UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Department of Computer Science
COS132 - Imperative Programming
Practical 6

1 Introduction

Deadline: 3rd June, 20:00

1.1 Objectives and Outcomes

The objective of this practical is to practice using pointers and classes.

1.2 Submission

You will have **10 submissions** where the total for the practical is **5 marks**. Submit your code to Fitchfork before the deadline. Students are **strongly advised** to submit well before the deadline as **no late submissions will be accepted**.

1.3 Plagiarism

Copying will not be tolerated in this course. For a formal definition of plagiarism, the student is referred to the COS132 Study guide. If you have questions regarding this, please ask one of the lecturers, to avoid any misunderstanding.

2 Practical Requirements

2.1 Dog Parlor

You are required to write a program that will simulate a dog parlor where clients give you the details of their animal and you record them in an object. You will be given the following files named main, animal and citizen.

The animal class must store the following information:

- Name
- Age
- Owner

To store the created animal object pointers you will use an array of animals (no imports are allowed). This array will never exceed an index of 10. Each animal object will contain a owner object, if the owner is set to null the following should be displayed "hoping for adoption".

The user will always have three options, the first is to add a new dog, the second to swap the names and the last option to stop the program.

Sometimes two dogs' names get swapped. You will need to implement a function named **swapNames** that will swap the names by reference. This function will be inside of the main file.

2.2 Example Program

The following is an example of the required program:

```
Enter 1 to add a dog, 2 to swap names or 3 to stop: 1
Please enter the name of the dog: Jake
Please enter the age: 12
Does the dog have an owner: Yes
Please enter the owners name: Steve
Please enter the address: Brooklyn
The list of dogs are as follows: Jake(Owner: Steve)
Enter 1 to add a dog, 2 to swap names or 3 to stop: 1
Please enter the name of the dog: John
Please enter the age: 4
Does the dog have an owner: No
The list of dogs are as follows: Jake(Owner: Steve), John(Owner: hoping for adoption)
Enter 1 to add a dog, 2 to swap names or 3 to stop: 1
Please enter the name of the dog: Lemon
Please enter the age: 9
Does the dog have an owner: Yes
Please enter the owners name: Craig
Please enter the address: Brooklyn
The list of dogs are as follows: Jake(Owner: Steve), John(Owner: hoping for adoption)
,Lemon(Owner: Craig)
Enter 1 to add a dog, 2 to swap names or 3 to stop: 2
Enter name of first dog: Jake
Enter name of second dog: Lemon
The list of dogs are as follows: Lemon(Owner: Steve),
John(Owner: hoping for adoption), Jake(Owner: Craig)
Enter 1 to add a dog, 2 to swap names or 3 to stop: 3
```

2.3 Extra Comments

If asked to swap names there will always be at least two dogs stored. If any invalid names are asked to swap you can output: "No dog found".

As with the previous practical you will not receive your marks until after the deadline. You are also not allowed to use any imports unless provided.

2.4 UML Diagram

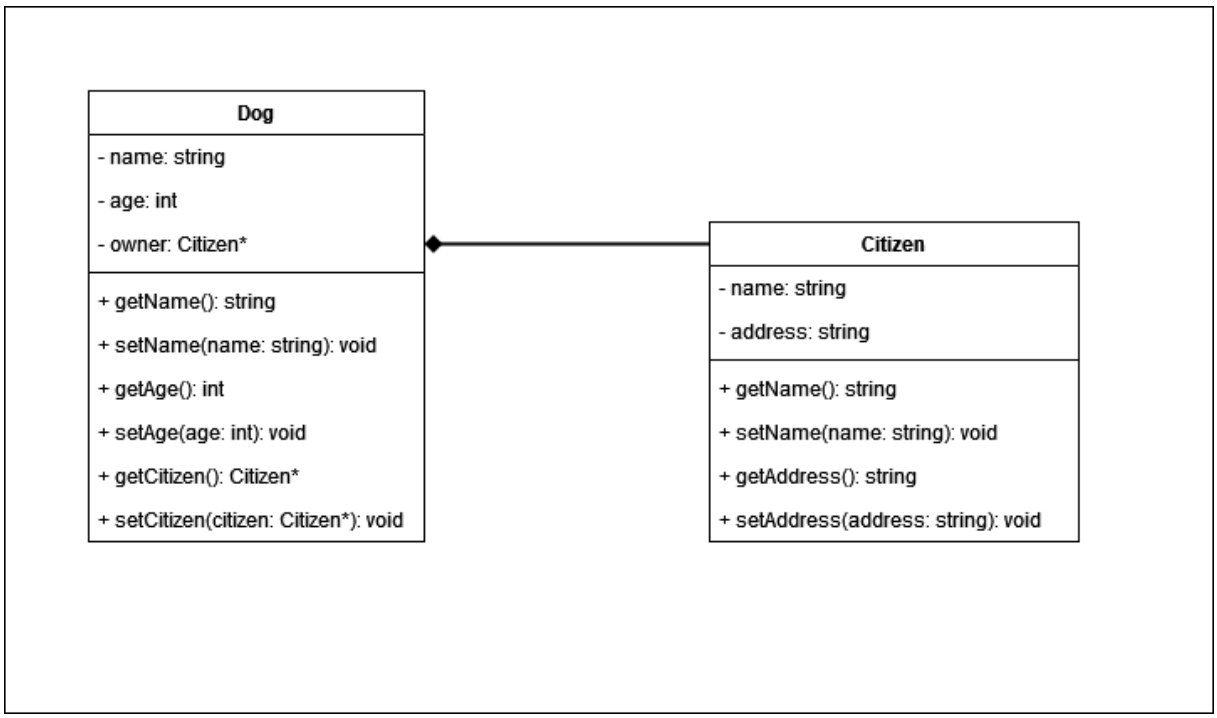

Figure 1: UML Diagram of dog parlor program