42nd ONTARIO ENGINEERING COMPETITION

Competition Package

PROGRAMMING

Table of Contents

1.0 Introduction

The goal of the Programming category is to encourage engineering students to produce a piece of industry-quality software with all the proper user and administrative documents. The teams will use their software development skills, their technical writing abilities, and their project management skills to design a solution to a problem posed. This solution will then be presented to the judging panel.

1.1 Competition Leads

Kaveen Kumarasinghe

oecprogramming2021@gmail.com

+1 (647) 702-1734

Trent Stauffer

oecprogramming2021@gmail.com

+1 (519) 277-3287

1.2 General Rules and Guidelines

1) Any competition questions should be asked publicly on Discord.
2) External online visitors are NOT allowed at any time during the competition. Only competition volunteers and judges are allowed.
3) If using an API, the API must be free or have a trial length
4) You may use any language you'd like!
5) Do not wear anything with your school name or logo if participating in video chat while presenting online
6) You must not disclose your school or any hints that give away your school at any time during the competition.

2.0 Competition Schedules

2.1 Schedule of Saturday, January 23rd, 2021

Timings	*Task*	*Location*
10:00 am – 10:30 am	Registration & Check-In	Hopin
10:30 am – 11:00 am	Competition Briefs and Intro	Hopin
11:00 am – 6:30 pm	Build Time / Work Period	Discord
6:30 pm – 7:00 pm	Final Submission Period	GitHub
8:00 pm – 10:30 pm	Night Social Event #1	Discord

2.2 Schedule of Sunday, January 24th, 2021

Timings	*Task*	*Location*
8:00 am – 8:30 am	Team 1 Judging	Zoom
8:30 am – 8:40 am	Transition	
8:40 am – 9:10 am	Team 2 Judging	Zoom
9:10 am – 9:20 am	Transition	
9:20 am – 9:50 am	Team 3 Judging	Zoom
9:50 am – 10:00 am	Transition	
10:00 am – 10:30 am	Team 4 Judging	Zoom
10:30 am – 10:40 am	Transition	
10:40 am – 11:10 am	Team 5 Judging	Zoom
11:10 am – 11:20 am	Transition	
11:20 am – 11:50 am	Team 6 Judging	Zoom
11:50 am – 12:00 pm	Transition	
12:00 pm – 12:30 pm	Team 7 Judging	Zoom
12:30 pm – 1:30 pm	**Lunch**	
1:30pm - 2:00pm	Team 8 Judging	Zoom
2:00pm - 2:10pm	Transition	
2:10pm - 2:40pm	Team 9 Judging	Zoom
2:40pm - 2:50pm	Transition	
2:50pm - 3:20pm	Team 10 Judging	Zoom
3:20pm - 3:30pm	Transition	
3:30pm - 4:00pm	Team 11 Judging	Zoom
4:00pm - 4:10pm	Transition	
4:10pm - 4:40pm	Team 12 Judging	Zoom
4:40pm - 5:00pm(approx)	Deliberation	

3.0 Programming Competition Description

3.1 Problem Background

A zombie apocalypse has started in Ontario! A few days ago, people started exhibiting symptoms of a trance-like state, erratic behaviour, and society is about to turn into a state of unrest. The CDC has identified and named the virus responsible for the zombie outbreak, **ZBY1.** Before the situation gets any worse, we're tasked with identifying the spread of the apocalypse. Since we can't identify the spread effectively in our entire province, we will choose to focus on one aspect of our society: schools. Children are always moving between classes, mingling in hallways, meeting and collaborating with peers, and interacting with their teachers and assistants. If a child gets infected with **ZBY1** and goes to school it is safe to assume that many others will be affected by it, but it is difficult to gauge exactly how much, and who specifically. However, schools usually have attendance logs and student records that show where they are supposed to be at specific times, and this can ultimately help us identify who would be affected if there were to be a **ZBY1** outbreak.

Unlike previous competitions, the challenge this year will be much more open ended. This approach has been taken this year because in light of recent events, it can be seen that creativity and outside-the-box thinking is now more important than ever.

