Create Task for Blackjack

Victoria Ruan

2a.

JAVA is the programming language used to build Blackjack game. The purpose of the program is to construct a game interface for people to play the Blackjack game. Blackjack is a card game in which players try to acquire cards with a face value as close as possible to 21 without going over. The video illustrates two rounds of BlackJack game played by a computer player and a human player from the terminal end.

2b.

Before writing the main function of the code, I outlined the necessary components of the game and constructed different classes incrementally. There were three classes: Card, Deck and Player. Card represented each individual card used in the game with the attribute values and shape as well as the ability to flip. Deck was built based on an array of Card and was able to shuffle and return a card to the player. Player had attribute of roles and was able to add a card to the hands, see the card values and calculate the sum of the drawn cards. After the three classes were built, the BlackJack file was created with the main function with the instantiation of the classes. The iterative development process was demonstrated in the while loop in the BlackJack. As long as the game conditions were met, the program would continuously ask the player to choose a card. The difficulty I encountered in the development was the frequent occurrence of NullPointerException which was caused by the absence of instance initialization. The development of Card and Deck classes were collaborative, and the rest were independent.

2c.

The program code pasted below is responsible for asking the human player to draw a card, adding the drawn card to the player's card list and helping the virtual computer player to determine if to draw a card. The bound for the loop is 12 since it is the maximum number for a deck of card to be summed without getting over 21. The reason to choose 15 as a criteria for the computer is that the card has the value only from 1 to 13, so there are 6 numbers (1-6) for 15 to be added without exceeding 21 and also 7 numbers (7-13) to surpass 21. The probability of going over and not are approximately equal. Thus, 15 is chosen for the criteria of drawing a card for the computer player.

Algorithm Code:

```
while(round < 12 && (hum || com)){
      System.out.println("My cards are " + computer.seeCard(computer.position));
      System.out.println("Your cards are " + human.seeCard(human.position));
       System.out.println("\nDo you want to add a card? (Type nubmer 0 to get a card, 1 to
wait for next round, 2 to stop drawing)");
      int add = input.nextInt();
      if(add == 0){
        human.addCard(myDeck.drawCard());
        indexHum++;
      }else if (add == 2){
        hum = false;
      }

      if(computer.getSum() <= 15){
        System.out.println("\nI will draw a card\n");
        computer.addCard(myDeck.drawCard());
        indexCom ++;
      }else{
        System.out.println("\nI will not draw a card anymore");
        com = false;
      }
      round ++;
   }
```

2d.

Abstraction is implemented in the construction of the Card class. The attribute of facing upward or downward is abstracted as a boolean variable. The boolean variable is used in the display format of the Card value and also the function flip( ). The Card class extracts the main properties and actions for a real card in a real Blackjack game and it acts like a mould for the creation of cards needed in the main execution program.

Abstraction Code:

```
public class Card{
   private int value;
```

```
private int shape;
private boolean faceDown;

public static final int HEART = 0;
public static final int DIAMOND = 1;
public static final int CLUB = 2;
public static final int SPADE = 3;
public static final int JOKER = -1;

public Card(){
  value = 0;
  shape = 0;
  faceDown = true;
}

public Card(int value, int shape){
  this.value = value;
  this.shape = shape;
  faceDown = true;
}

public void flip(){
  faceDown = !faceDown;
}

public int getValue(){
  return value;
}

public String toString(){
  String valueS = "";
  if(value == 11)
    valueS = "J";
  else if(value == 12)
    valueS = "Q";
```

```
      else if(value == 13)
        valueS = "K";

      String shapeS = "";
      if(shape == HEART)
        shapeS = "H";
      else if(shape == DIAMOND)
        shapeS = "D";
      else if(shape == CLUB)
        shapeS = "C";
      else if(shape == SPADE)
        shapeS = "S";
      else if(shape ==JOKER)
        shapeS = "J";


      String temp = "";
      if(faceDown){
        temp = "XX";
      }else if(value > 11){
        temp = valueS + shapeS;
      }else{
        temp = "" + value + shapeS;
      }
      return temp;
    }
}
```