

Assignment 2



Koala Rescue

Introduction

This assignment is due by 10pm AEST Friday of Week 12 (12 June, 2020).

This assignment is worth 25% of the marks for your final assessment in this unit. **Heavy penalties will apply for late submission.** This is an individual assignment and must be your own work. Please note the section on plagiarism in this document.

In preparing your assignment please note the following:

- The assignment must be done using the **BlueJ environment**.
- The Java source code for the assignment must be implemented according to the **FIT9131 Java Coding Standards**.
- **You must acknowledge all code in your assignment that you have taken from other sources.**
- **Only a text interface is to be used for this program**, there is to be no GUI. More marks will be gained for a game that is easy to follow with clear information/error messages.

Any points needing clarification may be discussed with your tutor in the lab classes. You should not make any assumptions about the program without consulting your tutor.

Completion of this assignment contributes towards the following FIT9131 learning outcomes:

1. *design, construct, test and document small computer programs using Java;*
2. *interpret and demonstrate software engineering principles of maintainability, readability, and modularisation;*
3. *explain and apply the concepts of the "object-oriented" style of programming.*

Specification

For this assignment you will simulate the work of a koala rescue team. This section specifies the required functionality of the program.

Background

The koala is a marsupial and is native to Australia. Koalas typically live in open eucalyptus (gum tree) forests. Koalas live mainly in trees and eat up to **1 kg of leaves per day**. Their diet is restricted to a few varieties of gum trees (e.g., *Manna Gum*, *Swamp Gum*, *Blue Gum*, and *River*

Red Gum). Koalas use some other varieties of trees (e.g. *Wattle*) for shelter on hot days. Only one koala at a time will occupy a shelter tree. Whilst moving on the ground between trees, koalas are exposed to attacks from predators.

After bushfires in January 2020, several koala reserves were left devastated when forest habitat burnt leaving many koalas dead or injured and without sufficient food. The koala rescue team's work is to inspect the reserves and provide help to the koalas. Unfortunately, the team operates within a restricted budget and sometimes has to make difficult decisions. The aim of the rescue team is to save as many koalas as possible within a limited budget.

The koala reserve consists of a series of observation points where the rescue team pauses, observes the koala population, the trees, and the predators, and decides how the koalas can be helped. The help may be to move a koala to a safe haven if it is injured or there is not enough food or shelter.

Koala Rescue Team simulation

The Koala Rescue Team simulation begins with a welcome message and an invitation to the rescue team leader to enter his/her name. The name cannot be blank but must be less than 16 alphabetic characters. The leader is then asked to enter the budget for the rescue. This is an amount from \$100 to \$200, inclusive.

The program then sets up the numbers of trees, koalas and predators in each of the 10 observation points as follows:

1. The numbers of trees in the reserve are read from a text file **trees.txt**. The numbers of each type of tree (*Manna Gum*, *Swamp Gum*, *Blue Gum*, *River Red Gum*, *Wattle*) at each observation point are read in from the file. The file has 10 lines, with 5 comma separated numbers on each line. Each line represents the tree numbers at each observation point. There is no other reading from the file during the actual running of the program.

Each tree is either used for shelter or food. Each shelter tree can hold a maximum of one koala. The food trees can produce a certain weight of leaves per day that can be eaten by the koalas (see Table 1). The weight is used to calculate the total food available for the koalas at any observation point. The food available is calculated by multiplying the number of each tree type by the weight of leaves it produces and summing these to get the total food available. Each koala, whether healthy or injured, can eat 1 kg of leaves per day.

Table 1 Types of gum trees and weight in Kg of leaves that can be eaten by one koala per day

Type of tree	Usage	Kg of edible leaves per day
Manna Gum	Food	1.00
Swamp Gum	Food	0.34
Blue Gum	Food	0.90
River Red Gum	Food	0.40
Wattle	Shelter	0

2. There is a random number of koalas at each observation point as follows:

- 0-9 healthy koalas.
- 0-2 injured koalas.

Each koala will have a randomly allocated age of 1-18 years.

3. There is a random number of 0-4 predators at each observation point.
4. There are no koalas in safe haven at the start of the rescue.

The rescue team visits each of 10 observation points in turn. At each observation point the rescue team considers the number of **trees**, the number and condition of the **koalas** (some of which may be injured) and the number of **predators**. The team takes actions to help the koalas at each point. If the budget runs out at any point then the rescue mission continues but no actions can be taken that involve cost. The rescue mission is considered successful when all areas have been observed and all koalas have survived.

Specific observation point actions

At each observation point the following actions are performed.

1. The trees are assessed for damage. For each type of tree there is 5% chance that one tree of that type has been burnt or has fallen over. If this happens then the number of trees at the observation point is updated. Note this must happen before the available food and shelter are calculated. (Hint: to calculate a probability of 5% generate a random number from 1 to 20. There will be a 5% chance of any of these numbers being generated. So, you can nominate one number, say 20, and test for that to give you the 5%).
2. The program then displays the status of the observation point and available budget as follows:
 - the number of injured koalas
 - the number of healthy koalas
 - the weight of available food (Each koala eats 1 kg of leaves per day, whether healthy or injured).
 - the number of shelter trees
 - the number of predators
 - the available budget
3. The rescue team assesses the situation and decides what actions to take to aid the koalas' survival within the available budget. Note that if the rescue budget runs out at any stage then no further actions can be carried out that involve a cost.