3.2 The Challenge

The premise of this challenge takes place in a high school, which has properties as follows:

- The high school teaches students from grades 9 to 12.
- The high school has 4 total class periods.
- There is a mandated lunch break after the first 2 class periods, where all the students are released for lunch. During this lunch period, all the classrooms are cleaned.
- There are 10 subjects taught at the school: physics, biology, functions, calculus, philosophy, art, drama, computer science, computer engineering, and humanities. Each subject has two sections A and B, which are taught by different teachers.
- Students can have spare periods where they do not have any class scheduled and are free to do anything of their choosing.
- There are 2 teachers per subject, which means there are 20 total teachers in the school.
- There are a total of 580 students in the entire school.
- There are 5 teaching assistants from the local teachers college present at the school.
- The school has various clubs that students participate in every day after school (Board Game Club, Football, Soccer, Video Game Club, Band, Computer Science Club, Choir, Basketball, Badminton, Baseball). Students may participate in multiple clubs after school
- Various students at the school have health conditions

There are also some additional considerations to keep in mind for the school that will be important for your analysis:

- We can safely assume that during a student's spare period and during lunch time, they will hang

out with other students that have a spare period and are more likely to hang out with peers in the same grade.
- The school classrooms are all cleaned during the student lunch break.
- It can be assumed that the school is small and the transition of periods happen quickly, it is likely that students leaving a class after a period ends will bump into or be in close contact with the new students coming into that same class.
- The classrooms locations are always static.
- The teaching assistants are highly dynamic, and will usually change classrooms every period, these travels are reflected in the record book. Teaching assistants interact very closely with teachers.
- Students with the same last name may be related and therefore be in closer contact with each other.
- It is safe to assume that all extracurriculars run every day at the end of school, and that the majority of people signed up for them attend regularly.
- For simplicity you can assume that the base set of 20 classes are global for all students, there are no grade specific classes at this school.

For this high school, you will be given a "record book" in the form of an excel spreadsheet, this spreadsheet will contain information about:
- All the students and what classes they attend for each of the school's class periods.
    - Also contains information about health conditions and club participation
- What teachers are mandated to teach which periods.
- What classrooms the teaching assistants are mandated to assist in during each period.
- The CDC has done a general screen of the society populace to see who could possibly have ZBY1, the screening information has been disseminated to all schools in the province. The associated data has been attached to a separate sheet in the record book called "ZBY1 Status".

Finally, the CDC has identified several characteristics of ZBY1 that may be important in your analysis
- Older individuals have a higher chance of catching the virus, as the age difference increases by 2, the chance of catching the virus increases by 50% (for example, a 2 year old would have 50% less chance of catching the virus compared to a 4 year old).
- It is safe to assume that the virus has a chance of spreading every time somebody is in close contact with each other, or if somebody touches a contaminated surface, just like the flu.
- The CDC has ruled that **any** pre-existing health condition increases the risk of an individual catching the virus by 70%.
- The virus can be completely eliminated by commonplace cleaning supplies.

You are tasked with using the record book to make correlations as to what students, teachers, or TAs could have been at risk for catching ZBY1. The virus status sheet in the spreadsheet is to be used as the starting seed and it is up to you to make correlations as to who could have further been infected from that starting set of students. Your correlations must be well reasoned and explainable, and it would help if you could attribute a % based risk for your correlations, although not required.

Each team will most likely have a different set of correlations, there is no static answer to this problem. The correlations that you come up with in your final project will be the product of your personal intuition based on the parameters that we've given you, and your analysis based on the school properties and the additional considerations.

4.0 Competition Deliverables

Teams are required to **design**, **develop**, **provide documentation**, and **construct a presentation** of their project during the build time provided in the first phase of the competition. Proceeding this, the competitors will then present and demonstrate their solution in the second phase of the competition.

Design:

- The design element for this challenge refers to the *planning* of the solutions code. Like all good developers, the individuals should have an idea of how they will split up and solve the problem before jumping into the code. Not only does this allow for a more in-depth understanding of the challenge at hand, it also allows the teams to catch unreasonable ideas before it is too late. Once again, this is open ended, but the judges will be expecting some sort of planning displayed in any fashion to be handed in or presented.

Development/Deliverables:

- As mentioned before, the challenge is open ended, which also allows the deliverables to be as well. Competitors can deliver their solution in any way they seem fit. Some examples include displaying their results through a graphical interface, exporting the data to a file, etc.
- This data, regardless of the medium, **must** be easily understandable to the user who is reading it. The people who would benefit from your solution may not be technologically adept, so the results must be easy to read.
- You can use any programming language to complete this challenge that you seem fit. On top of this, any frameworks and libraries are allowed as long as you properly accredit them in your presentation.
- Instructions on how to submit your code can be found in the next section of this document.

Documentation:

- The documentation element comprises of two components:
    - The **first** being that while someone is reading the solutions code, they should not get lost or confused about what each section is doing. This does **not** mean that every piece of code should be documented. Simpler code should be understandable through clear variable and function names. Full documentation should be left to complex portions that may not be as easily comprehendible.
    - The **second** part of documentation relates to user experience. This means clear instructions on how to install the solution, run it, and anything else that someone working with your program may need while using it.

