D array to store 100 entries of Bird Watching data.

Public List<List<Bird>> data = new List<List<Bird>>();

2.4 Outline how 2 functions of your IDE’s debugging tool assist you with your programming?

Breakpoints: Breakpoints can be a valuable asset when programming an application. They allow you to step through your code allowing you to see what is happening during each line of code.

Variable watch: A variable watch allows you to watch a variable, you will be able to see what the variable is set to helping you to find a solution to whatever problem you’re having.

2.5 In preparing a detailed set of documentation standards for a team of developers name four (4) sections or topics that would need to be covered

Planning

Design

Testing

Debugging

2.6 Outline 4 lines of well-considered test data this program. (An example might be: Magpie – 26th Oct – Riverside Peak). Explain why you would use each line.

Magpie – 26th Oct – Riverside Peak

Parrot – 14th May – Smile Park

Crow – 31st Sep – Bushy Forrest

Budgie – 20th Jul – Water Point

I would use each of the lines to test the application making sure it is saving the data correctly.

I would use the 3rd line to test whether the application accepts an invalid date.

Present a pseudo code listing for a method that saves all the Bird Watching Observation data to the data file.

StringBuilder sb = new StringBuilder();

Sb.AppendLine(birdType + “,” + date + “,” + location);

File.AppendAllText(@”fileLocation”, sb.ToString());

**Knowledge Evidence:**

2.8 Answer the following in detail:

1. Using an example, describe static and a dynamic variable and explain the difference between them.

A static variable has memory allocated to it at runtime meaning if the variable requires more memory

1. Describe with an example, a medium-size application. What strategies would you utilise to develop a medium-size application?

In a medium sized application I would use the waterfall development methodology as I like to be well organised and plan everything out before I start development. After planning how I am going to develop the application I would use project management software such as trello to keep track of what tasks I should be doing. I would also use version control software such as github and source tree making it easier to track what I have completed and when.

1. Describe each, with an example, a standard array structure, and standard algorithms for data file handling.

An array is a list of data each with an index.

List<int> numbers = new List<int>(0,1,2,3,4,5,6,7,8,9);

System.IO “Input/Output” handles data files whether its reading from a file or writing to it you will need to reference this class.

1. Describe with two (2) examples, user-defined data structures.

A data structure is the structure in which a computer holds data e.g. int, float, string.

A user defined data structure is a data structure or class that the user creates/defines.

e.g. Bird, Watcher

class Bird{

string type;

string name;

}

Class Watcher {

string name;

int age;

}

2.9 List two (2) development methodologies and their application

Waterfall Development Methodology:

The waterfall development methodology is mainly used for Medium to Big application as they require a lot more planning and development.

Rapid Application Development

The rapid application development is mainly used for small to medium projects as it focus’s more on prototyping and programming than planning.

2.10 What are the main features of programming languages?

The main features of C# are:

Object orientated

Automatic Memory management.

Simple

Cross Platform

**End of assessment**