I will be choosing to go with Canfield. I learned it also has the nickname of Demon, so I may use that in reference at times while coding because I enjoy it. I attached class, sequence, and activity UML diagrams. I was planning on using inheritance when it came to how each stack acts from the parent stackType. Enums will be used for the cards. I’ll end up adding in error handling within the classes to counter if the user tries to put cards where they can’t go or inputting an unavailable input. The overloading operators will be in the stackType class to add a card to a stack. How I think the user will see the game will be as follows:  
The top card is the base of the stack, and the bottom card is the current top of the stack.  
using: clubs (♣), diamonds (♦), hearts (♥), and spades (♠)  
/\*game start\*/  
~~Select random number from 1-100 to start game: 33 (this is the shuffle feature)~~  
-----------------------------   
-F1- -F2- -F3- -F4-  
|Q♣| | -- | | -- | | -- |  
|Q♣| | -- | | -- | | -- |  
----------------------------  
-T1- -T2- -T3- -T4-   
|2♥| |3♠| |J♥| |5♦| R: A♦  
| -- | | -- | | -- | | -- |   
-----------------------------  
S: 7♣  
Do you want to move cards? Y/N: N  
-----------------------------  
-F1- -F2- -F3- -F4-  
|Q♣| | -- | | -- | | -- |  
|Q♣| | -- | | -- | | -- |  
---------------------------  
-T1- -T2- -T3- -T4-   
|2♥| |3♠| |J♥| |5♦| R: A♦  
| -- | | -- | | -- | | -- |   
-----------------------------  
S: K♣  
Do you want to move cards? Y/N: Y  
Enter stack abbreviation to move: S  
Enter stack abbreviation to destination: F1  
-----------------------------  
-F1- -F2- -F3- -F4-  
|Q♣| | -- | | -- | | -- |  
|K♣| | -- | | -- | | -- |  
----------------------------  
-T1- -T2- -T3- -T4-   
|2♥| |3♠| |J♥| |5♦| R: A♦  
| -- | | -- | | -- | | -- |   
-----------------------------  
S: 3♣

Diagram

Description automatically generated

Diagram

Description automatically generated

Diagram

Description automatically generated