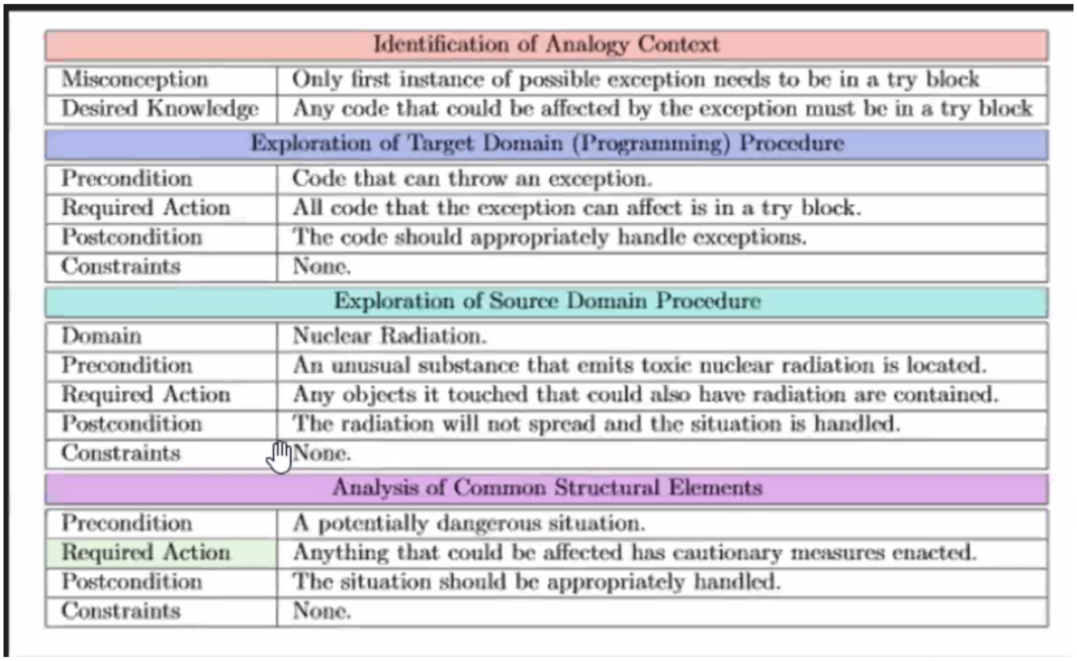
Team name: Programming Analogies
Date, time, and duration: January 17, 2023 4PM EST for 30 minutes
Location: Zoom meeting
Attendance

- Dr Bettin (Scientist)
- Emilie Rummer
- Ethan Jones
- Grayson Wagner
- Jack Grant
- Josh Staples
- Kevin Kulich

Discussions and Decisions: Open-panel discussion for application requirements and design ideas. All questions were clarified, with corresponding notes below in the "General Notes" idea.

General Notes

- Opal framework for analogies



Identification of Analogy Context	
Misconception	Only first instance of possible exception needs to be in a try block
Desired Knowledge	Any code that could be affected by the exception must be in a try block
Exploration of Target Domain (Programming) Procedure	
Precondition	Code that can throw an exception.
Required Action	All code that the exception can affect is in a try block.
Postcondition	The code should appropriately handle exceptions.
Constraints	None.
Exploration of Source Domain Procedure	
Domain	Nuclear Radiation.
Precondition	An unusual substance that emits toxic nuclear radiation is located.
Required Action	Any objects it touched that could also have radiation are contained.
Postcondition	The radiation will not spread and the situation is handled.
Constraints	None.
Analysis of Common Structural Elements	
Precondition	A potentially dangerous situation.
Required Action	Anything that could be affected has cautionary measures enacted.
Postcondition	The situation should be appropriately handled.
Constraints	None.

- Consider displaying where emphasis should be given in presenting an analogy (asterisks in the data)
- Don't use dummy data, come up with creative analogies

Users

- Computer science instructors
- Undergrad TAs, graduate TAs, professors, etc
- Have some level of computing expertise

- May be older but not necessarily elderly (could be near retirement–accessibility concerns)

Requirements

- Browsing analogies
- Searching
- Accessible design

Very Nice to Haves

- Sorting
- Creating analogies (not with a form, walking through components)
- Analogy creation tutorial page (what is each component)
- Measuring who’s seeing what analogies and popular searches
- Responsive design (work on mobile)
- Comparing analogies (side by side comparison)

Kinda Nice to Haves

- Light and dark mode
- Emphasizing different parts of the analogy
- Favoriting analogies (relies on having a user)