Western Engineering Competition

Programing Competition

Package

TABLE OF CONTENTS

Introduction

The goal of the Programming competition is to allow students to produce a piece of industry-level software to solve a real-world problem. The teams will use their software development skills, technical writing abilities, and project management skills to design a solution that will be presented to a judging panel.

Competition Leads and Contact Information

Dominic Bazina-Grolinger; Programming Director
dbazinag@uwo.ca

Felix Zhou; Programming Director
jzhou744@uwo.ca

James Morales; Programming Director
jmorale6@uwo.ca

Luis Delotavo; VP Competitions
ldelotav@uwo.ca

General Rules and Guidelines

A. Internet use for finding resources and indirect assistance is permitted. But plagiarism will be penalized
B. You may NOT use a library or open-source code to solve everything for you
C. Any competition questions should be sent to the competition directors on the WEC Discord
D. Visitors are NOT allowed at any time during the competition. Only competition volunteers and judges are allowed
E. USB Keys are permitted provided they do not contain direct help related to the cases
F. You are allowed to use any material provided by the competition leads
G. Workspaces must be cleaned at the end of the competition

Programming Competition Description

It is highly recommended that every participant reads this section very carefully and understands it. This will contain information regarding the problem and assessment.

Problem Background

In light of the increasing frequency and severity of natural disasters, it's imperative to focus on disaster preparedness and recovery efforts. These events, such as hurricanes, wildfires, and earthquakes, can strain our emergency response systems in similar ways as we've seen during crises like the pandemic. Emergency services continue to be overwhelmed and are lacking in resources. These days when a major natural disaster occurs emergency services can be strained to the point where there are not enough firefighters, paramedics, etc. Additionally, global warming has caused these events to be more intense and occur more frequently.

We want to create a system that will help emergency services track natural disasters. This information is invaluable to emergency services as it helps them distribute resources, send warning messages, etc.

The Challenge

The task is to create a program that serves as a GUI to locate, track, and add cases of natural disasters. Each natural disaster must have the following attributes:

- A Name (ex. “Hurricane Dave”)
- Location (latitude and longitude)
- Date (start date)
- Intensity (scale 1-10)

The information on (fictional) current natural disasters will be provided named **MOCK_DATA.csv**. The software must read all of the natural disasters off of this csv file and display them in an organized easy to read format, a good visual representation is preferred. The end user also must be able to create a new natural disaster event with the same attributes mentioned above. All attributes, including the user added points should be retained through a refresh/relaunch of your application. The user should also be able to filter the markers on the map by type, intensity, name, longitude, latitude, and date.

GUI

The GUI is essential to the project. It should serve as a visually appealing and intuitive way for anyone of any technological fluency to be able to use. Be sure to include any features you believe will make the program more useful and accessible to the end users. Additionally, to go above and beyond, the team should create a “twist” or any extra relevant component which adds to the visual appeal, intractability, or usefulness of the program.

Assessment

The judges will review the program’s results and determine whether or not the initial text file was read properly and displayed on the GUI/interface of the team's application. Note that hardcoded natural disaster information will receive no points, they must be read from the file. The judges will then request that the team input a new natural disaster with attributes they make up. They may provide incomplete, missing, or incorrect data to test how the team’s program handles errors in input such as missing fields or an input that falls outside the expected parameters of normal operation (Ex. a latitude of 999999, or an intensity of -1). They will also check if your application retains data between sessions and if the filtering features work.

In addition to the basic operation of the system such as the initial disasters being present and the ability to add natural disasters, each team can also be expected to be graded on the visual appeal of their application, the quality of their presentation, and their final report. More specifics on deliverables can be seen below.

Competition Deliverables

Teams in the Programming competition are required to design, develop, test, provide documentation, and construct a presentation of their project during the allotted time for the first phase of the competition. In the second phase, teams will deliver an oral presentation and demonstrate their solution to a judging panel.

The oral presentation should shortly summarize the team’s layout, the design process, the management process, and the development process. If there were required components that could not be constructed in the time given, 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 highlight the components that contain the code and if there is an alternative library that should have been used. The presentation should introduce the software, present the core functions of the software, how the program's components work from a development standpoint, and any unique components of the solution that were not suggested in the problem. Teams should also provide a short demo of the product and its components for the 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.

In the report you are required to explain the "Principles of Technological Stewardship in Engineering" used in the solution (Please refer to the **appendix** for the criteria). You must select at least 2 out of the 8 pillars to incorporate in your solution.

Note: You can use any programming language you are comfortable with as long as you have the correct environment set up.

