**CA3 Programming Fundamentals 1**

|  |  |
| --- | --- |
| Vet Application | |
| You are required to develop a Vet menu system  This will involve the development of three new classes:   * Animal * Vet * VetApplication   You will use the EasyScanner for user input. |  |

|  |  |
| --- | --- |
|  | The **Animal** class.  This is a template / blueprint class for an Animal. It stores data on the id, animal name, age, animal type, owners name, vaccination (assume they are false until changed), year they registered the animal and length of time with vets (assume it starts at 0 until changed). The constructor updates **some** of these instance variables (attributes) with the information passed as a parameter.  You need an accessor and mutator methods for each attribute. |

|  |  |
| --- | --- |
|  | The **Vet** class.  This class creates an array of animals and handles it. You will need to maintain a total attribute that will hold the number of animals stored in the system (2 attributes).  Methods include:   * 1. A constructor that will set the size of the array and set the total number of animals to 0.   2. to return the total number of animals in the system   3. add an animal to the system (to the array)   4. delete an animal from the system   5. check if the vet system is empty   6. check if the vet system is full   7. find the oldest animal in the system   8. find the youngest animal in the system   9. search the system for an animal   10. show details of all of the animals stored in the system   11. show details of a particular animal   12. to update **ALL** the animals’ length of time in the vets.   13. add details to the animal already in the system   Please take note of the parameter list and return type from the UML diagram on the left for each method. |

|  |  |  |
| --- | --- | --- |
|  | **Vet** **Application**  This class is the application class and contains the main method. This class displays the menu repeatedly to the user until they enter 8. Note that there is a private method behind each option on the menu. i.e. if the user enters 1 it will call another method to execute option 1, if the user enters 2 it will call another method to execute option 2 etc. |  |
| When the user selects option 6, a sub-menu is displayed to the user.  Please see the runtime version for the operation of these menus.  **You must ensure error checking every time the user enters data (where applicable i.e. age, year registered etc).** |  |

**Important Points**

1. This is an individual project.
2. Any sign of cheating/copying will result in a **ZERO GRADE**.
3. By uploading your assignment, you are electronically signing the WIT anti-plagiarism declaration. Please see the WIT website for more details on this policy.
4. If the runtime version is deemed to be **decompiled** this will also result in a **ZERO GRADE.**
5. Code **MUST** be commented.
6. Your program **MUST** run as sample program runs.
7. You will be interviewed on your programs when they are submitted to determine authorship and understanding.
8. Make a copy of your assignment for interview purposes and remove **ALL** comments prior to interview. You will only be interviewed on un-commented code. This will also be uploaded to Moodle.
9. You must demo your code or it will result in a **ZERO GRADE.**
10. You will receive a grade for your interview. This is a multiplier for your assignment grade.
11. You must submit the fully commented project via moodle (zip and upload, folder called YourName).
12. You must ensure you have uploaded the correct file as after the deadline no submission will be accepted.
13. **You will be asked to demo your code on XXXX of May 2021 via Zoom between 9am – 5pm (To be confirmed)**
14. **Please submit your project by XXX of May 2021 by 12pm. (To be confirmed)**