




- Introduction
 - a. I coded a program that allows you to play the card game Black Jack and War as many times as you'd like. The reason for these choices is because I used to play the two card games with my sister and brother when I was younger and we all loved these games.
 - b. I spent maybe about 5 days spread out about 2 weeks working on this project, it has **791 lines** shared between the main program file and all the header files. There were **3 classes in total: Deck, Hand, Card**.
 - c. The github repository : <https://github.com/KELECHIDAVIS/BlackJackAndWar>
- Approach To Development
 - a. I started with drawing/planning out all the subclasses that would be needed for my desired program. Then took my time and implemented it according to my blueprints.
 - b. Mostly used version control (Github) at the end so I could transfer the files from pc to my laptop and vice versa
- Sample Input and Output
 - a. **Top level :**
 - First you choose which game you would like Black Jack or War and respond with one character (b/w)
 - b. **BlackJack:**
 - The dealer keeps asking if you would like another hit until you say no (the character n) or you bust (have a hand value of over 21)
 - Then at the end the you are asked if you would like to play blackjack for another round (y/n)
 - If no is selected you are sent back to the top level
 - c. **War:**
 - Every round after the deck is split, the player is prompted to press the enter key to deal their top card
 - If both that card and the computer's top card equals each other, WAR is started, three cards from each player is put into the pot and the player is prompted with an enter key again.
 - War games can run long so you can hold enter to reach the end quicker.
- Checkoff Sheet
 - a. Container Classes
 - **List (line 28 Deck.h)** : Used as a way to store the card suit names
 - **Set (line 48 main.cpp)** : Used to check for valid console responses
 - **Map (line 97 main.cpp)** : Used to keep track of player scores
 - **Stack (line 24 Deck.h)** : The main structure used to hold cards in deck class
 - **Queue (line 24 Hand.h)** : The main structure used to hold cards in hand class
 - b. Iterators:
 - **Forward Iterator (line 128 Deck.h)** : Used to get correct suit name when printing

- c. Algorithms
 - **Search (Recursive) (line 95 Hand.h)** : I developed a recursive function that searches the hand for aces and returns the highest value that the hand could possibly be with aces without busting for blackjack.
 - **Fill (line 39 Deck.h)** : Fills deck with cards
 - **Min/Max (line 139->226 main.cpp)** : at the end of each round of blackjack we check who has the highest hand value without going over 21
- Documentation of Code
 - a. Pseudo-Code:  PseudoCode.pdf
 - b. UML:  UML Diagram.pdf
 - c. Flowchart:  Flowchart.pdf