Andrew Mayak & Diadem Jones

Professor Yang

CMPSC 132

April 27, 2025

Project 2

1. Description of the project
    - This is a project that implements many areas learned throughout the course
        1. This Project implements these goals:
            1. Encapsulation.
            2. Classes.
            3. Lists.
            4. Stack concepts (LIFO, enqueue - adding to end of list, dequeue – removing from the beginning of list).
            5. Testing.

2. Significance of the Project
    - This project is meaningful as a rudimentary system to assess the likelihood that fire might pose a threat near a person's location. The person can be added to a queue based on the damage stated when the person is created as an object (see section 3).
        - Here is a summary of the two classes:
            1. ImpactedPerson Class
                1. Assesses the expected impact a fire would have of the person.
            2. HelpQueue Class
                1. Adds person to either a priority or standard queue that removes the person that is there the longest.

3. Instructions and functionalities
    - Installation – Install the '.py' file which is v1.0.1 after clicking "Go to file" named the same. Click the tree dots below the other three dots on the top right. press the "download" button. After installing the file open it in your IDE and follow the instructions below.
    - Object Creation - You can create objects that represent People using the ImpactedPerson Class attributes.
    - The Impact class has the attributes:

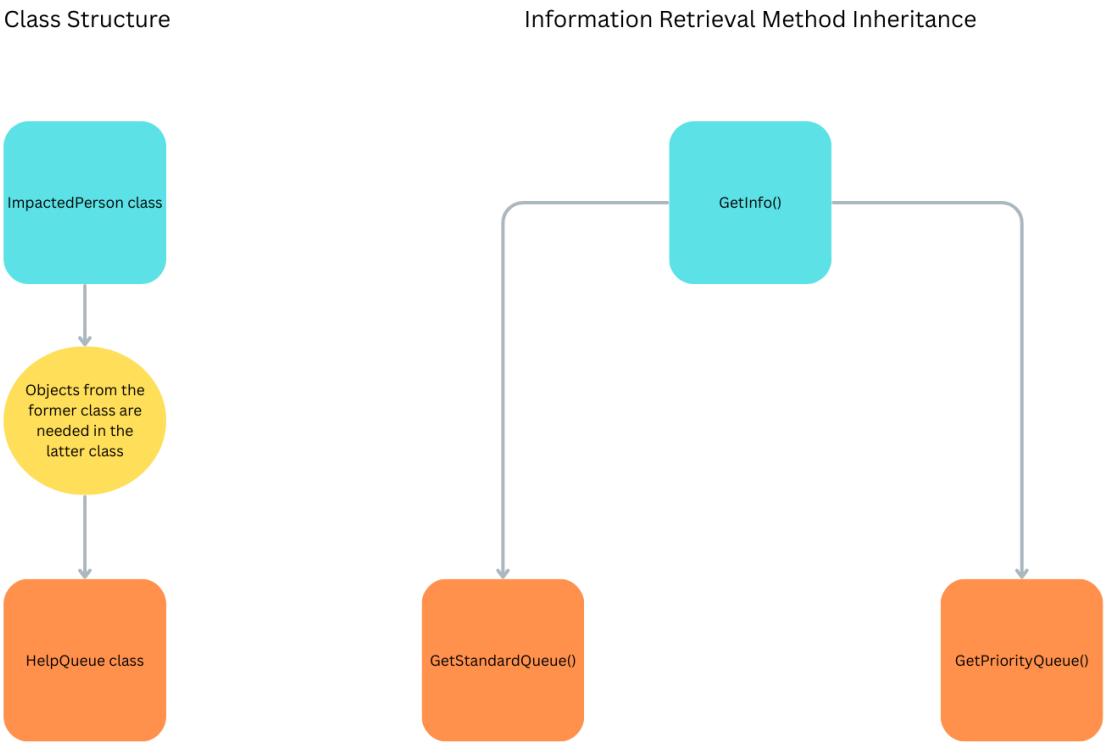
1. name (of owner of impacted area)
2. address (of impacted area)
3. biome (ex: forest, wetlands, mountain etc)
4. damage (either “Minor Damage”, "Moderately Damaged" or "Severely Damaged" describing the damage of the person’s property)
5. (optional: FireSpreadChance can equal 0,1 or 2 ranging representing “safe”, “Caution” , and “danger”)

