**Card Game**

*Student A Candidate Number: 014485*

*Student B Candidate Number: 054530*

*(Marks to be allocated 50:50)*

**Table of Contents**

[Development Log 3](#_Toc54484895)

[Design Analysis 4](#_Toc54484896)

[Test Design 5](#_Toc54484897)

# Development Log

## 23/10/2020 – 12:45pm (5 hours) [*Signed: 014485, 054530]*

Started with A as driver and B as navigator. Created *CardGame.java*, and added input for number of players and location of pack to load. Created objects for the list of cards in the pack, list of players, hand of each player, and deck of each player. Added exceptions for non-integer inputs, non-existing file inputs, and packs with invalid values or the wrong number of cards.

Swapped roles (B as driver, A as navigator), then added a method *dealCards(packArr, numPlayers, playerObj, deckObj)* to deal cards to the players, and a method *countFrequencies(packArr, numPlayers)* to check if the pack of cards is winnable. Created *Card.java* with setter and getter methods for card values and holders. Created *CardDeck.java* with getter methods for the owner, deck, deck card, and deck size, a setter method for deck card, and methods to remove from deck and add to deck.

## 25/10/2020 – 12:30pm (3 hours 30 minutes) [*Signed: 014485, 054530]*

Started with B as driver and A as navigator. Created threaded class *Player.java*, and created method *identical()* to check if the player has four cards of the same value to win the game.

Swapped roles (A as driver, B as navigator). Student A optimised *Player.java* by creating method *play()* for players to continually draw and discard cards until a winner is found. A new method *isWinner(player)* was created in *CardGame.java,* replacing *identical()*, to perform checks for a win condition and declare the winner of the game.

## 28/10/2020 – 12:30pm (1 hour 30 minutes) [*Signed: 014485, 054530]*

Started with A as driver and B as navigator. Student A created a new method *random(len)* to randomise the index of which card from the deck to take from, and end the game if there are no cards left. The *play()* method was updated to utilise this new random method.

# Design Analysis

# Test Design