Name: Ethan Huang

ID: 406244368

**Notable Obstacles:**

An obstacle I faced was figuring out whether there was a more efficient way to code this other than using brute force if-statements. I found that I could make it more efficient by grouping together duplicates. Another obstacle I realized was the order of the if statements (whether to check pairs, soft, or the total values first). If total values were before soft, then there would be cases where if a hand was (ace, 8), the if statement would use the strategy for 9 instead of the strategy for ace, 8. Besides this, the only other obstacle was figuring out how to build the program on the g32 linux server.

**Test cases:**

Hand(ace, seven) - any dealer card

* This case would test whether or not the strategy for the total value of 8 would be called, or the strategy for the soft hand for ace seven would be called.

Hand(five, five) - any dealer card

* This would test whether or not the pairs were functional

assert(Hand(six, eight).evaluateHand(five) == Hand(five, nine).evaluateHand(five))

* This would test whether or not the duplicate strategy if statements were working properly

assert(Hand(ace, seven).evaluateHand(seven) == Choice::STAND)

* This checks whether or not more complicated strategies (hands where there are 3 different choices depending on dealer card) work.

assert(Hand(ace, seven).evaluateHand(six) == Choice::DOUBLESTAND)

* Same reason as above

assert(Hand(ten, ten).evaluateHand(four) == Choice::STAND)

* Checks if hands where all options are the same regardless of dealer-hand work correctly

assert(Hand(two, three).evaluateHand(six) == Choice::HIT)

* Checks if hands that are not on the strategy card work correctly

assert(Hand(jack, queen).evaluateHand(six) == Choice::STAND)

* Checks if face cards work correctly

For all of the above, you should also assert(Hand.isPair()) and assert(Hand.isSoft()) to check whether those two methods are also functional.

My program works for all of the above cases, I can’t think of anything that would cause it to fail.