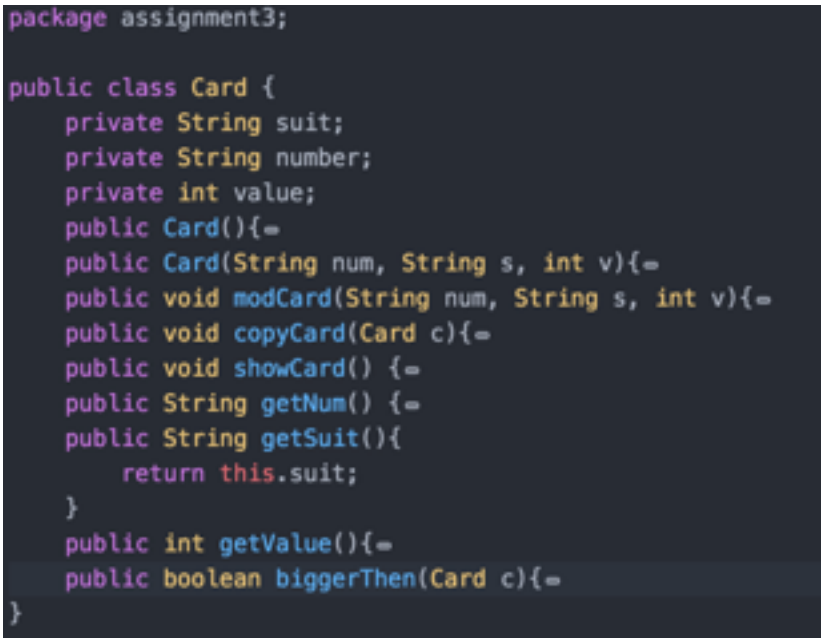
(1)Name and school ID:

劉冠宏 B02705010

(2)Class Structure and Reason

In this homework, I design 6 classes. Including Card, Player, OldMaid, VariantOne, VariantTwo and PlayGame(each of them has their own java file). Next, I will introduce them one by one.

- **Card**



Card is to store the information of every particular cards including **suit**, **number** and **value**. There are several ways to interact with a card instance. First, **modCard(String, String,Int)** provides a way for user to reset the cards. Second, **copyCard(Card c)**, will pass the value of c to the calling card instance. **showCard()** prints out the suit and the number. At last **biggerThen(Card c)**, is a function for class types to compare with one another (return true if the call instance is bigger than card c).

The other functions, **getNum()**, **getSuit()** and **getValue()** returns the corresponding variable.

- **Player**

Player is to store the information of a player including **ID**, **cardNum**(Card numbers on hand), **Win**(whether a player won),**carSet**(card array to store the cards on hand). Methods including **modPlayer(…)** used when distribute cards from deck, **showPlayer()** printing out the player ID and cardSet.

dropCard() dropping the pairs in cardSet, **lossCard() and drawCard()** for one to draw cards from another, **sortCard()** used to sort the cardSet in ascending order, **checkWinner()** returning true when a player won, at last **report()** is used when we wants to know the card numbers while **reportID()** is for ID.

- **OldMaid**

OldMaid is to store the information of a old maid game, including **cardArray**(the deck using for the game)**, players**(the players participating the game). The methods are, **createDeck()** used to create a card deck in the way designer wanted, **dealCards(CardArray)** used to deal the cards to each player in players, **playerDropCards(players)** calling the dropCard(in class Player) for each player in players, **gameStart()** is used to start the game with calling the functions of players and cards to draw card from one another, it also prints out the messages of games.

```
package assignment3;
import java.util.*;
import assignment3.Card;
import assignment3.Player;

public class OldMaid{
    Card[] cardArray = createDeck();
    Player[] players = dealCards(cardArray);
    public Card[] createDeck(){
    public Player[] dealCards(Card[] cardArray){
    public void playerDropCards(Player[] players){
    public void gameStart(){
    public void shuffleCard(Card[] array){
}
```

- **VariantOne**

Extends from the OldMaid while in **createdDeck**, deck created is different (52 cards without jokers). And in **dealCards(cardArray)** it deals the first 51 cards (leaving one card single) to the players. At last, **whoIsLoser()** return the loser information and print out the messages of mocking.

```
package assignment3;
import assignment3.Card;
import assignment3.Player;
import assignment3.OldMaid;

public class VariantOne extends OldMaid{
    public Card[] createDeck(){
    public Player[] dealCards(Card[] cardArray){
    public Player whoIsLoser(){
}
```