The following menu options are displayed:

- A. *Move an injured koala to the safe haven* – an injured koala can be sent to the safe haven where it can be treated. If an injured koala is not taken to the safe haven then it does not survive. The cost of moving each injured koala is \$20.
- B. *Move a healthy koala to the safe haven* – if there is a shortage of food or shelter then a koala is sent to the safe haven to await possible relocation. The cost of moving each healthy koala is \$10.
- C. *Relocate a koala to this location* – a koala is relocated to this location from the safe haven. The oldest healthy koala in the safe haven is chosen each time. Note that koalas can only be relocated to a location where there is enough food, enough shelter, and fewer than three predators. For each koala relocated, \$5 is added to the rescue budget.
- D. *Take no further action* – rescue actions have finished.

After each action is taken the status of the observation point and available budget is redisplayed. The team can take as many actions as they decide.

4. When the team decides to take no further action, the survival (or not) of the koalas is determined and the numbers of koalas updated. There are several possibilities:
 - a. *Injured koalas* not taken to the safe haven do not survive.
 - b. *Shortage of food*: Each koala in excess of the number that can be sustained by the available food supply has an 80% chance of not surviving.
 - c. *Lack of shelter*: Each koala in excess of the number that can be sheltered by the available trees has a 20% chance of not surviving.
 - d. *Predators*: If there are more than three predators then there is a 50% chance of one koala in the population being killed by a predator.

Note that the numbers of koalas must be adjusted after each assessment. For example, if there is a koala who does not survive because of the food shortage then there is one less koala to consider for the shelter. Finally, if there are more than three predators then there is a 50% chance of a koala being killed by a predator.

5. When the rescue team finishes their work the following information about the observation point is displayed to the screen. The totals of:
 - koala deaths (the total of injured koalas who were not taken to the safe haven, koalas who did not survive a food shortage or lack of shelter, and koalas killed by predators, as determined from the scenarios above)
 - amount remaining in the budget

Specific actions at the completion of the observations

After all the observation points have been visited then the following summary totals are displayed on the screen:

- number of trees that have been lost.
 - healthy koalas (both in the reserve and safe haven)
 - injured koalas taken to the safe haven
 - koalas relocated
 - amount spent on the rescue
6. Finally, the last line of the summary should include the statement "Rescue was successful, with no koala deaths" or "Rescue completed with 'n' koalas deaths".
 7. The updated totals of trees are written back to the file **updatedTrees.txt**.

Class Design

Your program must consist of at least seven classes. A suggested class design is shown in Figure 1. You may want to include comments in your program to state any assumption made.

