University of Newcastle
School of Electrical Engineering and Computing

SENG6110 Programming Assignment 1 – Trimester 1, 2020

Due: ***Electronic submission (Blackboard) by 11:59pm on Sun 22 March 2020.***
Worth: 15%

Marsupial database

Introduction
The objective of this assignment is to implement an object-oriented program using Java, to manage a database of objects containing information about marsupials.

Marsupials are native animals to Australia. You will be familiar with kangaroos and koalas, but would you recognise a potoroo, dasyurid or macropod? You are encouraged to use the common names of the nine animals used in this assignment to look for images of these animals.

This assignment must be completed individually.

Before you start
Carefully read the specification below. Make sure you have all the information necessary to start writing the program. If you are uncertain of something, do not make assumptions. Post your questions to the discussion board forum named "Assignment 1" and check it regularly.

Try to start the assignment as soon as possible. There are additional resources on Blackboard which you can use as a starting point.

Specification
The database will hold nine marsupials with the attributes listed below. See the file “marsupial_data.csv” for the complete list of data.

When run, the program will display a menu of actions to the user, including one that exits the program. Until the user chooses to exit, the menu is displayed again after each action is completed.

1. Create the marsupial objects.

Create the nine marsupial objects. This involves

a. Write a Marsupial class with the attributes listed above. You should write getter and setter methods for all attributes.

b. The class should have two constructor methods, one which will set the values of all attributes from parameters in its call, the other which expects no parameters.

c. Create the objects by using the first constructor. Hard-code the data into your program, getting it from the marsupial_data.csv file.

2. List all the data you have entered into the database. That is, all attributes for all objects.

3. Delete the third macropod animal from the database.

4. Manually add the third macropod back again. This means reading all the attribute values from the keyboard. When the user has finished, the program should display the values added.

5. Change the value of the status of the second potoroo by

   a. Read a new value from the keyboard
   b. Change the value in the database
   c. Display the values for this animal.

6. Calculate mean length and weight for the Dasyurids.

7. List common name, state and length for a given family. The family to be displayed should be read from the keyboard.

8. List names and lengths of all marsupials over a given length. The length should be read from the keyboard.

9. (Challenge) Create a Macropod class and add the three macropods to it as attributes. The attribute values for at least one of the three should be read from the keyboard (the others can be hard-coded). When all three are added, display the values of the attributes for all three by calling a method in the Macropod class.

10. (Challenge) Implement the interface using GUI.

11. Provide a UML class diagram of your program.

Program Requirements
The program should consist of three classes with the following names and instance variables (attributes). You may need other variables in these classes.

Name: Marsupial

family – the family names are *Potoroo*, *Dasyurid* and *Macropod*.
commonName – the name most people use for the animal.
scientificName – the binary Latin name unique to the species.
state – the geographic location of the animal. The value here can be "Many" for animals which are found in multiple states, otherwise WA, SA, VIC, TAS, NSW, QLD or NT.
length – The length of the animal in mm.
weight – The weight of the animal in g.
status – Is the species endangered? Values are "CR" (critical), "EN" (endangered) or "NT" (no threat).

Name: Interface

Marsupial potoroo1, potoroo2, potoroo3

    Marsupial dasyurid1, dasyurid2, dasyurid3
    Marsupial macropod1, macropod2, macropod3

Name: Macropod
    Marsupial m1, m2, m3

All instance variables of your classes should be private (this is imposed so that you apply the principles of encapsulation).

Your classes will need methods to provide the required functionalities. The only class which should have a **main method** is Interface.java, which should create an instance of the class Interface, and call a method `run()`, which will display the menu to the user. This class will be the only one that takes input from and sends output to the user. It may do so using either TIO or GUI methods (your choice). A template is shown below.

```
public class Interface {
    private void run(){
        // All your code here ...
    }

        // Create an instance of this class and run the code above
        public static void main(String[] args){
        Interface infFace = new Interface();
        intFace.run();
    }
}
```

You **must not use arrays** in this assignment.

Marks will be awarded for layout (including visual aspects (variable names, indentation) and structural aspects (variable scope, method usage)), documentation (comments), and the submission's ability to perform as specified. A more detailed marking schema will be made available on Blackboard.

What to submit

You should submit only the three .java files (Marsupial.java, Macropod.java and Interface.java) in a .zip file. Use the Assignment 1 link in the assessment tab on Blackboard. Do not include .class files in your submission.

Make sure that your name is in a comment block at the beginning of each .java file.

Late Penalty and adverse circumstances

Note that your mark will be reduced by 10% for each day (or part day) that the assignment is late. This applies equally to week and weekend days. You are entitled to apply for special consideration if adverse circumstances have had an impact on your performance in an assessment item. This includes applying for an extension of time to complete an assessment item. See https://www.newcastle.edu.au/current-students/learning/assessments-and-exams/adverse-circumstances for more details.