- **VariantTwo**

Extend from VariantOne with a extra method added, **punishment().** The loser will have to draw a card from the shuffled deck(52 cards without joker) to decide his/her punishment. The punishment depends on the suit of drawer card. Suit C stands for 100 push-ups, D for 100 knee-bands, H for 100 sit-ups and S for 10km jogging.

```
package assignment3;
import assignment3.Card;
import assignment3.Player;
import assignment3.OldMaid;
import assignment3.VariantOne;

public class VariantTwo extends VariantOne{
    public void punishment(){
}
```

- **PlayGame**

Where the program runs on. Will create game instance for the three games.

```
import assignment3.OldMaid;
import assignment3.VariantOne;
import assignment3.VariantTwo;

public class PlayGame{
    public static void main(String[] args){
        System.out.println("Now Playing: OldMaid");
        OldMaid game = new OldMaid();
        game.gameStart();

        System.out.println("Now Playing: VariantOne");
        VariantOne game1 = new VariantOne();
        game1.gameStart();
        game1.whoIsLoser();

        System.out.println("Now Playing: VariantTwo");
        VariantTwo game2 = new VariantTwo();
        game2.gameStart();
        game2.punishment();
    }
}
```

(3)How a human player should play with your program for the two variants?

The part how each player draws cards stays the same for both of the variants and the origin maid game. However, variantOne(jiji-nuki) is using a different deck. When the game started , we randomly select one card from deck and remove it and when distributing the cards, player0 to player2 will get 13 cards and player3 gets 12 cards. While for variantTwo, which is extended from variantOne, the losing player will be punished. The punishment is decided by the game ends. Having the loser draw a card from a shuffled deck and the suit of the card will represent different kinds of punishment(As shown above).

(4)How you tested the correctness of your program?

As we are rewriting the program of homework1. For the original old maid, what I focused is to make sure that whether the variables are modified with the right state by each functions correctly because in homework1 the routine of the game is tangled in the static main section in PlayGame. But cards and players are classes of their own.

After making share the OOP old maid works fine, I started to write the variants. What I checked I'm this part is whether the method is successfully overriding.

(5)The sample output from each variant of your program.

I Skipped the result of the original old maid.

