Group Project
Group 2

Course: SYST17796

Class: 1241_58189

Contents

Team Contract

SYST 17796 TEAM CONTRACT

Team Name: Group 2

Please negotiate, sign, scan and include as the first page in your Deliverable 1.

Please note that if cheating is discovered in a group assignment each member will be charged with a cheating offense regardless of their involvement in the offense. Each member will receive the appropriate sanction based on their individual academic integrity history.

Please ensure that you understand the importance of academic honesty. Each member of the group is responsible to ensure the academic integrity of all of the submitted work, not just their own part. Placing your name on a submission indicates that you take responsibility for its content.

For further information, read Academic Integrity Policy here :

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Sim Yuan Hee	SYH	991704213
Annu Annu	AA	991717839

https://caps.sheridancollege.ca/student-guide/academic-policies-and-procedures.aspx

By signing this contract, we acknowledge having read the Sheridan Academic Integrity Policy

Responsibilities of the Project Leader include:

- Assigning tasks to other team members, including self, in a fair and equitable manner.
- Ensuring work is completed with accuracy, completeness and timeliness.
- Planning for task completion to ensure timelines are met.
- Notifying the professor of any issues in a timely manner so that corrective measures can be taken.
- Any other duties as deemed necessary for project completion.

What we will do if . . .

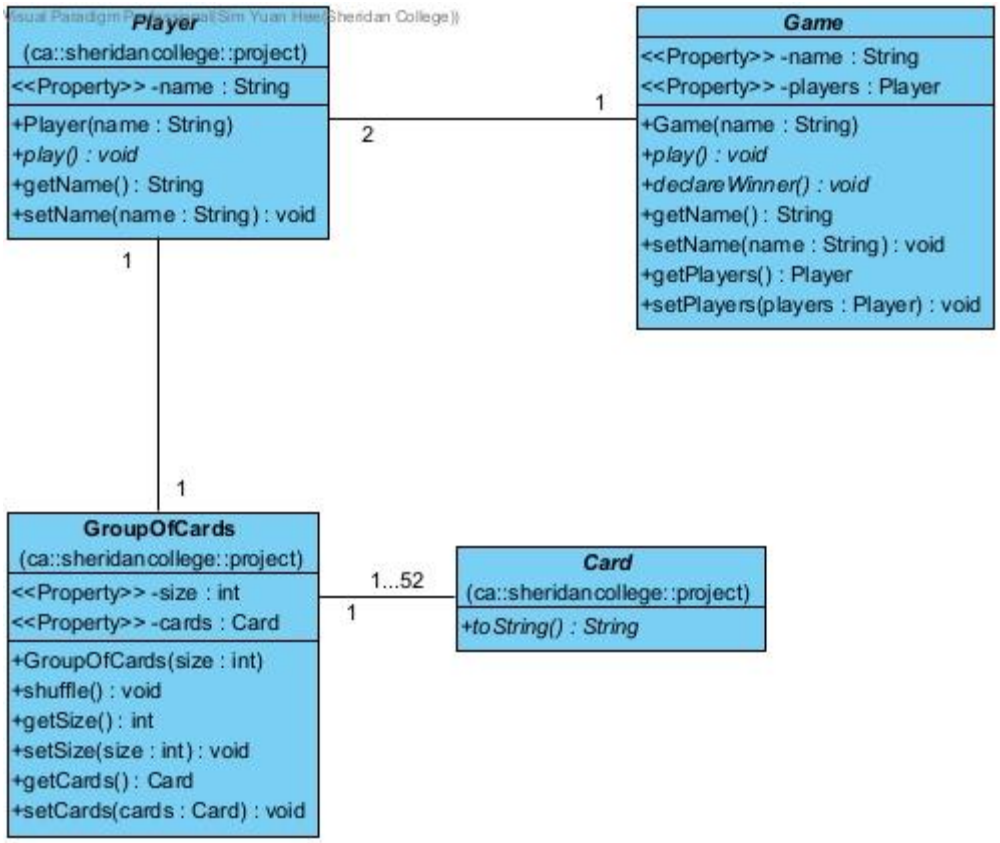
Scenario	**Accepted initials**	**We agree to do the following (Put an X corresponding to your choice in each box)**
Team member does not regularly attend team meetings and/or does not respond to communications in a timely manner.	AA SYH	Project leader emails the student citing the concerns and cc's the professor so they are aware of the situation at the very onset _X_ (**Mandatory**). a) __ In addition to above, the leader/team will (add your own content here):
Team member does not deliver component on time due to severe illness or extreme personal problem.	AA SYH	a) Team absorbs workload temporarily _ b) Team seeks advice from professor __ c) Team shifts target date if possible __ d) __ Other (specify):