- Creating objects (specifying people) uses all the attributes listed above:
    1. Object creation examples:
        1. For the ImpactedPerson class specify attributes in the above order:
        2. Example: Person1 = ImpactedPerson("Teddy", "666 No-brainer Ave", "Forest", "Moderately Damaged")
- This class uses these methods:
    1. GetInfo() - prints all the attributes of any person specified as an object using the ImpctedPerson class (the first word before the period needs to the class name.)
        1. Command example: Person1.GetInfo()
    2. FireSpreadAssessment() – Assesses if a person is safe based on Humidity Percentage and Windspeed Percentage (written like using decimals).
        1. Person1.FireSpreadAssessment(0.18, 0.25)

- The HelpQueue class has the attributes:
    1. __Queue (Do not define)
    2. __PriorityQueue (Do not define)
- Creating objects (specifying people) uses all the attributes listed above:
    1. Object creation examples:
        1. Example: Queue1 = HelpQueue()
- This class uses these methods:
    1. AddToQueue() – Add person to Queue based on whether or not the person’s specified damage is “Severely Damaged” and prints out which queue they are added to.
        1. Example: Queue1.AddToQueue(Person1)
            1. For this you need to specify the person from the ImpactedPerson class in order to add to the queue
    2. RemoveFromQueue() – Removes person from Queue starting from the Priority Queue and transitioning to the Standard Queue once the Priority Queue empties.
        1. Example: Queue1. RemoveFromQueue ()
    3. GetStandardQueue () - prints all the attributes of all prople in the standard queue
        1. Command example: Queue1.GetStandardQueue()
    4. GetPriorityQueue () - prints all the attributes of all people in the standard queue

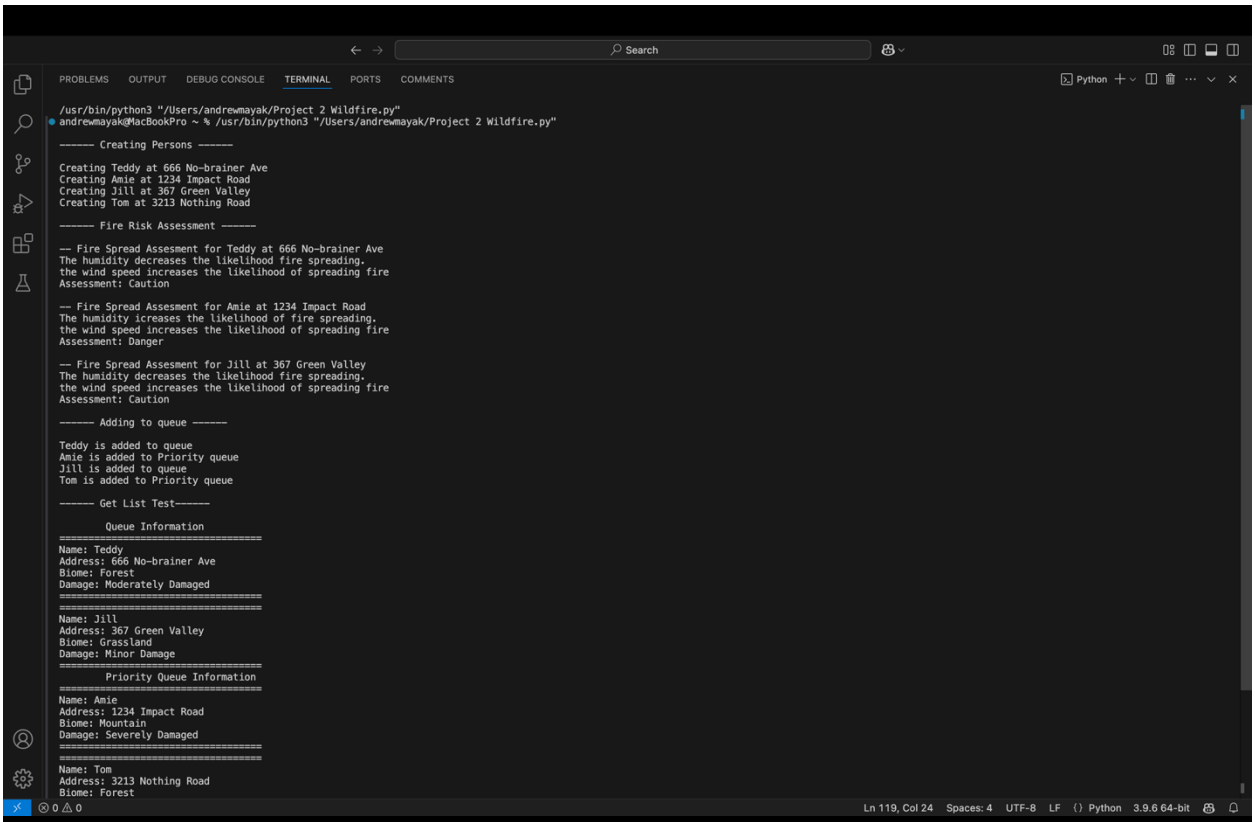
1. Command example: Queue1.GetPriorityQueue()

