TCP1201 Objected-Oriented Programming and Data Structures

Assignment 2 (20%)

Trimester 2, Session 2020/2021
Faculty of Computing and Informatics
Multimedia University

DUE DATE: 14 March 2021, 11:59pm

1. FORWARD

- This is a group assignment with maximum of **three (3)** students per group.
- **Start you work early**. No extension of deadline.
- Do not submit late. Submit even if it is incomplete. Make sure it is **runnable**.
- **Interview** will be conducted. All group members must attend the interview.
- **Do not share your code** with any other group. If detected all parties involved may get zero marks.

2. TASKS

You are developing a card game that requires one deck of 52 cards as shown below:

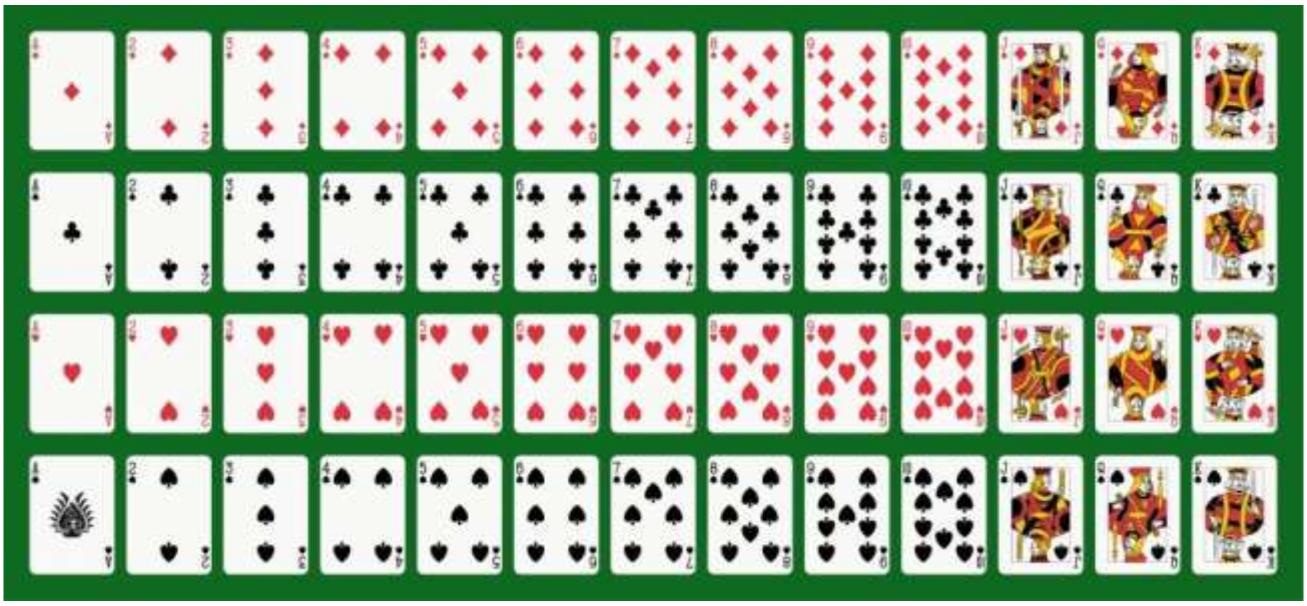

Photo Credit: Mannaggia / Fotolia

The 52 card has 4 suits from top to bottom: diamonds (d), clubs (c), hearts (h), and spades (s). Each card has a point as shown in the table below.

Face	Point
Ace/A	1
2 – 9	Follow the face
10/X	10
Jack/J	10
Queen/Q	10
King/K	10

2.1 Game Rules

1. The game has two (2) phases: 3 Players, and 2 Players.
2. The game begins with the 3 Players phase. Three (3) rounds are played in the 3 Players phase. The top 2 players with the highest score proceed to the 2 Players phase. Four (4) rounds are played in the 2 Players phase.
3. At the end of the 2 Players phase, the player with the highest score wins the game.
4. The 52 cards are shuffled and can be re-shuffled at the beginning of each phase in the game.
5. In each round, players are each dealt 5 cards. As shown in the sample run, Ali receives the 5 cards in the following order
   `Ali: c5 s6 sK dA cK`
   The program shall sort the cards by suit then face.
   `Ali: c5 cK dA s6 sK`
6. In each round, the player with the highest point wins that round. If there is a tie, both players win.
7. The point is chosen from the suit that gives the highest point (study the sample run).
8. At the end of the game, the player with the highest score is declared the winner.