Presentation:

- The presentation **must be delivered as presentation slides**, but they can be as basic or as complex as you would like. IE, basic slides and you sharing your screen for your demo would work nicely.
- The oral presentation should shortly summarize the **design** process and the **development** process.
- The presentation should introduce the software and its design, alongside why you chose this route. Teams should present the core functions of the software, how the program's components work from a development standpoint, it's user/developer documents, the installation method, and any unique components of the solution that were not suggested in the problem.
- If there were unfinished components, teams need to highlight the mistakes made and provide an explanation on how the problem could be solved in the future.
- If the solution included any open-source libraries, the presentation should acknowledge this in the presentation and possibly highlight the components that contain the code.
- From an engineering standpoint, the presentation should also include who the potential stakeholders may be, what benefits the program would give this stakeholder.\
- Finally, the team is to present a short demo to the judging panel. Judges reserve the right to ask questions during the presentation. Teams must deliver answers within a reasonable time (determined by event official) to avoid deduction and/or a committee investigation (based on the context of the question).
- **Your presentation/demo should show an example set of correlations that your program made, and you should be able to explain and validate them clearly and concisely.**
- **Be no more than 20 minutes in length, NOT including a 10-minute Q & A period**
- Do not wear anything with your school name or logo if participating in video chat while presenting online

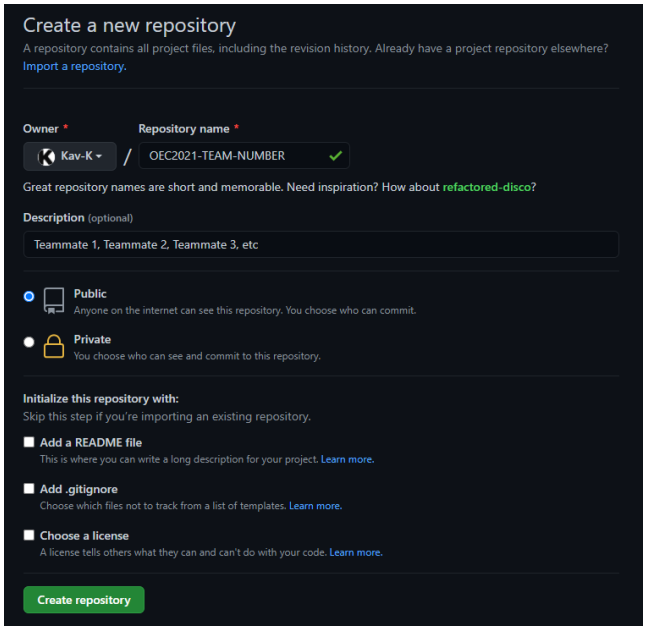
4.1 Submission Details

Your submission is to be fully contained in a GitHub repository. This section contains instructions on how to get registered with GitHub, create a repository, and upload your files to it.

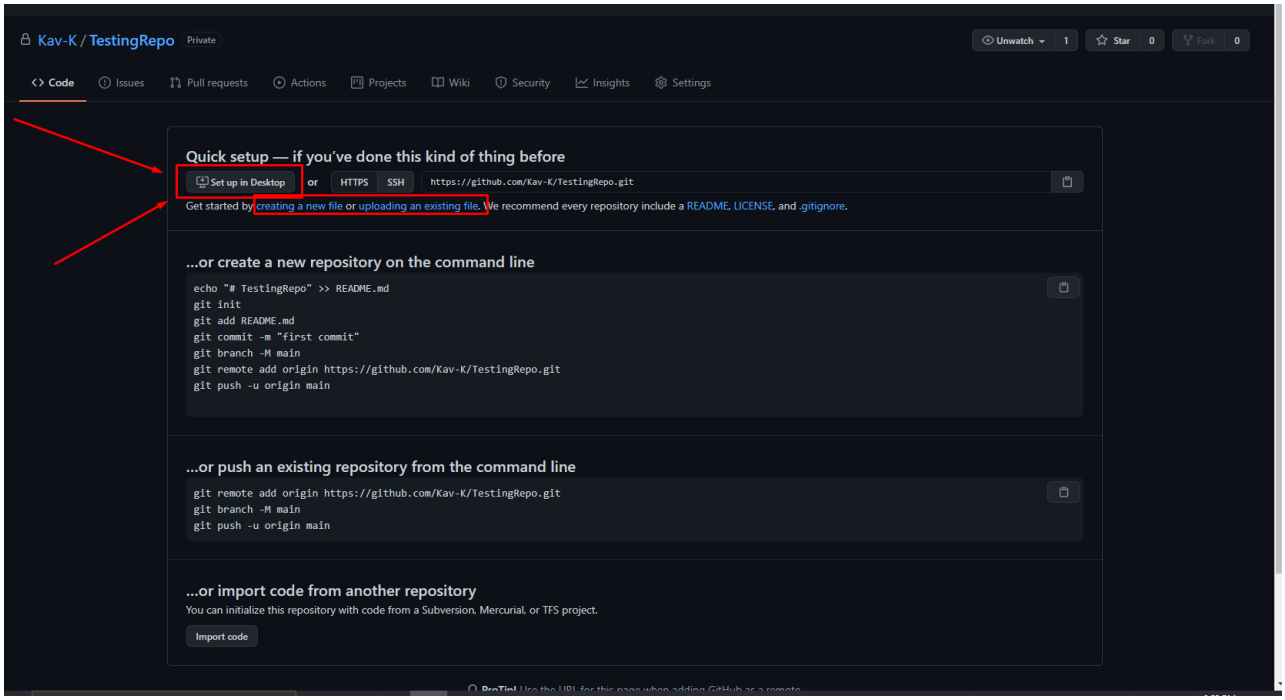
To create a GitHub account, you can visit https://github.com/join, if you already have an account, you can simply log in to your existing account and continue from there.