Scenario	Accepted initials	We agree to do the following (Put an X corresponding to your choice in each box)
Team member has difficulty delivering component on time due to lack of understanding or ability.	AA SYH	a) Team reassigns component __ b) Team helps member __ c) Team member must ask professor for help __ d) __ Other (specify):
Team member does not deliver component on time due to lack of effort.	AA SYH	a) Team absorbs workload __ b) Team member(s) ask professor to request a Participation Form from <u>all</u> team members. This *may* result in individualized grades being awarded for a deliverable __ c) Both a. and b. above __ d) __ Other (specify):
Team cannot achieve consensus leaving one or more member(s) feeling that their voice(s) is/are not being heard in a decision which affects everyone.	AA SYH	a) Team agrees to abide by majority vote __ b) Team seeks advice from the professor __ c) __ Other (specify):

Scenario	Accepted initials	We agree to do the following (Put an X corresponding to your choice in each box)
Team members do not share expectations for the quality of work on a particular deliverable.	AA SYH	a) Team members will draw on each other's strengths to help bring the quality of the deliverable to a minimal acceptable level __ b) Team votes on each submission's quality __ c) Team member(s) ask professor to request a Participation Form from all team members, which may result in individualized grades being awarded for a deliverable __ d) __ Other (specify):
Team member behaves in an unprofessional manner, e.g. being rude, uncooperative and/or making one or more member(s) feel uncomfortable.	AA SYH	a) Team agrees to avoid use of all vocabulary inappropriate to a business/college setting __ b) Team attempts to resolve the issue by airing the problem at a team meeting __ c) Team requests a meeting with the professor to discuss further __ d) __ Other (specify):
There is a dominant team member who insists on making all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members	AA SYH	a) Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote __ b) Team will express subordination feelings and attempt to resolve issue __ c) Team seeks advice from the professor __ d) __ Other (specify):

Scenario	Accepted initials	We agree to do the following (Put an X corresponding to your choice in each box)
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted	AA SYH	a) Team forces decision sharing by routinely voting on all issues __ b) Team routinely checks with each other about perceived roles __ c) Team discusses the matter at team meeting __

CLASS DIAGRAM FOR PROJECT STARTER CODE



SYST 17796 DELIVERABLE 1

DESIGN DOCUMENT TEMPLATE

OVERVIEW

1. Project Background and Description

Our project aims to develop a Card game, which is War. This game is played with a deck of 52 cards. The deck is divided equally, one at a time, face down and the players are not allowed to see the cards. Each player places their stack of cards face down, in front of them. The game starts and each player turns up a card and the player with the higher card wins and collects all the cards revealed during the round and puts them, face down, on the bottom of stack. If there is a tie, it is War, then each player faces down three cards and face-up one card and the player having the higher card wins all the cards. This game continues until a player has won all the 52 cards or a specified number of rounds are completed.

