**Project: Card Game**

| **Expected Effort (approximate):** | *40 hours per person* | |
| --- | --- | --- |
| **Weightage:** | *20% of final grade* | |

| **(Tentative) Grading Components** | **Marks** |
| --- | --- |
|  |  |
| **Code** | **10** |
| * The code is modularized with multiple classes and methods. * The code is clean. <http://smu.sg/cleancode> . * The code is designed and layered to known design principles. For example, Single Responsibility Principle (SRP). * The configuration parameters are externalized. [https://reflectoring.io/externalize-configuration](https://reflectoring.io/externalize-configuration/) * Adherence to Java coding convention.   <http://www.oracle.com/technetwork/java/codeconventions-150003.pdf>   * + The code is appropriately documented.   + The variable, class, and method names are meaningful.   + The code is indented using tabs or spaces consistently. |  |
|
|
|
|  |  |
| **Application** | **5** |
| * Usability of your application. It can be a Java console, GUI or web application.   + Nielsen’s 10 heuristic evaluation <https://www.nngroup.com/articles/ten-usability-heuristics/> * Submission - source code + readme file |  |
|  |  |
| **Presentation** | **5** |
| * Time allocation:   + 10 minutes for presentation & demonstration.   + 10 minutes for Q & A. * Presentation Content:   + State your team's object-oriented design.   + List all the open source libraries used. If you evaluated a few, state why you picked that particular library.   + How the team designed the application. Any design principle(s) or considerations?   + Demonstrate the features of your client application. * Presentation is well-rehearsed.   + Slides are clean and visible.   + There is no unnecessary switching of laptops/applications.   + The transition is smooth. |  |
|  |  |
| **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. |  |

**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 the 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 the 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 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 **eLearn > Assignments**. All submissions via email will be ignored.
* Submission deadline is at 11:59pm, 1 week before your presentation (aka class) day.
* No commit is allowed after the deadline.
* The zip file should contain the following folders and files:
  + A *GX-TY.pptx* of your presentation slides.
* **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.

* + **images**

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

* + **lib (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.

**Presentation**

* During week 13 class.
* Attendance: Presentation schedule will be released nearer to date.
  + Every team member is to be present for their team’s allocated schedule time slot only.
  + Be 5 minutes earlier so as to minimize delays to the later teams.
* Up to 15 minutes per team plus a 5 minutes Q&A.

**Project Requirements**

Implement any card game with the following requirements:

1. It should allow 2 or more players.
2. The standard 52-card deck is used. You can include Joker cards if your game requires it.
   1. 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.
   2. The suits of the cards are clubs(♣), diamonds(♦), spades(♠) and hearts(♥).

**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. Intellij IDEA: [https://www.](https://www.jetbrains.com/student/)jetbrains[.com/student](https://www.jetbrains.com/student/)

This is a good IDE to use when creating GUI applications (if you want to).