

2022 VIRTUAL SKILLS ONTARIO CONTESTS  
les Olympiades virtuelles 2022 de Compétences Ontario

CODING / PROGRAMMATION  
 TI APPLICATION LOGICIELLE

Thursday, April 28, 2022



# **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.

**Competition Description**

Welcome to the 2022 Virtual Skills Ontario Coding Contest. As a competitor, you should be prepared to use your own computer with an internet connection operating with Windows or Mac. You are allowed to use the following programming languages: C#, C++, C, and JAVA.

If you are using C#, C++, or C, you should ensure that your code is openable in Visual Studio.

JAVA projects are required to use and submit a **Maven pom.xml** file.

Note that only standard libraries are allowed: Java SE 8, .NET 4.7+, and Microsoft C/C++ runtime.

You must submit an executable .exe for all C language projects or an executable .jar for projects completed with JAVA.

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.**

**Good Luck!**

# 

# **Part A – Knowledge and Problem Solving**

# Please read this section carefully:

# You will submit your answers as a Microsoft Word, Apple Pages, PDF, or image file.

# The section is valued at 25%.

# There should be no communication between candidates.

# There are no resources allowed except online help.

# **Part A – Knowledge and Problem Solving**

1. Create a flowchart walking through the process of purchasing online, installing, and using a new piece of software or video game. Solution should have at least 5 processes.
2. Explain the difference between a frontend and a backend.
3. Explain what happens when you instantiate a new instance of a class in as much detail as you can. For example, ‘new AnimalAdoptionAgency(logger)’.

# The flowchart and answers can be submitted as a Microsoft Word, Apple Pages, PDF, or image file. Hand-written answers may also be submitted as long as they are legible. Neatness and attention to detail count!

# Requirements for Completion:

# Save the file for Part A as **YourName\_PartA**.

# Include required files in your submission .zip archive named FirstName\_LastName.zip.

# Part A – Judging Criteria:

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Grade Range | | |
| Knowledge Questions | All questions were correctly answered in a detailed manner. **(8-10)** | Most of the questions were answered correctly; solutions could be more detailed. **(4-7)** | Questions were not answered correctly. No attention was given to a detailed answer. **(0-3)** |
| Flowchart | The flowchart was detailed and contained all the necessary steps for solving the problem. The flowchart was professional, clean, and readable. Proper shapes and symbols were used correctly. **(0-15)** | | |

# 

# **Part B – Code Review and Deployment**

# Please read this section carefully:

# You will use one of the following programming languages: C#, C++, C, or JAVA.

# If you are using C#, C++, or C, you should ensure that your code is openable in Visual Studio.

# JAVA projects are required to use and submit a **Maven pom.xml** file.

# Only standard libraries are allowed: Java SE 8, .NET 4.7+, and Microsoft C/C++ runtime.

# Submit an executable .exe for C language projects or an executable .jar for projects completed with JAVA.

# The section is valued at 75%.

# There should be no communication between candidates.

# There are no resources allowed except online help.

# Save the files you create with the names provided.

# **Part B – Code Review and Deployment**

You will be creating an application for helping an Animal rescue not-for-profit organization manage incoming animals. This Application can be written in either C#, C++, C, or Java and submitted as a .exe or .jar file. 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 (No Presentation)

* Create a README.md file that contains how to use your application.
* Create multiple printscreens of each process when running the application such as Add, Delete, and Edit.
* Document your printscreens above.

# Post-Secondary Only

Above are the mandatory requirements for the Application for all contestants. The following is a list of required additions for post-secondary contestants only:

* In addition to the basic CRUD functions (create, read, update, delete), your application should be able to **archive** and **restore** the information of animals that have been adopted. Add any fields necessary.
  + An archived animal will not show up in the regular search
  + Include an option to search for archived animals within a timeframe
  + Include an option to archive all animals adopted at least three months ago
* 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

# Bonus (Optional) Marks

Bonus marks will be given for:

* Creativity and innovation.
* Including a “Docker file” and instructions on how to run it.
* Create a new addition to the existing program, except related to adopting an animal. For example, include the person interested in adopting’s name, contact information, the animal they want to adopt, why they are a great fit for the animal, etc.

# Submission requirements

# Save the files for Part B in a folder named **YourName\_PartB.**

* Make sure all your printscreens are included.

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)** |