After registering or logging in, you can create a new repository by clicking the + icon on the top right of your screen, and click “New Repository”;

Afterwards, you will be shown an interface where you can name your repository and create it. Set the name of your repository to be OEC2021-TEAM-{TEAMNUMBER}, and in the description, put the names of all of the team’s members;



After creating your repository, you will need to upload your project files to the repository. Github itself provides command line instructions for doing this in your project directory (**NOTE: You must have Git installed on your system to run these git commands, you can install Git at** https://git-scm.com/book/en/v2/Getting-Started-Installing-Git **).** Alternatively, you can click the “Set up in Desktop” or upload your files manually to the repository;



If you require assistance at any time with setting up your git environment and uploading items to your repository, please contact one of the competition directors as soon as possible.

When the submission period begins, you will be given a google forms link where you are to input your team member names, your team number, and the link to the public repository with your project in it. **After submitting this form, you will be finished the development part of the competition and will be waiting for judging the next day.** There are no resubmissions unless there are extenuating circumstances.

5.0 Competition Scoring and Marking Methods

CRITERIA	*TOTAL POINTS*

PERFORMANCE/CODE:
● Able to MAKE REASONABLE CORRELATIONS
  ○ Correlations made by your program must be reasoned and make sense alongside the provided data
  ○ Quality of correlations will be taken into account
● Structure of code and comments included
● Display of correlations are seamless and easy to understand
● Inline code comments to explain major components
● Code is easily readable
	/30

DESIGN/STRATEGY/ALGORITHM: ● Does the design work? ● Evidence of planning/Design ● How well does the design meet the requirements of the project? ● Does the solution include relevant extra components on top of those requested? ● Did the solution come with appropriate user documents? ● Is the solution simple, sustainable, and future proof? ● Solution is well-thought out and analyzed ● Design synthesizes application of engineering principles ● Simplicity	/30
PRESENTATION: ● Design process ● Design justification ● Design critique ● Were the benefits and principles of the solution clearly explained? ● Was time used appropriately? ● Logical structure in presentation ● If solution had error, were they identified during the presentation? ● Did all team members participate equally in the presentation? ● Voice/Audibility ● Was the code well explained & demonstrated during the presentation? ● Quick overview of user documents and install packages included ● Were the program's components demonstrated thoroughly? ● Response to questions	/30
ORIGINALITY & RESOURCE MANAGEMENT: ● Approach is original and well reasoned ● All group members equally contributed in every stage	/10

● Programs CPU usage IS REASONABLE ● Memory usage efficient	
PENALTIES: ● Code is plagiarized or solution includes open-source code ● Incomplete project submitted/Late submission folder ● Project does not compile/run ● Presentation under/over time by more than 2 minutes ● Insufficient citation ● Absent team member ● Disclosure of school (verbally and physically)	-50 -50 -15 -25 -5 -10 -50 -25
TOTAL	***/100***

6.0 Appendix and Additional Resources

6.1 Useful Libraries

https://openpyxl.readthedocs.io/en/stable/ - The OpenPyXL library can be used to manipulate excel spreadsheets in python

https://poi.apache.org/ - The Apache POI library can be used to manipulate Microsoft documents in the Java language