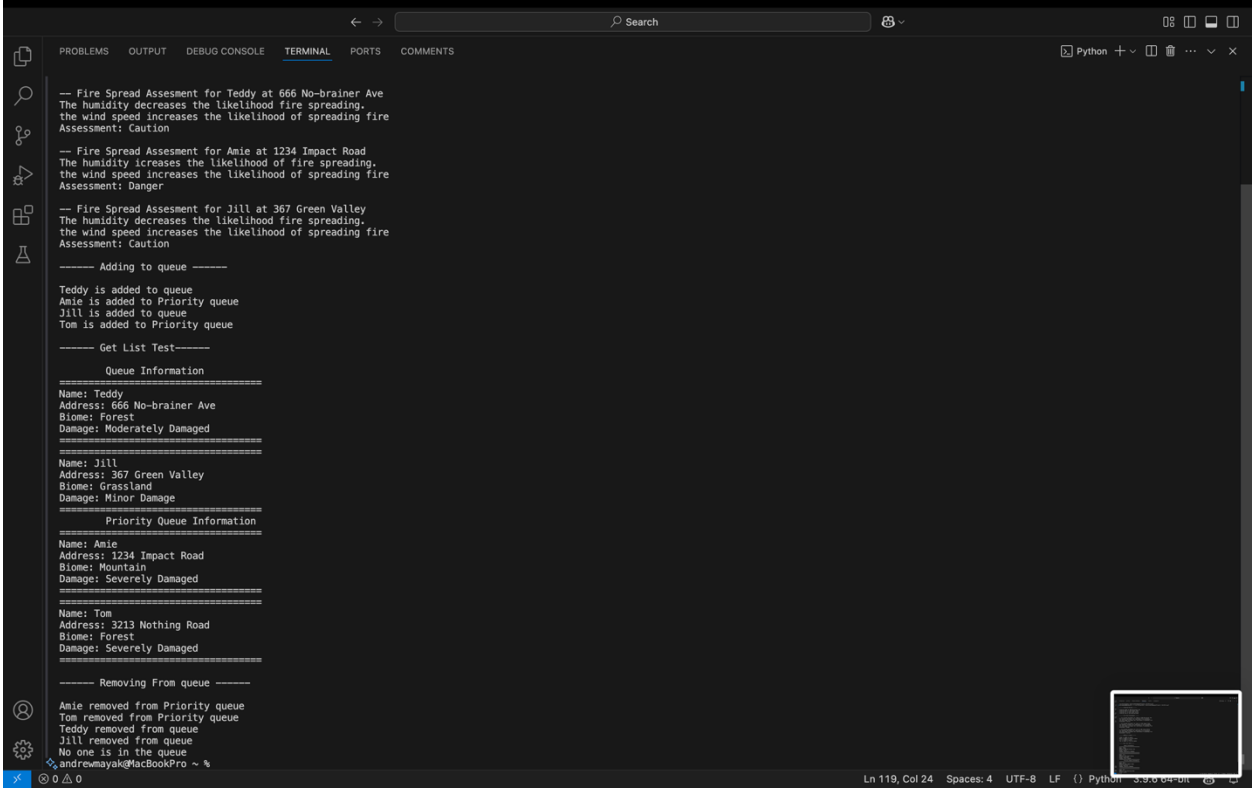
4. Code Structure (diagram)

