

## FIT9131 Assignment B: Paw Patrol for Penguins



(image from: [Big White Dogs Save the World's Smallest Penguin in Australia \(goodnewsnetwork.org\)](https://www.goodnewsnetwork.org/big-white-dogs-save-the-worlds-smallest-penguin-in-australia/))

### Introduction

This assignment is due by 11:55pm Friday of Week 12 (**Friday 20 October, 11.55pm**). It is worth 35% 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. You must attribute the source of any part of your code which you have not written yourself. Please note the section on plagiarism in this document.

In preparing your program, please note the following:

- 

You must use the workspace environment in the Ed platform to code all parts of your program. You must not copy and paste large sections of code from somewhere else.

- 
- 

You must acknowledge all code in your assignment that you have taken from other sources.

- 
-

The Java source code for this assignment must be implemented according to the FIT9131 Java Coding Standards.

- 
- 

Only a text interface is to be used for this program. More marks will be gained for a program that is easy to follow with clear information/error messages.

- 
- 

In this assessment, you must *not* use generative artificial intelligence (AI) to generate any materials or content in relation to the assessment task.

- 

Any points needing clarification may be discussed with your tutor in your applied learning class. 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 and construct Java programs according to standard object-oriented principles

2.

3.

Apply and demonstrate debugging processes to Java applications

4.

5.

Develop strategies for efficient and effective program testing

6.

7.

Document code according to specific programming standards

8.

9.

Identify and apply the "object-oriented" concepts of encapsulation, abstraction and polymorphism

10.

11.

Explain and apply software engineering principles of maintainability, readability and modularisation

12.

## Specification

For this assignment you will write a program to evaluate the effectiveness of employing specially trained dogs to protect the fairy penguins in the penguin reserve on Phillip Island. This section specifies the required functionality of the program.

## Background

On Phillip Island, Victoria, there is a colony of fairy penguins that inhabit the shoreline of Summerland peninsula. The penguin is known to the indigenous Boonwurrung people as Djinan Yawa-dji Goyeep.

Penguins build nests in the grasslands by the shoreline where they lay eggs and raise their chicks. The breeding season is between August and February and during this time the female penguin will lay from zero to two eggs each month. During the day the penguins fish in the ocean, returning at sunset to feed their chicks. The return of the penguins from the ocean is known as the "penguin parade".

Unfortunately, the penguin colony is vulnerable to predators and has been under threat in recent years, particularly the baby penguins left in their nests while their parents are out fishing. The main predators on land are foxes and cats, and in the sea the main predators are sharks.

The Phillip Island rangers who look after the penguin colony have an idea that they could use specially trained dogs to help protect the colony. The dogs can chase predators away and sometimes kill them. The use of such dogs has been successful in protecting other penguin colonies. A local Paw Patrol group can train dogs for this purpose; however, the dogs are expensive to train and maintain. To understand more about the effectiveness of using dogs to protect penguins the rangers would like some more information.

In this assignment you will write a computer program to simulate the penguin colony over the course of a year with different levels of Paw Patrol dog protection.

The simulation will start with an established colony and update the numbers once a month. At the end of the year the program will report the changes to the colony.

## Paw Patrol simulation

The Paw Patrol for Penguins program will simulate one year of a colony of penguins. The simulation steps through each month of the year, beginning in July and ending in June, 12 months later. The user chooses whether to run the simulation with no Paw Patrol dog protection, with one Paw Patrol dog or with two Paw Patrol dogs. The colony will start with the same number of penguin families for each simulation. At the end of a simulation the penguin colony survival statistics are reported.

### Program start up

The program begins by displaying a brief welcome message.

The user is prompted for the number of dogs they would like from Paw Patrol to protect the penguins. The user can choose no dogs, one dog or two dogs.

The numbers of penguin families, foxes, cats and sharks are read in from a file *colony.txt*. These will be four comma separated integers. There is no other reading from the file during the actual running of the program.