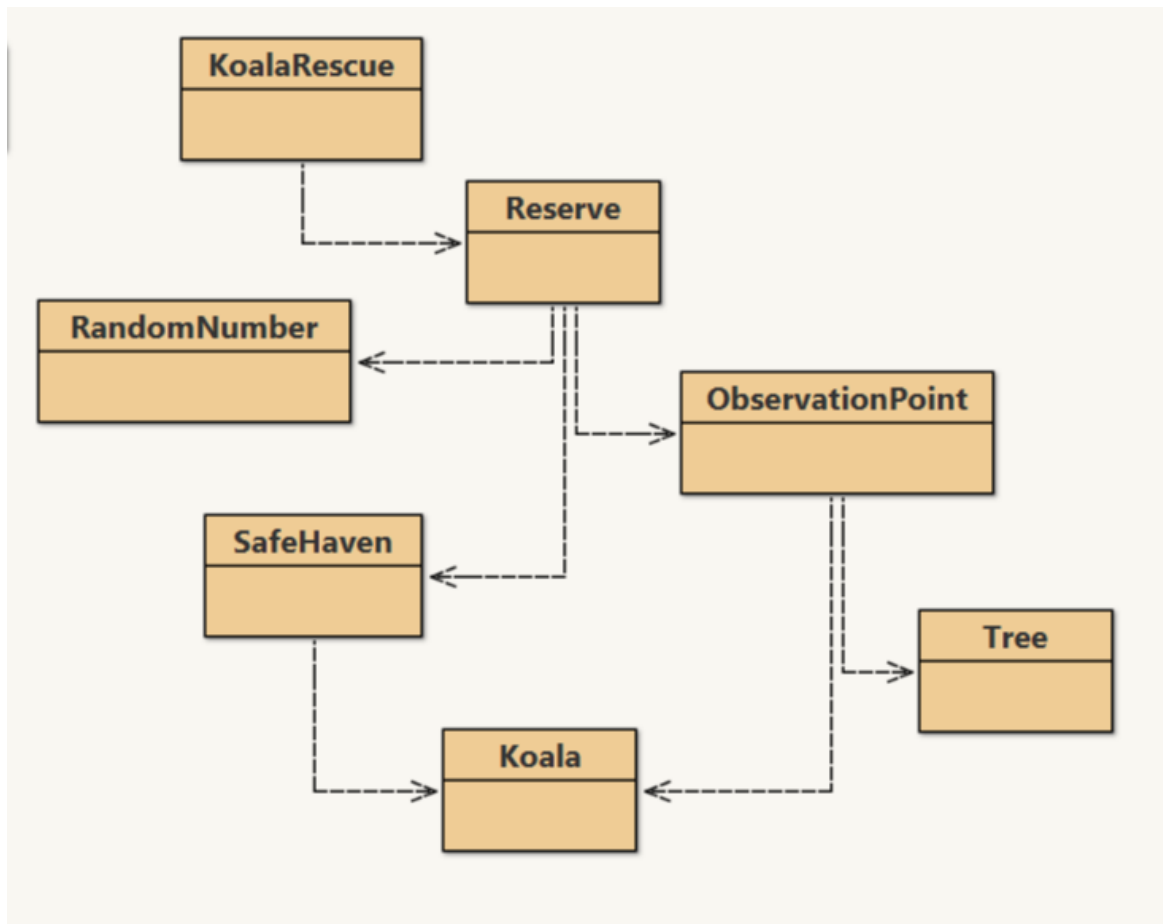


Figure 1 Suggested class design

Important Notes

1. Your program must demonstrate your understanding of the object-oriented concepts and general programming constructs presented in FIT9131. Consider carefully your choice of classes, how they interact and the fields and methods of each class. You must use appropriate data structures to store the various objects (observation points, koalas, trees, etc.) in the program. You must make use of **both Arrays and ArrayLists** in your program. Make sure that you discuss your design with your tutor. You must document any additional assumptions you made.
2. You will be required to justify your design and the choice of any data structures used at the interview.
3. Validation of values for fields and local variables should be implemented where appropriate. You should not allow an object of a class to be set to an invalid state (i.e. put some simple validations in your mutator methods).
4. Your program should handle incorrect or invalid input and present the user with relevant error messages. No invalid input should crash the program.
5. Exception handling should be used where appropriate.

Assessment

Assessment for this assignment will be done via an **interview** with your tutor. The marks will be allocated as follows:

- **10%** - Test strategy for the **Tree** class.
- **35%** - Java code and object-oriented design quality. This will be assessed on appropriate implementation of classes, fields, constructors, methods and validation of the object's state.
- **55%** - Program functionality in accordance to the requirements.

You must submit your work by the submission deadline on the due date (a **late penalty of 20% per day**, inclusive of weekends, of the possible marks will apply - up to a maximum of 100%). **There will be no extensions** - so start working on it early.

Marks will be deducted for untidy/incomplete submissions, and non-conformances to the FIT9131 Java Coding Standards.

All submitted source code must compile. Any submission that does not compile, as submitted, will receive a grade of 'N'.

Interview

You will be asked to demonstrate your program at an “interview” following the submission date. At the interview, you will be asked to explain your code/design, modify your code, and discuss your design decisions and alternatives. **Marks will not be awarded for any section of code/design/functionality that you cannot explain satisfactorily** (the marker may also delete excessive in-code comments before you are asked to explain that code).

In other words, **you will be assessed on your understanding of the code**, and not on the actual code itself.

The interviews will be organised during week 12 and will take place online via Zoom or other video facility after that time. **You must have audio and video available and operating during the interview**. It is your responsibility to make yourself available for an interview time and ensure that you have the audio and video capabilities. **Any student who does not attend an interview will receive a mark of 0 for the assignment.**

Submission Requirements

The assignment must be uploaded to the FIT9131 Moodle website by the due date.

The submission requirements for Assignment 2 are as follows:

A .zip file uploaded to the FIT9131 website containing the following components:

- the BlueJ project you created to implement your assignment.
- a document in MS Word format containing your Test Strategy for the **Tree** class. Note the JUnit facility in BlueJ is NOT to be used for this assignment.
- a completed **Assignment Cover Sheet**. This will be available for download from the unit's Moodle site before the submission deadline. You simply complete the editable sections of the document, save it, and include it in your .zip file for submission.

The .zip file should be named with your Student ID Number. For example, if your id is 12345678, then the file should be named 12345678_A2.zip. **Do not name your zip file with any other name.**

It is your responsibility to check that your ZIP file contains all the correct files, and is not corrupted, before you submit it. If your tutor cannot open your zip file, or if it does not contain the correct files, you will not be assessed.

Marks will be deducted for any of these requirements that are not complied with.

Warning: there will be no extensions to the due date. Any late submission will incur the 20% per day penalty. It is strongly suggested that you submit the assignment well before the deadline, in case there are some unexpected complications on the day (e.g. interruptions to your home internet connection).

Plagiarism

Cheating and plagiarism are viewed as serious offences. In cases where cheating has been confirmed, students have been severely penalised, from losing all marks for an assignment, to facing disciplinary action at the Faculty level. Monash has several policies in relation to these offences and it is your responsibility to acquaint yourself with these.

Plagiarism (<http://www.policy.monash.edu/policy-bank/academic/education/conduct/plagiarism-policy.html>)