Project: The Hearts Game

Expected Effort (approximate):	*40 hours per person*
Due Dates:	*Week 14*
Weightage:	*20% of final grade*

Final Project Submission (20 marks)

Project Review	**Marks**

Analysis & Design	**5**
▪ Design class diagram for your application	
▪ Object-oriented design considerations when developing your application	
▫ Design principles/patterns that you have applied. Use specific examples to illustrate usage.	
▫ The various approaches when tackling design and why you choose one approach over the other	
● [Optional. Good to have!] Use case diagrams & Sequence diagrams to explain your design.	

Code	**10**
▪ Use Git continuously & consistently.	
○ Use the private repository provided. The use of public repositories to host your code will be severely penalized.	
○ Each commit to the repository should have a useful commit message that describes what you changed.	
○ Follows the directory structure specified under Professionalism	
▪ Relates to class design	
▪ Clean Code (http://blue.smu.edu.sg/cleancode)	
▪ Modularized with multiple classes and methods	
▪ Adherence to Java coding convention (http://www.oracle.com/technetwork/java/codeconventions-150003.pdf)	
▫ Code are appropriately documented: program comments (to explain algorithm) and Javadoc comments (for generating API documentation)	
▫ Meaningful names for variables, classes, methods	
▫ Indentation (Use tabs or spaces consistently)	
▪ Good Junit test cases that covers the important logic in the application	

Additional Functionalities (High effort, Low rewards)	**5**
▪ Graphical User interface (e.g. Java Swing. Or JavaFx)	2
▪ Client-Server Application (that supports up to 4 human players in a game)	3

Penalty	
▪ Up to 10% penalty of your total marks if the application has any usability issues (include input validations)	
▪ Up to 50% penalty of your total marks if any of your Java source files does not compile or run.	

Total	**20**

Distribution of Work

- Each member of the team is expected to be familiar with all aspects of the deliverables submitted.
- Your team may be selected for an interview at the end of the term to walkthrough your team's deliverables with us.

Peer Evaluation

- If you have any teaming issues, do highlight to your instructors early and not wait till the peer evaluation.
- You will be awarded a 2-point (of final grade) penalty if you fail to complete the peer evaluation before the deadline.
- There is one team grade for the project. However, your individual grade can be higher or much lower than the team grade based on peers' evaluation (and instructors' evaluation based on the team's interview and observation across the semester).
- We will severely penalize free-loaders, as well as team members who decide to complete the project alone.
- Guidelines
  - Please be frank when evaluating your peers (Do not lie).
  - Do the evaluation seriously. You should not give everyone in the team the same score.
  - You must use the comments field to explain why you think a particular individual deserves the score. Note that the system will not force you to add comments, but we require it. You need to input comments about every member in your team for every metric. Be frank and honest with your comments.
  - Your team peer evaluation score will not be compared against other teams' scores. Having a lower team score than another team will not affect your final grade.
- You are expected to evaluate yourself and your team members based on 4 criteria on a scale of 1 – 25.
  - Commitment
  - Working with others
  - Quantity of work
  - Quality of work

The grading rubric is:

	Evaluation Scale (1 – 25)			
	1 – 8	**9 - 15**	**16 - 21**	**22 - 25**
Commitment	Often missed meetings (>75% of total meetings).	Missed one or two meetings without reasons and failed to inform members about absence prior to the meeting.	Missed one to two meetings with valid reasons and informed team about absence prior to the meeting.	Turned up consistently for meetings. Initiated and set agenda for meetings.
	Frequently late for meetings by more than 30 minutes or left early after a short duration of time meetings.	Frequently late by 5-10 minutes and occasionally late for more than 30 minutes. Occasionally left early.	On time for meetings. Occasionally late for meetings by 5-10 minutes. Stayed for the whole duration of meetings.	Never late for meetings. Stayed for the whole duration of meetings.
	Little or no contribution to discussions and activities.	Some contributions to discussions and activities.	Consistently contributed to all discussions and activities.	Led the team in discussions and activities. Willing to follow when others took initiative to lead.
Working with others	Unwilling to interact with others. Inflexible and negative to new ideas suggested.	Usually helpful and receptive towards new ideas suggested by others. At times, insisted on own way of doing things or implemented ideas without team's consensus.	Helpful and receptive towards new ideas suggested by others. A pleasure to work with.	Expressed ideas clearly, honestly and with respect for others and their work. Willing to listen without arguing every point.
	Never bothered to contribute any ideas to the team.	Contributed some good ideas to the team.	Constantly suggested new ideas to the team. Not all the ideas were applicable though.	Consistently suggested new ideas or was able to build on ideas suggested by others and the all of them were good or creative.
Quantity of work	Did less than their fair share of load.	Contributed a fair share of the team's load. However, did more of a specific type of task only (e.g. coding)	Contributed a fair share of the team's load in all aspects.	Contributed a fair share of the team's load in all aspects. Enthusiastically took on responsibilities in various tasks areas. Coached others in the team about how to get their work done.
Quality of work	Work often had flaws and omissions that needed fixing.	Acceptable quality of work that needed frequent minor fixes or the occasional rework.	Good quality of work with occasional minor fixes.	High quality of work that went well beyond expectations.

Professionalism

- All submissions are to be done via GitHub. All submissions via email will be ignored.

