***Conestoga College Skills Competition***

***Coding***

***Feb 13, 2023***

# **Welcome to the Competition!**

# The purpose of this contest is to evaluate your understanding and ability in solving a problem using the software as well as displaying coding skills. Different projects have been developed that will require you to build flowcharts and create programs.

***Make sure you read all of this competition before starting.***

**Competition Description**

Welcome to the 2023 Conestoga Coding Contest. As a competitor, you should be prepared to use your own computer with an internet connection operating with Windows or Mac. Please use programming language: C#.

Make sure that your code is openable in Visual Studio.

Coding standards, such as proper use of comments and spacing, will be marked. **All files should be submitted in a single .zip file. The final submission file should be named FirstName\_LastName.zip. That file needs to be uploaded to IT Programs, here is the link: https://conestoga.desire2learn.com/d2l/lms/dropbox/admin/mark/folder\_submissions\_users.d2l?db=715069&ou=63593**

**Good Luck!**

# **Code Review and Deployment**

**Please read this section carefully:**

* **You will use one of the following programming languages: C#.**
* **Make sure ensure that your code is openable in Visual Studio.**
* **There should be no communication between candidates.**
* **Documentation is important within your code.**

# **Save the files as YourName\_2023.**

# **Code Review and Deployment**

**You will be creating an application for helping an Animal rescue not-for-profit organization manage incoming animals. Your submission should also include the source code, documentation, and deployment files or installations script.**

# Requirements for Completion:

**The first step in managing the animal rescue is creating a flat file with all the animals in the shelter now. This task will need all the basic CRUD functions (create, read, update, delete).**

**The file must include the following:**

* **ID: A generated incrementing 0 padded 8-digit number.**
* **Species: Dog, Cat, Bird, Rabbit, Small & Furry, Fish, Barnyard, Other**
* **Name: The animal’s name.**
* **Gender: F or M**
* **Spayed: Yes or No**
* **Breed: Collie, Beagle, Siamese, Calico, unknown, etc.**
* **Colour: Brown, Tabby, White, etc.**
* **Birthday: Date of the estimated animal’s birth. Format dd/mm/yyyy.**
* **Vaccine Status: Up to date, late, unknown.**
* **Identification: Bar code, Micro-chipped. If yes, what is the number**
* **Adoption fee: < $300.**

**The Application itself can function in one of two ways, either by displaying a user interface or by accepting command line arguments.**

**Your application must have the following functionalities:**

* **Add Animal.**
* **The breed should be valid**
* **Remove animal by ID.**
* **Search for an animal by name or species.**
* **Display animals sorted by species.**
* **Display the three oldest animals for each species.**
* **Usage instructions or a help option**

# Deployment (Video Presentation)

**Generate a training video so a user would know how to run your application, make sure you point out:**

* **your code**
* **anything you feel that is unique.**

**Your application should be able to calculate the adoption fee for each animal based on the criteria below:**

* **Kittens, puppies, and other young animals (below one year of age): $300**
* **Senior animals (above ten years of age): $100**
* **All other animals: $200**

# Marks

**Marks will be given for:**

* **Creativity and innovation.**
* **Including a “Docker file” and instructions on how to run it.**

# Submission requirements

# **Upload to the dropbox a complete zip file of your project and the mp4 video file. IF the files cannot be open no assessment.**

# **Make sure your Application runs and does not crash!**

# **A penalty will be given to any applications that crashes!**

# Judging Criteria:

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Grade Range | | |
| Elements Included | All required elements were included. **(15-25)** | Most of the required elements were included. **(5-15)** | Missing more than 2 of the required elements. **(0-5)** |
| Program Content and Functionality | The program was easy to understand and use. The content was clearly displayed. Data and functions worked as intended. **(30-40)** | The program could be clearer to understand and use. Most functions worked properly. **(20-30)** | The program was not functional and/or crashed. Most functions were missing. **(0-20)** |
| Code Quality | The code was well organized and easy to read. Good use of comments. **(7-10)** | The code could be better organized and easier to read. Missing some comments. **(4-7)** | The code was disorganized and hard to read. No comments were used. **(0-4)** |