2.2 Sample run

To make testing easier and save time during interview, your program shall **never clear screen**.

```
******************
* 3-Player Phase *
******************
Enter player 1 name: Ali
Enter player 2 name: Baba
Enter player 3 name: Cindy

Available Cards:
Ali  : c5 s6 sK dA cK, h5 h3 dJ d8 s7, cX c2 h4 hA d2, hJ hX s2
Baba : d4 h7 c4 cQ sA, d5 s3 d3 h2 h8, c9 hK d6 sJ sX, s8 d7
Cindy: cA dX h6 dQ d9, c8 h9 hQ sQ cJ, dK c6 s9 s4 c7, s5 c3

Press S to Shuffle or ENTER to start.
<Enter is pressed>


*** ROUND 1 ***
Cards at Hand:
Ali  : c5 cK dA s6 sK | Point = 16
Baba : c4 cQ d4 h7 sA | Point = 14
Cindy: cA d9 dX dQ h6 | Point = 29 | Win
```

```
Score:
Ali   = 0
Baba  = 0
Cindy = 29

Available Cards:
Ali  : h5 h3 dJ d8 s7, cX c2 h4 hA d2, hJ hX s2
Baba : d5 s3 d3 h2 h8, c9 hK d6 sJ sX, s8 d7
Cindy: c8 h9 hQ sQ cJ, dK c6 s9 s4 c7, s5 c3

Press ENTER to next round.


*** ROUND 2 ***
Cards at Hand:
Ali  : d8 dJ h3 h5 s7 | Point = 18
Baba : d3 d5 h2 h8 s3 | Point = 10
Cindy: c8 cJ h9 hQ sQ | Point = 19 | Win

Score:
Ali   = 0
Baba  = 0
Cindy = 48

Available Cards:
Ali  : cX c2 h4 hA d2, hJ hX s2
Baba : c9 hK d6 sJ sX, s8 d7
Cindy: dK c6 s9 s4 c7, s5 c3


*** ROUND 3 ***
Cards at Hand:
Ali  : c2 cX d2 hA h4 | Point = 12
Baba : c9 d6 hK sX sJ | Point = 20 | Win
Cindy: c6 c7 dK s4 s9 | Point = 13

Score:
Ali   = 0
Baba  = 20
Cindy = 48

Available Cards:
Ali  : hJ hX s2
Baba : s8 d7
Cindy: s5 c3

***** Baba and Cindy proceed to 2-Player phase *****


******************
* 2-Player Phase *
******************

Available Cards:
```

```
Baba : hQ dX h7 c2 cX, h4 d6 c5 s5 h9, cA c7 d7 d8 d9, cK c3 dA c6 d4,
       c4 s3 hA hX dK, dQ
Cindy: h6 hJ c8 sJ hK, s4 h5 dJ s7 sA, h3 d5 d3 sQ c9, d2 s8 h8 s9 cQ,
       cJ h2 s2 s6 sX, sK

Press S to Shuffle or ENTER to start.


*** ROUND 1 ***
Cards at Hand:
Baba : c2 cX dX h7 hQ | Point = 17
Cindy: c8 h6 hJ hK sJ | Point = 26 | Win

Score:
Baba  = 20
Cindy = 74

Available Cards:
Baba : h4 d6 c5 s5 h9, cA c7 d7 d8 d9, cK c3 dA c6 d4, c4 s3 hA hX dK, dQ
Cindy: s4 h5 dJ s7 sA, h3 d5 d3 sQ c9, d2 s8 h8 s9 cQ, cJ h2 s2 s6 sX, sK

Press ENTER to next round.


*** ROUND 2 ***
Cards at Hand:
Baba : c5 d6 h4 h9 s5 | Point = 13 | Win
Cindy: dJ h5 sA s4 s7 | Point = 12

Score:
Baba  = 33
Cindy = 74

Available Cards:
Baba : cA c7 d7 d8 d9, cK c3 dA c6 d4, c4 s3 hA hX dK, dQ
Cindy: h3 d5 d3 sQ c9, d2 s8 h8 s9 cQ, cJ h2 s2 s6 sX, sK

Press ENTER to next round.


*** ROUND 3 ***
Cards at Hand:
Baba : cA c7 d7 d8 d9 | Point = 24 | Win
Cindy: c9 d3 d5 h3 sQ | Point = 10

Score:
Baba  = 57
Cindy = 74

Available Cards:
Baba : cK c3 dA c6 d4, c4 s3 hA hX dK, dQ
Cindy: d2 s8 h8 s9 cQ, cJ h2 s2 s6 sX, sK

Press ENTER to next round.
```

```
*** ROUND 4 ***
Cards at Hand:
Baba : c3 c6 cK dA d4 | Point = 19 | Win
Cindy: cQ d2 h8 s8 s9 | Point = 17

Score:
Baba  = 76
Cindy = 74

Available Cards:
Baba : c4 s3 hA hX dK, dQ
Cindy: cJ h2 s2 s6 sX, sK

***** Baba is the WINNER! *****
```

2.3 Class Design

Your program shall have the following 3 classes at minimum:

1) Game – A game has players and cards.
2) Player – A player has cards.
3) Card

You may add classes to support inheritance, association, aggregation, composition, or any significant features.