The program creates collections of penguin families, foxes, cats and sharks. The details of these are as follows:

1. Each penguin family will have:

- 

a unique identification code in the following format **Pnnn** (where nnn represents a sequence of 3 digits

- 

- 

two penguins (one male and one female). Each penguin will have flag indicating whether it is a male or female and a status indicating whether it is alive or not.

- 

- 

a number of eggs. Each egg will have an age (in months) and a status indicating whether it is alive (i.e., has not been eaten) or not

- 

- 

a number of chicks. Each chick will have an age (in months) and a status indicating whether it is alive or not.

- 

2. At the start of the simulation, each penguin family will have a live male and a live female penguin, no eggs and no chicks.

3. Each fox will have separate counts of the penguins, chicks and eggs it has killed and a status indicating whether it is alive or not.

4. Each cat will have separate counts of penguins and chicks it has killed and a status indicating whether it is alive or not.

5. Each shark will have a count of the penguins it has killed.

**Specific actions each month**

Each month the following events can occur:

- 

In the months from August to February a female penguin from a family group can lay 0, 1, or 2 eggs, with each number of eggs have equal probability.

- 

- 

The ages of existing chicks and eggs is increased by 1 month.

- 

- 

Eggs that were laid in the previous month (i.e., are one month old) will be due to hatch. The probability of a successful hatching is 0.70. (Hint: to calculate the probability of this event, generate a random number from 1 to 10. There is a 10% chance of each of these numbers being generated so you can nominate numbers 1-7 for a successful hatching). If a hatching is successful, the number of chicks in a family group is increased by one and the number of eggs is decreased. If a hatching is unsuccessful then the number of chicks remains the same and the number of eggs is reduced by one.

- 

- 

The penguins, chicks and eggs are vulnerable to predators. Foxes can eat penguins, chicks and eggs. Cats can eat penguins and chicks, and sharks can eat penguins. The probability of each of these events each month is as follows:

- 

○

The probability of each penguin, chick or egg being killed or eaten by a fox is 0.08. With Paw Patrol protection with one dog the probability of each event is 0.02 and with two dogs is 0.008.

○

○

The probability of each penguin or chick being killed or eaten by a cat is 0.04. With Paw Patrol protection with one dog the probability of each event is 0.01 and with two dogs is 0.004.

○

○

The probability of each penguin being killed by a shark is 0.02. Unfortunately, the Paw Patrol dogs cannot help penguins in the sea.

○

●

If a penguin is eaten then its alive status changes to false. If the other penguin in the family group is alive then they continue looking after any eggs and chicks, but no more eggs are laid for the rest of the season. If the other penguin in the family group is not alive then unfortunately the eggs and chicks do not survive. In this case the numbers of eggs and chicks in the family group are set to 0.

●

●

If a chick is eaten, then its alive status changes to false and the number of chicks in the family group is reduced by 1.

●

●

If an egg is eaten, then its alive status changes to false and the number of eggs in the family group is reduced by 1.

●

●

The cats and foxes can be killed by the Paw Patrol dogs. The probability of each of these events each month is as follows:

●

○

the probability of each cat or fox being killed is 0.01, if there is one Paw Patrol dog.

○

○

the probability of each cat or fox being killed is 0.1, if there are two Paw Patrol dogs.

○

After these events a summary of the status of the colony is displayed.

●

Number of complete family groups

- 
- 

Number of live chicks

- 
- 

Number of live (uneaten) eggs

- 

### **Specific actions at the completion of the simulation**

At the end of the simulation a summary is displayed.

- 

Number of complete family groups.