All deliverables must be uploaded to Google Classroom.

Each team must submit three components. First will be for the team's code zipped into a folder with the naming convention "[team name]-WEC2024-Programming.zip". Second, the presentation. Third, for the report. Further details for each component and their required naming conventions below.

Report

Include in the submission a 2-page report outlining the issue presented and your solution:

- Identify the stakeholders and the problem, explaining why the problem is relevant to stakeholders
- Identify the chosen solution and how it addresses the problem
- Identify the target users
- Give a high-level overview of the solution design, including any algorithms, GUIs, APIs, etc.
- Explain at least 2 of the "Principles of Technological Stewardship in Engineering" used in the solution (Please refer to appendix for the criteria)
- Shortly summarize the design process, management process, and development process
- If there were required components that could not be constructed in the time given, 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, highlight the components that contain the code and if there is an alternative library that should have been used
- Indicate the core functions of the software and how the program's components work from a development standpoint

Clearly mention any unique components of the solution that were not suggested in the problem. The report must use the following constraints:

- **Font: Times New Roman**
- **Size: 11**
- **Single-spaced**
- **Margins: 1 inch**

Submission Details:

- Reports must be submitted as a Google Doc
- Rename the document to include your team's name. The naming convention should be as follows: "[Team Name]-WEC2024-Programming-Report.doc"

Presentation

Include in the submission folder your final presentation for the judges demonstrating the solution and how it works. Oral presentations should:

- Be no more than 5 minutes in length and there will be a 5-minute Q & A period at the end of the presentation
- Briefly summarize the contents of the report
- Walk the judges through the implementation, including technologies used and any difficult technical challenges the team faced
- Demonstrate the solution with given test cases
- All team members must be present for the presentation
    - Contact competition leads if you have an exception

Submission Details:

- Presentations must be submitted as a Google Slide or Powerpoint Presentation
- Rename the slideshow to include your team name. The naming convention should be as follows: "[TeamName]-WEC-2024-Programming-Presentation.pptx"

Competition Scoring and Marking Methods

CRITERIA

Performance and Code:	
• Able to read all lines from the reading file	/5
• Results are present	/5
• Results will be evaluated against other teams	/3
• GUI/results are interactable	/5
• Able to add GUI	/4
• GUI is visually appealing	/7
• Code is structured and not hard coded to suit an individual case	/1
Total	**/30**
Design, Strategy, and Algorithm:	
• How well does the design meet the requirements?	/14
• Were there relevant extra components on top of those requested?	/3
• Did the solution come with appropriate user documents?	
• How well did the report outline the engineering problem solved?	/3
• Simplicity	/10
	/5
Total	**/35**
Presentation:	
• Design Process	/3
• Design Justification	/3
• Design Critique	/3
• Proper explanation of the benefits and principles	/3
• Proper use of time	/1
• Team member participation breakdown	/1
• Visual Aids	/1
• Flow and Logic	/1
• Was the target audience identified?	/1

• Explanation of code • Quick overview of user documents and install packages • Responses to questions • Audibility **Total**	/3 /2 /2 /1 **/25**
Resource Management: • Solutions are found in the build time • Usage of open source code is properly cited **Total**	 /5 /5 **/10**
Penalties • Plagiarism/Copying Code Without Citations • Incomplete or late deliverables • Absent Team Member (contact director for exceptions) • Unrealistic results or GUI • Program cannot compile or run • Program cannot respond to user input • Presentation under or above 10 minutes • Program does not contain all the requested components or information	 -50 -50 -25 -20 -25 -25 -10 -10
Total	**/100**

Appendix

Principles of Technological Stewardship in Engineering

- Widen Approaches: Used a wide range of new and innovative technologies to address the problem
- Expand Involvement: Performed extensive research on the proposed topic and applied technical expertise
- Advance Understanding: Foster dialogue on technologies used to create awareness
- Shared Action: Shared attention and collective effort ensures optimal results
- Realize Diversity: Ensure the society's (stakeholders and users) needs are met
- Deliberate Values: Understand technological development as inherently infused with values and consider underlying values connected to the project and actively discuss value trade-offs related to the project
- Seek Purpose: Does the work impact the world? Does the problem engage with urgent global issues? Consider broad/positive outcomes in project conceptualization and rationale
- Take Responsibility: Consider potential impacts (Economic, environmental, and social), ways to track the impacts, and precautions to take when serious unintentional problems are encountered. Can this technology be misused/abused? Consider the complex impacts of technology across the entire life cycle.