- The repository should contain the following folders and files:
  - A *teamX.pdf* containing all the analysis and design, and project management artifacts. Only 1 single PDF file is to be submitted; all other files (e.g. JPEG and VSD files will be ignored). Your PDF file should:
    - be free of hand-written, or hand-drawn content
    - include page numbers and an accurate table of content
    - use decent margin and readable font sizes of at least 10-11 points
    - Visible diagrams and practice good diagramming guidelines (http://www.agilemodeling.com/style/general.htm). Diagrams in your PDF file which are not visible will be ignored.

  - compile.bat and run.bat batch files
    - The compile.bat file compiles all your Java source files and automatically stores them in the classes directory.
    - The run.bat runs your application.

    Note: You can assume that the instructors' laptops have the PATH environment variable set correctly so that javac.exe and java.exe can be executed at the DOS command prompt from any directory. The instructors will run compile.bat and run.bat on their laptops during assessment.

  - ***src***
    This directory contains all your Java source files.

  - **classes**
    This directory should be left empty during submission. After compile.bat runs, the class files will be stored here automatically.

  - **docs**
    This directory contains the generated javadocs of your application.

  - **images**
    This directory contains any image files used by your application. It will be empty if you build a console application.

  - **lib (Only if using any external libraries)**
    This directory contains any jar files that you use for your application.

- Only the use of external free or open source libraries is allowed. No other form of code-sharing is allowed. Co-development of code with members from other teams is a definite NO-NO.

Project Functional Requirements

You are to write the hearts card game.

The rules of the game are:

1. There are 4 players — the user controlling one hand and the computer controlling the other three hands. The objective of the game is to win by collecting the fewest point cards.

2. The standard 52-card deck is used.
    a. The rank of the cards are, in descending order: Ace (A), King (K), Queen (Q), Jack (J), 10, 9, 8, 7, 6, 5, 4, 3, 2.
    b. The suits of the cards are clubs(♣), diamonds(♦), spades(♠) and hearts(♥).

THE DEAL

3. The cards are shuffled and distributed to each of the players in sequence. Each player is dealt 13 cards.

THE PASS

4. After looking at their hand, a player selects any three cards and passes them face down to the player on their left. The player must pass their three cards before looking at the cards received from the player on their right.

5. The passing rotation in a 4-player game is:
    I. to the player on your left,
    II. to the player on your right,
    III. to the player across the table,
    IV. no passing

    The rotation then repeats until the game ends.

6. The computer will adopt a simple strategy of picking of the highest card of any suit.

7. The three computer players are independent. At no point should they "collaborate" with each other and cause the computer player to lose.

THE ROUND ("TRICK")

8. The player holding the 2 of clubs (♣) after the pass will play the first card. We say that the leading suit is clubs(♣).

9. Each player must follow suit (i.e. discard a card of the same suit).

10. For the first round:
    a. No player can play a card that is a point card i.e. Queen of Spades or any heart card.
    b. If a player lacks a card of the leading suit, a card of any other suit (with the exception of point cards) may be discarded. For example, if the person who is leading the round places a 2♦, the person to the left will
        i. play a diamond card (e.g. 3♦) if available.
        ii. otherwise, discard any card that is not a point card (e.g. 4♣).

11. For the second round or later:

   a. If a player lacks a card of the leading suit, a card of any other suit may be discarded. If the player discards a point card, then the constraint of leading a round with a ♥ card is broken. This is known as "breaking hearts". Following this, any player may lead a round with ♥ cards.

12. The highest card of the suit led wins a "trick" and the winner of that trick leads next. The winner of the trick collects the cards played in the round (and places it face down to form a neat "book" or stack of cards)

13. The computer will adopt the following strategy:
    a. If it is the first player in the round, pick the card of the smallest rank that adheres to the game rules.

    b. Otherwise,
        i. If it is not the last player, pick the largest rank card it owns of the same suit that does not exceed the largest card of the leading suit discarded whenever possible
        ii. If it is the last player, pick the largest card.

    c. If it is void of the suit led, then pick the card of the highest rank (refer to point 2) available that adheres to the game rules.

THE SCORE

14. The scoring is:
    a. The queen of spades(♠) will get 13 points.
    b. Any card of suit heart(♥) will get 1 point.

15. However, if one player has won all 13 hearts and the queen of spades, this is termed "shooting the moon".
    a. The player holding all the point cards will get 0 points.
    b. Every other player will get 26 points.

16. It does not matter how many rounds a player won. The scoring is based solely on who wins tricks containing hearts and/or the queen of spades.

17. The game is ends when one player hits 100 points . The player with the lowest score wins.

References:

1. Images of all the cards: https://www.waste.org/~oxymoron/files/cards/

2. A list of commonly used classes in Card games (Card, Deck, Hand, Rank, Suit). You are welcome to reuse them: https://github.com/BigRedS/java/tree/master/CardGame/src

3. How to write good JUnit test cases
   http://www.maheshsubramaniya.com/article/junit-best-practices-how-to-write-good-junit-test-cases.html

4. Some references on playing Hearts:
   a. https://www.youtube.com/watch?v=RSr_89K_65c
   b. https://www.youtube.com/watch?v=qzk1eOc5cns
   c. https://worldofcardgames.com/hearts-card-game-rules.html

5. Intellij IDEA: https://www.jetbrains.com/student
   This is a good IDE to use when creating GUI applications.