- 
- 

Number of live penguins

- 
- 

Number of live chicks

- 

*Family group survival rate:*

$(\text{total\_penguin\_families\_with\_two\_parents} / \text{total\_penguin\_families\_at\_the\_start}) * 100$

*Penguin survival rate:*

$(\text{total\_penguin\_alive} / \text{total\_penguins\_at\_the\_start}) * 100$

*Egg survival rate:*

$(\text{total\_eggs\_hatched} / \text{total\_eggs\_laid}) * 100$

*Chick survival rate:*

$(\text{total\_chicks\_alive} / \text{total\_chicks\_hatched}) * 100$

*Overall colony survival:*

$(\text{total\_penguins\_alive} + \text{total\_chicks\_alive}) / \text{total\_penguins\_at\_the\_start}$

(note that an overall colony survival < 1 means that the colony will decline and an overall colony survival > 1 means that the colony will grow)

A summary is written to the file *colonyFinal.txt*. The details written to the file will be the four survival rates and overall survival.

## Program and Class Design

The design of the program will be discussed in your Applied Class in Week 9. It is important that you attend this class.

## 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 (penguins) 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.

3.

You will be required to justify your design and the choice of any data structures used at the interview.

4.

5.

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).

6.

7.

Your program should handle incorrect or invalid input and present the user with relevant error messages. No invalid input should crash the program.

8.

9.

Exception handling should be used where appropriate.



10.

## Assessment

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

- 

10% - Progress of test strategy and code development, as shown via Ed workspace environment. Your tutor will assess your work during your applied session in weeks 10 and 11.

- 

- 

5% in week 10 for the **PenguinFamily** class and its **test strategy**

- 

- 

5% in week 11 for a draft of the **class diagram** and two further classes, which must be a **FileIO** class and another class which is a *client* class (please see Lesson 8.1.2 for an explanation of client class). Note that the class diagram should show the individual classes and the interactions between the classes but does not need to include the details within the classes.

- 

- 

10% - Test strategy for the **PenguinFamily** class.

- 

- 

10% - Class diagram, Java code quality and object-oriented design quality. This will be assessed on code quality (e.g., compliance with coding standards) appropriate design and implementation of classes, fields, constructors, methods, and validation of the object's state.

- 

- 

10% - Program functionality in accordance to the requirements.

- 

-

60% - Oral assessment.

- 

Note that it is not a requirement that you use polymorphism and inheritance concepts in this assignment, but incorporating these concepts will enhance your chances of achieving a high distinction (HD) grade.

A reminder that you must use the workspace environment in the Ed platform (opposite this assignment specification) to code all parts of your program. You must not copy and paste large sections of code from other sources, and you must acknowledge *any* code in your assignment that has been taken from other sources.

Marks will be deducted for untidy/incomplete submissions.

You must submit your work by the submission deadline on the due date (a late penalty of 10% per day, inclusive of weekends, of the possible marks will apply). There will be no extensions - so start working on it early.

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

## Oral assessment

As part of the assessment you will attend an interview following the submission date. At the interview, you will be asked questions about your code. You will be asked to explain your code/design, modify your code, and discuss your design decisions and alternatives. Marks will be awarded for your answers (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.

Interview times will be arranged in the applied classes in Week 12 and will take place on campus during the following week.

It is your responsibility to make yourself available for an interview time. **Any student who does not attend an interview will not receive a pass grade for the assignment.**

## Submission Requirements

The assignment must be submitted by **11:55pm Friday of Week 12 (20 October 2023)**.

The submission requirements for Assignment B are as follows:

- 

The main class in your program **MUST** be called **PawPatrol.java** and it should contain the **main()** method to start the program.

- 

-

Class diagram submitted as a pdf file.

- 
- 

Test strategy for **PenguinFamily** class submitted as a pdf file.

- 
- 

Submit all your work (coding, class diagram and test strategy) via the Ed platform.

- 
- 

Re-submissions are allowed (and encouraged) before the submission deadline. Please ensure however that you do not click on the submit button *after* the due date. Your final submission will be used for grading purposes, and any submission made after the deadline will incur a late penalty.

- 
- 

A signed Assignment Cover Sheet. [Note: You are required to download the [Assignment Coversheet](#), sign the document and upload the pdf file in the Ed platform (you may drag and drop to the Toggle Pane)]

- 

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 a 10% 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).