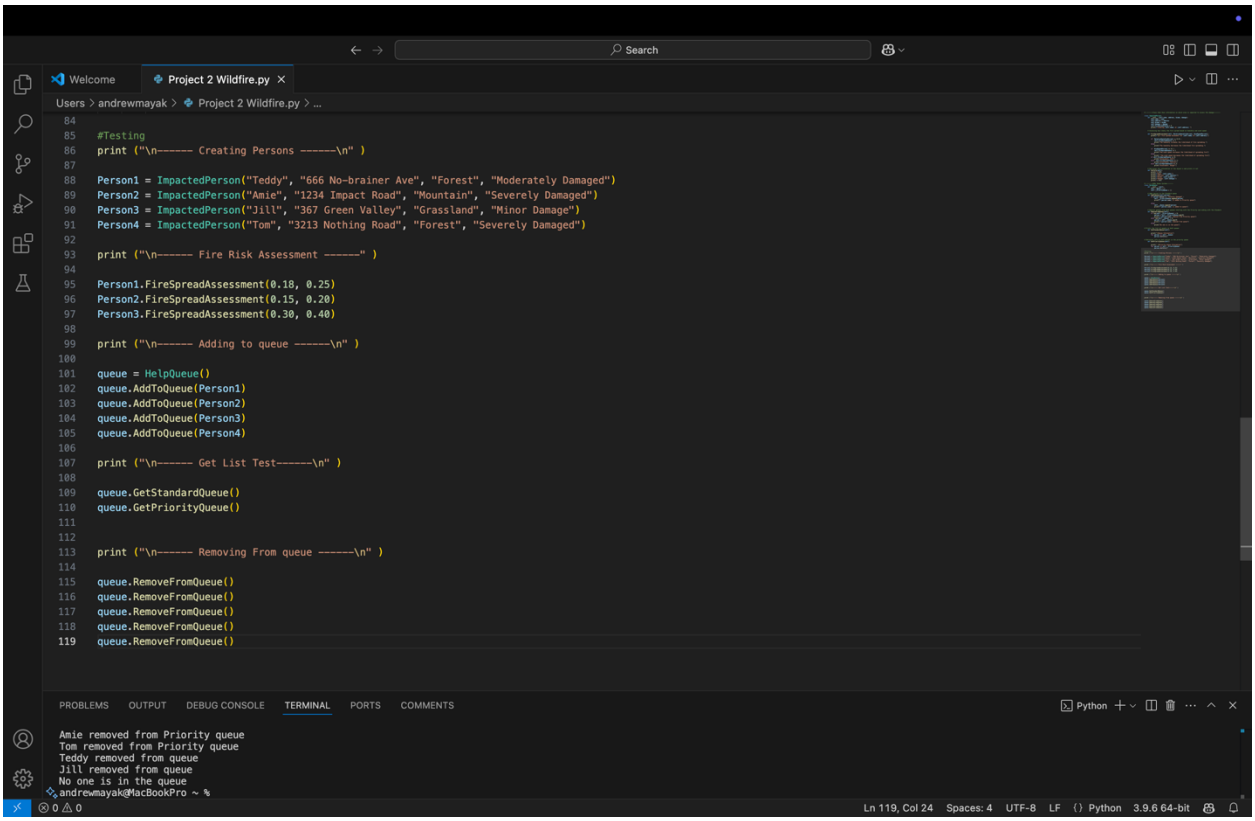


5. This is a project that Achieves the many goals:
    1. This Project implements:
        1. Encapsulation – private queues are implemented
        2. Classes - implemented
        3. Lists – used via queues
        4. Stack concepts – objects are added to the end of list and removed from the beginning
        5. Testing - check section 6

6. Test Results and execution results (All Screenshots are Below).
    - All screenshots have the implemented multiple objects in different Queues as well as tests and here are the comments for the screenshots:
        1. Product Class – functional, methods work, objects can be created. Retrieves information using GetInfo().
        2. HelpQueue – functional, methods work, objects created from prior class can be added to both the priority and standard queue. Retrieves information using GetStandardQueue() and GetPriorityQueue().


