AI prompt: “Example of Generic classes, “how and why we use enum types in Java.”

YouTube:

https://www.youtube.com/watch?v=1TMtqy0Glh8

https://www.youtube.com/watch?v=K1iu1kXkVoA

https://www.youtube.com/watch?v=Yat8l37XGFA

https://www.youtube.com/watch?v=wq9SJb8VeyM

https://www.youtube.com/watch?v=k0iTgTuiEGY

How I used AI:

During this project, I encountered several compile-time errors that initially prevented the code from running.

One major issue was that my StandardCard and UnoCard classes did not initially implement the ICard interface. Since the Deck class was defined with a generic type constraint (<T extends ICard>), Java requires that any card type used with the deck must implement ICard. Without this implementation, the compiler generated an error stating that the inferred type did not meet the required bounds. I didn't realize at first that I was allowed to change the card classes to include implements ICard.

I tried defining the generic type as <ICard extends ICard>, but that resulted in an inheritance error. After further research on generics and watching additional tutorial videos, I learned that <T extends ICard> means any type that implements ICard, which resolved the issue.

I learned that newline characters, such as \n, must be included in System.out.println statements to format the output correctly. Without the \n, the output was harder to read. It improved clarity without affecting the program logic.

I attempted to make StandardCardFactory generic using <T>, which caused multiple compilation errors. I later understood that factory classes in this project should return concrete types as List<StandardCard>, whereas generics should be used in the Deck class and the testDeck method.