Starting Base Code Description-

The base code consists of four classes Card.java, Game.java, GroupOfCards.java, and Player.java.
1. Card.java:
This abstract class represents a playing card and contains an abstract ‘toString()’ method.
2. GroupOfCards.java:
This class has information about managing groups of cards like making a list of cards, shuffling the cards, and also has a getter and setter for the size of the group of cards.
3.Player.java:
This abstract class represents the players and contains a constructor, getter, setter and an abstract method ‘play()’
4. Game.java:
This abstract class has the instance variables for the game’s title and the list of the players. This class provides methods to play the game and has an abstract method ‘declareWinner()’ to declare the winner of the game.

The base code follows object-oriented programming concepts like inheritance and abstraction. This base code can be used to make different games related to cards. In our project, we will be using it to make a ‘War’ card game. The base code follows the modular design principles because classes are doing different works, and a class is responsible for only one thing so there is high cohesion. And loose coupling can be achieved when the classes are not too much dependent on each other.

2. Project Scope

There are two members in our group - Annu Annu and Sim Yuan Hee.

Both the members will work on the code together. Annu will collaborate with Sim to implement the codebase. Annu will be ensuring the implementation of object-oriented Programming principles and best practices. Sim will be responsible for creating the class diagram and testing the game.

The project goal is to develop an application for a war card game, allowing users to play the game. The program mainly requires input validation, decision structure, and data collection which includes shuffling and dealing with the cards, displaying the cards, comparing the cards from the players, and declaring the winner.

Firstly, the interface will introduce the game rules. Then, it prompts the player to enter the name and on starting the game it allows to reveal the top card for each player and announce the result of each round. If there is a tie, it will print war. The number of cards held by each player will be displayed and the winner will be announced at the end of the game.

Sim will test the game to ensure functionality and usability. The project will be completed when the game is fully functional and bug-free. The game rules are implemented accurately, and classes exhibit high cohesion and loose coupling and flexibility. Through testing has been conducted, and identified bugs have been addressed, the project will be completed successfully.

3. High-Level Requirements

The new system must include the following:

- Ability for each player to register with the game
- Ability to validate the user's input
- Players are required to have unique usernames
- Players should be able to reveal the top card of their stack and compare them with other cards of that round.
- Ability of the game to determine the winner of each round based on written comparisons.
- Ability for players to know their status (score) at all times.

4. Implementation Plan

Git repository URL: https://github.com/Sim-Yuan-Hee/Group_2_SYST17796_1241_58189
Developers are required to check the code once a week and add commits to ensure regular updates and if required branches can be made to make sure that the changes work properly before merging them to the main branch.

The directory structure for the project->

Group_2_SYST17796_1241_58189 – name of the project

UML Class Diagram – folder for the class diagram and contains the vpp file that has the class diagram for the project starter code.

nbproject - a folder that contains several files with .xml and .properties extensions.

src/ca/sheridancollege/project – folder that contains the four classes Card.java, Game.java, GroupOfCards.java, Player.java

For the coding standards, group members will make use of indentation, proper formatting and will add comments for a better understanding of the code.
The tools we will be using are NetBeans as the IDE and Visual Paradigm for the class diagrams and GitHub will be used to make the remote repositories for our project.

5. Design Considerations

- Encapsulation – Player.java and GroupOfCards are the encapsulated classes present in the starter code for the project because these classes have a constructor, getters, and setters to access the attributes. Card class also needs private attributes and public methods to access them. The player class encapsulates the name attribute and GroupOfCards encapsulates the size. The Card class will encapsulate the attribute like Suit.

- Delegation – This principle is included in all four classes of the program, using abstract leads to delegation for example in the Player.java class, the method play() is abstract and it will be overridden by the subclass of the Player class. Also, in the Game.java class, there is an abstract method declareWinner() which will be implemented by the subclass of the Game.java class. GroupOfCards.java has a method shuffle that makes use of collections to shuffle the card instead of implementing a code by itself so that is also delegation.

- Flexibility/Maintainability – In the base code, different classes are made to perform different tasks and there are abstract methods also present which provide flexibility to our code. By adhering to good coding practices, there will be high cohesion and loose coupling in our code, which will help in the maintainability of the program.