2.4 Data Structures

You program should use more than array and array list. Consider whether stack, queue, set, and/or map could be used.

3. RECOMMENDED TASK DISTRIBUTION

If working separately is inevitable, the group shall first agree on how the cards and players be represented in the program, and what data structures to be used. Meet often for discussion.

3.1 Without Graphical User Interface (GUI)

1) A member programs the sorting of the 5 cards at a player, and the calculation of highest points (you my start by randomly select 5 cards so that you don't have to wait for other members to complete their part).
2) A member programs the shuffling of cards, the removal of the played cards from Available Cards, and identification of the winner in a round (you may start by selecting the correct winner from 3 randomly generated points so that you don't have to wait for the other members to complete their part).
3) A member programs the input of player names, the sum of score, and the overall flow of the game, i.e. 3 rounds in 3 Players phase and 4 rounds in 2 Players phase (you may start by randomly selecting a winner with a random point for each round so that you don't have to wait for the other members to complete their part).

3.2 With Graphical User Interface

- A member programs the sorting of the 5 cards at players and the calculation of highest points.
- 2 members work closely to program the GUI.

4. SUBMISSION

i. Java **source code**. Make sure the code can be compiled and run.
ii. A file named **GroupX.txt** where X is the **group number** registered in MMLS. The file shall list down the group members' id, name, and contribution.

```
Member 1 Id Name
Member 2 Id Name
Member 3 Id Name

TASK DISTRIBUTION
1. Sorting and determining the highest point
2. Dealing of cards and identify the winner in each round
3. Sum of score and the overall flow of the game
4. Others (specify)
```

Zip the files above and label it in the following format:

TTX_MMLSGroupNumber.zip

For example:

TT1L_13.zip

Mark Sheet (20%)

Criteria	Items (Mark for an item is awarded if it works and student can explain)
1. Program Execution (14 marks)	**1.1. Correct program features and output (12m)** i. Get player names and display them in the game. (1m) ii. Available Cards are evenly distributed among all players. (1m] iii. Can shuffle all 52 cards at the beginning of each phase. (1m) iv. The 5 cards at each player are sorted by suit then face. (1m) v. Determine the highest point correctly based on suit. (1m) vi. Identify the winner for each round. (1m) vii. Track the score for all players. (1m) viii. Played cards are removed from Available Cards for all players. (1m) ix. Complete playing 3 rounds in 3 Players phase. (1m) x. Identify the 2 winners in 3 Players phase. (1m) xi. Complete playing 4 rounds in 2 Players phase. (1m) xii. Identify the winner in 2 Players phase. (1m)
	1.2. User friendliness (2m) 2.0m – Input and output are clear without ambiguity. 1.5m – One input or output are unclear with ambiguity. 1.0m – Many input or output are unclear with ambiguity. 0.5m – The program is unusable.
2. OOP Design and Data Structures (4 marks)	**2.1. Association, Aggregation, and/or Composition (1m)** 1.0m – Good design (sensible modeling with good identifier name) 0.5m – Poor design (insensible modeling or poor identifier name) 0.0m – Do not use
	2.2. Inheritance (1m) JavaFX, Swing, or Android is considered to have included inheritance.

	1.0m – Good design (sensible modeling with good identifier name) 0.5m – Poor design (insensible modeling or poor identifier name) 0.0m – Do not use
	2.3. Implement Comparable or Comparator Interface to sort cards at players (1m) 1.0m – Sort by suit AND face correctly 0.5m – Sort by suit OR face only 0.0m – Do not implement
	2.4. Use appropriate data structures (1m) Use more other than array and array list. 0.5m – Use stack or queue 0.5m – Use a set or map
3. Bonus (2 marks)	2m – JavaFX/Swing/Android for the whole game. For partial GUI, 0.5m – JavaFX/Swing/Android for getting player names. 0.5m – JavaFX/Swing/Android for displaying the 5 cards at the hand of each player. 0.5m – JavaFX/Swing/Android for displaying the score of all players.
4. Presentation & Interview (2 marks)	2.0m – Very smooth and well prepared 1.5m – Overall fine with minor issues 1.0m – Average 0.5m – Poorly prepared or has major issues
5. Late submission, not attending interview, or plagiarism	0 mark for this assignment