Now Playing: VariantOne
Deal cards
Player0: C3 H5 D6 S6 C7 D9 S9 C10 D10 H10 HQ SK SA
Player1: D2 H2 D3 H3 C5 D5 C6 S7 C8 H8 S8 CQ HA
Player2: C2 S2 S3 C4 H4 S5 D7 H9 S10 CJ DQ SQ HK
Player3: D4 S4 H6 H7 D8 C9 DJ HJ SJ CK CA DA
Drop cards
Player0: C3 H5 C7 H10 HQ SK SA
Player1: C6 S7 S8 CQ HA
Player2: S3 S5 D7 H9 S10 CJ HK
Player3: H6 H7 D8 C9 SJ CK
Game start
Player0 draws a card from Player1 S7
Player0: C3 H5 H10 HQ SK SA
Player1: C6 S8 CQ HA
Player1 draws a card from Player2 HK
Player1: C6 S8 CQ HK HA
Player2: S3 S5 D7 H9 S10 CJ
Player2 draws a card from Player3 SJ
Player2: S3 S5 D7 H9 S10
Player3: H6 H7 D8 C9 CK
Player3 draws a card from Player0 H10
Player3: H6 H7 D8 C9 H10 CK
Player0: C3 H5 HQ SK SA
Player0 draws a card from Player1 HK
Player0: C3 H5 HQ SA
Player1: C6 S8 CQ HA
Player1 draws a card from Player2 S3
Player1: S3 C6 S8 CQ HA
Player2: S5 D7 H9 S10
Player2 draws a card from Player3 H7
Player2: S5 H9 S10
Player3: H6 D8 C9 H10 CK
Player3 draws a card from Player0 C3
Player3: C3 H6 D8 C9 H10 CK
Player0: H5 HQ SA
Player0 draws a card from Player1 S8
Player0: H5 S8 HQ SA
Player1: S3 C6 CQ HA
Player1 draws a card from Player2 S5
Player1: S3 S5 C6 CQ HA
Player2: H9 S10
Player2 draws a card from Player3 C3
Player2: C3 H9 S10
Player3: H6 D8 C9 H10 CK
Player3 draws a card from Player0 H5
Player3: H5 H6 D8 C9 H10 CK
Player0: S8 HQ SA
Player0 draws a card from Player1 CQ
Player0: S8 SA
Player1: S3 S5 C6 HA
Player1 draws a card from Player2 H9
Player1: S3 S5 C6 H9 HA
Player2: C3 S10
Player2 draws a card from Player3 H10
Player2: C3
Player3: H5 H6 D8 C9 CK
Player3 draws a card from Player0 SA
Player3: H5 H6 D8 C9 CK SA
Player0: S8
Player0 draws a card from Player1 H9
Player0: S8 H9
Player1: S3 S5 C6 HA
Player1 draws a card from Player2 C3
Player1: S5 C6 HA
Player2:
Player2 wins
Basic game over
Continue
Player3 draws a card from Player0 H9
Player3: H5 H6 D8 CK SA
Player0: S8
Player0 draws a card from Player1 HA
Player0: S8 HA
Player1: S5 C6
Player1 draws a card from Player3 CK
Player1: S5 C6 CK
Player3: H5 H6 D8 SA
Player3 draws a card from Player0 S8
Player3: H5 H6 SA
Player0: HA
Player0 draws a card from Player1 CK
Player0: CK HA
Player1: S5 C6
Player1 draws a card from Player3 H6
Player1: S5
Player3: H5 SA
Player3 draws a card from Player0 CK
Player3: H5 CK SA
Player0: HA
Player0 draws a card from Player1 S5
Player0: S5 HA
Player1:
Player1 wins
Player3 draws a card from Player0 HA
Player3: H5 CK
Player0: S5
Player0 draws a card from Player3 H5
Player0:
Player3: CK
Player0 wins
Bonus game over
Player3 is jiji-nuki

Now Playing: VariantTwo
Deal cards
Player0: H2 S2 H3 H4 D6 S6 H7 S7 C9 D9 SK HA SA
Player1: C2 D2 S3 D4 C5 C6 C8 D8 H10 S10 DJ HQ HK
Player2: D3 C4 S4 D5 H5 C7 H8 S8 C10 D10 SJ DQ CK
Player3: C3 S5 H6 D7 H9 S9 CJ CQ SQ DK CA DA
Drop cards
Player0: H3 H4 SK
Player1: S3 D4 C5 C6 DJ HQ HK
Player2: D3 C7 SJ DQ CK
Player3: C3 S5 H6 D7 CJ DK
Game start
Player0 draws a card from Player1 C5
Player0: H3 H4 C5 SK
Player1: S3 D4 C6 DJ HQ HK
Player1 draws a card from Player2 CK
Player1: S3 D4 C6 DJ HQ
Player2: D3 C7 SJ DQ
Player2 draws a card from Player3 C3
Player2: C7 SJ DQ
Player3: S5 H6 D7 CJ DK
Player3 draws a card from Player0 C5
Player3: H6 D7 CJ DK
Player0: H3 H4 SK
Player0 draws a card from Player1 D4
Player0: H3 SK
Player1: S3 C6 DJ HQ
Player1 draws a card from Player2 DQ
Player1: S3 C6 DJ
Player2: C7 SJ
Player2 draws a card from Player3 CJ
Player2: C7
Player3: H6 D7 DK
Player3 draws a card from Player0 SK
Player3: H6 D7
Player0: H3
Player0 draws a card from Player1 C6
Player0: H3 C6
Player1: S3 DJ
Player1 draws a card from Player2 C7
Player1: S3 C7 DJ
Player2:
Player2 wins
Basic game over
Continue
Player3 draws a card from Player0 C6
Player3: D7
Player0: H3
Player0 draws a card from Player1 DJ
Player0: H3 DJ
Player1: S3 C7
Player1 draws a card from Player3 D7
Player1: S3
Player3:
Player3 wins
Player0 draws a card from Player1 S3
Player0: DJ
Player1:
Player1 wins
Bonus game over
Player0 is jiji-nuki
The card drawed from the deck is HQ
Player0! Your punishment is 100 Sit-ups!