Search
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python
/usr/bin/python3 "/Users/andrewmayak/Project 2 Wildfire.py"
andrewmayak@MacBookPro ~ % /usr/bin/python3 "/Users/andrewmayak/Project 2 Wildfire.py"
------ Creating Persons ------
Creating Teddy at 666 No-brainer Ave
Creating Amie at 1234 Impact Road
Creating Jill at 367 Green Valley
Creating Tom at 3213 Nothing Road
------ Fire Risk Assessment -------
-- Fire Spread Assesment for Teddy at 666 No-brainer Ave
The humidity decreases the likelihood fire spreading.
the wind speed increases the likelihood of spreading fire
Assessment: Caution
-- Fire Spread Assesment for Amie at 1234 Impact Road
The humidity icreases the likelihood of fire spreading.
the wind speed increases the likelihood of spreading fire
Assessment: Danger
-- Fire Spread Assesment for Jill at 367 Green Valley
The humidity decreases the likelihood fire spreading.
the wind speed increases the likelihood of spreading fire
Assessment: Caution
------ Adding to queue -------
Teddy is added to queue
Amie is added to Priority queue
Jill is added to queue
Tom is added to Priority queue
------ Get List Test------
Queue Information
Name: Teddy
Address: 666 No-brainer Ave
Biome: Forest
Damage: Moderately Damaged
Name: Jill
Address: 367 Green Valley
Biome: Grassland
Damage: Minor Damage
Priority Queue Information
Name: Amie
Address: 1234 Impact Road
Biome: Mountain
Damage: Severely Damaged
Name: Tom
Address: 3213 Nothing Road
Biome: Forest
Ln 119, Col 24 Spaces: 4 UTF-8 LF Python 3.9.6 64-bit


Search
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
Python
-- Fire Spread Assesment for Teddy at 666 No-brainer Ave
The humidity decreases the likelihood fire spreading.
the wind speed increases the likelihood of spreading fire
Assessment: Caution
-- Fire Spread Assesment for Amie at 1234 Impact Road
The humidity icreases the likelihood of fire spreading.
the wind speed increases the likelihood of spreading fire
Assessment: Danger
-- Fire Spread Assesment for Jill at 367 Green Valley
The humidity decreases the likelihood fire spreading.
the wind speed increases the likelihood of spreading fire
Assessment: Caution
------ Adding to queue -------
Teddy is added to queue
Amie is added to Priority queue
Jill is added to queue
Tom is added to Priority queue
------ Get List Test------
Queue Information
Name: Teddy
Address: 666 No-brainer Ave
Biome: Forest
Damage: Moderately Damaged
Name: Jill
Address: 367 Green Valley
Biome: Grassland
Damage: Minor Damage
Priority Queue Information
Name: Amie
Address: 1234 Impact Road
Biome: Mountain
Damage: Severely Damaged
Name: Tom
Address: 3213 Nothing Road
Biome: Forest
Damage: Severely Damaged
------ Removing From queue -------
Amie removed from Priority queue
Tom removed from Priority queue
Teddy removed from queue
Jill removed from queue
No one is in the queue
andrewmayak@MacBookPro ~ %
Ln 119, Col 24 Spaces: 4 UTF-8 LF



7. Conclusion:

Andrew's experience: In this project we had issues with GitHub with updating code and checking if it works. In regard to coding, I had issues with implementing GetStandardQueue(), GetPriorityQueue () and GetInfo().The issues with those methods were aesthetically minor like when I wrote print(GetInfo())for the latter two functions which printed out "None" after every call but fixed when removing print()all info on each attribute was printed anyway in the GetInfo() function.

This project is not very large. One cannot specify greater priorities when assigning people to Queues, but you can retrieve relevant information such as an impacted person's address and their location.

This course helped me understand the fundamental Object-Oriented principles, one of which being encapsulation that I implemented with the Queues. This course helped me better understand classes, methods and attributes which I have implemented and listed above. The concept of stacking exists since that object at the end of the list to the stack is the first one removed.

Diadem's experience: I faced a similar challenge as my project partner, but my focus was on coming up with examples that represent key elements of a wildfire. This included the people affected by the wildfire, the specific locations, and determining the appropriate scale for assessing the fire's spread, among other factors. The main issue I encountered

was integrating my code with that of my team member. Through this project, I gained a better understanding of Object-Oriented Programming concepts, as well as the use of queues, deques, and stacks.

8. Personal Quality Assessment:

   We believe we have achieved enough and revised the project multiple times. Comments were added and the naming these aspects such as methods were revised for better understandability. On the users end, we took actions in helping the user understand with feedback.