FIT1051 Assessment Two

Submission deadline: Thursday 18th April 11:55pm AEST (11:55pm MYT) via Moodle

Weight: This assessment is worth 15% of the unit total. Submission of your code is worth 40% of this total, and your interview component (held later in Week 8) is worth 60%

Late Penalty: 10% mark deduction per day

Instructions: Below are the coding tasks that you need to complete for Assessment 2. Your work and your submission should be independent. Please download the IntelliJ project folder below and unzip it. This will provide you with partial code in which to program your answers. Please complete each task in the appropriate section of the partial code, and submit to Moodle upon completion.

The programming portion of this assessment contains 40 marks and is comprised of the following components:

- Code correctness is worth 35 marks:
    - Task 1 is worth 4 marks
    - Task 2 is worth 2 marks
    - Task 3 is worth 7 marks
    - Task 4 is worth 11 marks
    - Task 5 is worth 7 marks
    - Task 6 is worth 4 marks
- Writing good quality code that adheres to the **FIT1051 Coding Standards** is worth 5 marks.

Academic Integrity: Please be reminded of the academic integrity standards that are expected of you at Monash, which were mentioned in Week 1. You should code alone and ask the unit staff for help if needed. Do not post your code in public forums or send your code to anyone. Do not copy/paste code from other sources and present it as your own — this includes use of generative AI tools. Breaching these academic integrity requirements can incur serious penalties.

Task 1 (4 marks)

In the provided file **NameDatabase.java** in the downloaded project, create a new **NameDatabase** class with a single field, **names**, which holds an ArrayList of String values (1 mark).

Implement the following methods for the **NameDatabase** class:

- (1 mark) a default and a non-default constructor.
- (1 mark) a getter and setter method for the **names** field
- (0.5 marks) a method **public void addName(String name)** which allows the user to add a new name to the end of the **names** ArrayList
- (0.5 marks) a method **public void removeName(int index)** which allows the user to remove the name at position **index** from the names ArrayList

Task 2 (2 marks)

In the **NameDatabase** class, implement a method called **toString** with the signature **public String toString().** When called, this method should return (not print) a String containing each value in the **names** ArrayList on a separate line (in the same order in which they are stored).

For full marks, your toString method should use a **for-each** loop.

Task 3 (7 marks)

Please implement all your code for this task in the method labelled task3() in the **Assessment2** class of the downloaded project.

Write code which does the following:

- (1 mark) Instantiates a new **NameDatabase** object called **db**, with the **names** field set to an empty ArrayList.
- (5 marks) Adds ten **randomly generated** "**names**" to **db's** names list, where:
    - each name is between 3 and 10 characters in length;
    - each name contains only a single letter repeated - for example, aaa, bbbbb, and zzzzzzzzzz are all valid names that could be generated,
    - there is an equal chance of generating any valid name.
- (1 marks) Prints out the contents of db's names list, using the toString method defined in Task 2.

Task 4 (11 marks)

(a) (3 marks) In the **NameDatabase** class, write a method **public double getMeanNameLength()** which returns the mean value of all the word lengths in the **names** list.

(b) (8 marks) In the **NameDatabase** class, write a method **public char getMostCommonStartingLetter()** which returns the most common starting letter of all the names in the **name** list. If there is more than one most common starting letter, you should return the first one that occurs in the list.

Task 5 (7 marks)

In the NameDatabase class, write a method public int removeCommonStartNames() which:

- removes all **names** in the names list that begin with the most common starting letter, ***except*** for the first such name encountered in the list;
- prints out "Removing name ***name***" for every name that is removed, ***as soon as*** the remove operation takes place (where ***name*** should be the actual name being removed);
- prints a summary of the removal operations at the end in the form: "Removed ***x*** names beginning with ***c***";
- returns the total number of names found beginning with the most common starting letter.

For full marks, your implementation should use an Iterator object.

Task 6 (4 marks)

In the **Assessment2** class inside the method labelled task6(), write code which does the following:

- (1 mark) Instantiates a new **NameDatabase** object, and adds the following twenty names to the object's **names** list (taken from a list of the world's most popular first names):
  - `"Maria", "Nushi", "Mohammed", "Jose", "Wei", "Ahmed", "Yan", "Ali", "John", "David", "Li", "Abdul", "Anna", "Ying", "Michael", "Juan", "Mary", "Jean", "Robert", "Daniel"`
- (0.5 marks) Outputs the mean of all name lengths using the **getMeanNameLength** method written in Task 4a.

- (2 marks) Using the **removeCommonStartNames** method written in Task 5, removes names from the list until all the names left start with a different letter. (You should not remove any more names than necessary in order to achieve this outcome.)
- (0.5 marks) Outputs the remaining names, using the **toString** method written in Task 2.

Adherence to coding standards (5 marks)

All code written should adhere to the guidelines set out in the FIT1051 Coding Standards.

Submission Instructions: Please submit your IntelliJ project folder as a .zip file and submit to the Assessment 2 link on the Moodle Assessments page as shown below. If you are not sure how to zip your project, please refer to the video here. MAKE SURE YOU DOWNLOAD FROM MOODLE AFTER THAT TO CHECK IT IS THE RIGHT SUBMISSION!

Interview component: You will be asked to demonstrate your program at an interview in Week 8, following the code submission date. You will need to book an interview time with your tutor via email in Week 7.

The interview will be 10 minutes in length, and consist of 6 questions. You may be asked to explain your code, your program designs, to modify your code, to discuss your coding decisions, or to explain the any of the coding concepts taught in Weeks 1-6 that this Assessment covers.

Interviews will take place in person during Applied class time, and also online via Zoom. You must have access to a stable internet connection and a working webcam, and your webcam must be switched on for the duration of the interview. Interviews will be recorded for marking integrity purposes, and recordings will be deleted at the end of semester.

It is your responsibility to make yourself available for an interview time. The interview is worth 60% of your assessment mark, and any student who does not attend an interview will receive a fail grade for the assignment. Your interview must take place before the end of Week 8.