

CONTINUOUS ASSESSMENT / ASSIGNMENT

Programme Title/Year:	HDIP in Computing (Feb 2019 PT)
Module Title(s):	Algorithms & Constructs
Lecturer Name(s)	Amilcar Aponte
Assessment Title:	Vet Clinic
Assessment Type:	INDIVIDUAL OR PAIRS (i.e. Groups of 2)
Assessment Weighting:	Algorithms & Constructs – 50%
Date Issued:	10 th March 2020
Due Date (Deadline):	Saturday, 16 th May 2020 at 23:59
Late Submission Penalty:	<p>Late submissions will be accepted up to 5 days after the deadline. All late submissions are subject to a penalty of 10% <u>of the mark awarded</u>.</p> <p>Submissions received more than 5 days after the deadline above <u>will not</u> be accepted.</p>
Method of Submission:	Moodle
Feedback Method:	Results and feedback sheet posted through Moodle
Instructions for Submission:	<p>Full Eclipse/NetBeans project containing all source code, submitted as a compressed zip file. In your zip file, include a txt file with the link to your private git repository.</p> <p>The lecturer must be a collaborator to the repository in order to be able to access it.</p>

ALGORITHMS AND CONSTRUCTS

- Develop and implement custom algorithms and constructs for a wide variety of application-specific tasks, taking performance and system resources into consideration.
- Employ a variety of different constructs and to implement and categorise different searching and traversal algorithms over said constructs

Assignment Detail

You have been tasked to create a working program to demonstrate a “Vet Clinic” as a prototype

The Vet Clinic must include the following entities:

- Admin Staff (e.g. Receptionist; IT Nerd)
- Medical Staff (e.g. Veterinarian; Nurse; Trainee Vet)
- Animals (e.g. Dog, Cat, Rabbit)

When the program is executed, it must dynamically create as follows:

- At least 10 Admin staff of various types (they cannot all be the same type). There must be at least 2 different types of Admin staff.
- At least 30 Medical staff of various types (there must be 5 Vets, and as above, the 30 cannot all be the same type). There must be at least 3 different types of Medical staff.
- At least 1000 animals. There must be at least 3 different types of Animal. However, you can include additional Animal types for extra marks

Each staff member must be given (a) a name, (b) a UNIQUE staff number and (c) a salary level. The Staff name must be randomly generated in some manner (but must be text and should look like a name!)

Each animal must be given (a) a name, (b) an age (realistic age!) and (c) a medical condition. The medical condition does not need to be realistic for the animal in question, but should look like an illness (see example below). Animal names must be generated randomly in some manner (but must be text and should look like a name)

Once all the entities have been created, the system should assign EACH animal to a member of medical staff (for treatment). This means you need some way of knowing which Medical Staff member is assigned to which animal.

As any member of the medical staff will be assigned to treat more than one animal, the order in which they are going to be looked after is something that must be considered. Once an animal has been looked after, the doctor needs to know who is the next one.

The system must also assign a task to each member of the Admin staff. These tasks can be invented by you, but there should be several different tasks – e.g. Filing, Making Phone Calls, On Holidays, etc.

The user should be able to:

- List all staff
- List staff by categories (e.g. list all Nurses)
- List all Admin staff performing a certain task
- Search for a specific member of staff by name
- List all animals
- List animals by various types (e.g. list all dogs)
- Search for a specific animal by name
- List all the animals assigned to a member of medical staff
- List the order in which pets will be looked after by a particular member of the medical staff
- For a given member of the medical staff, pass to the next pet.

In order to perform these tasks, you should implement APPROPRIATE constructs, data structures as well as searching and/or sorting algorithms.

Specific Restriction

This is a command line program, and the use of graphic user interfaces is not allowed (i.e. JavaSwing, JavaFX, GUI builders, etc.).

Marking Schedule

Description	Weighting
<p>Appropriate constructs / Data structures have been used to store the objects in memory.</p> <ul style="list-style-type: none">- Staff / Animals- Tasks- Sequence of pet to be seen by a member of the medical staff <p>The structures allow to list all elements contained in it.</p> <p>Code is well structured and commented.</p>	35
<p>Appropriate filtering of elements from the structures to display by groups.</p> <ul style="list-style-type: none">- Staff by category- Animals by type <p>Code is well structured and commented.</p>	20

<p>Appropriate searching algorithm have been utilised within the program to find specific elements in the structure:</p> <ul style="list-style-type: none"> - Specific member of staff by name - Specific animal by name <p>This is working correctly. Code is well structured and commented.</p>	20
<p>Appropriate display of other groups/categories</p> <ul style="list-style-type: none"> - Admin staff performing a given task - Animals assigned to a doctor <p>This is working correctly. Code is well structured and commented.</p>	20
<p>The user can remove the first pet from the queue for a specific doctor when this has been looked after. Code is well structured and commented.</p>	5
Total (See note re: individual contribution)	100%

ASSESSMENT OF INDIVIDUAL CONTRIBUTION (where assignment is completed in Pairs)

Each group member inherits the overall group project mark. This mark is then adjusted by their individual contribution.

Individual contribution will be assessed by reference to your Git repository.

For example:

- Final group result: 76%
- Student with perfect contribution receives: $100\% * 76\% = 76\%$
- Student with just sufficient contribution receives: $50\% * 76\% = 38\%$

It is important therefore that (if you are completing this assignment as a pair) both of you need to ensure that you are contributing to the coding work!

ANIMAL EXAMPLE (Displayed on screen)

Type of Animal: Dog
 Name of animal: Rover
 Age: 3
 Medical Condition: Rickety Bones

Type of Animal: Rabbit
 Name of Animal: Fluffy
 Age: 5
 Medical Condition: Carrot Allergy