

Team Contract

SYST 17796 TEAM PROJECT

Team Name: Group 9

Please negotiate, sign, scan and include as the first section 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 honesty 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 Honesty Policy on AccessSheridan or visit the faculty office and speak with the Program Support Specialist.

Team Member Names (Please Print)	Signatures	Student ID
Project Leader: Rakesh Kumar		991556750
Sapan Ashokbhai Patel		991576618
Henis Prakashkumar Patel		991555637
Jashan Goyal.	Jashan Goyal..	991544005.

By signing this contract, we acknowledge having read the Sheridan Academic Honesty Policy as per the link below.

<https://policy.sheridanc.on.ca/dotNet/documents/?docid=917&mode=view>

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
- Any other duties as deemed necessary for project completion

What we will do if . . .

Scenario	Accepted Y/N + initial	We agree to do the following
Team member does not deliver component on time due to severe illness or extreme personal problem	Y J.G. Y R.K. Y S.P. Y N.P.	a) Team absorbs workload temporarily <input checked="" type="checkbox"/> b) Team seeks advice from professor ____ c) Team shifts target date if possible ____ d) Other:
Team member cannot deliver component on time due to lack of ability	Y J.G. Y R.K. Y S.P. Y N.P.	a) Team reassigns component ____ b) Team helps member <input checked="" type="checkbox"/> c) Team member must ask professor for reference material ____ d) Other:
Team member does not deliver component on time due to lack of effort	Y N.P. Y J.G. Y R.K. Y S.P.	a) Team absorbs workload ____ b) Team "fires" team member by not permitting his/her name on submission <input checked="" type="checkbox"/>

		c) Other:
--	--	-----------

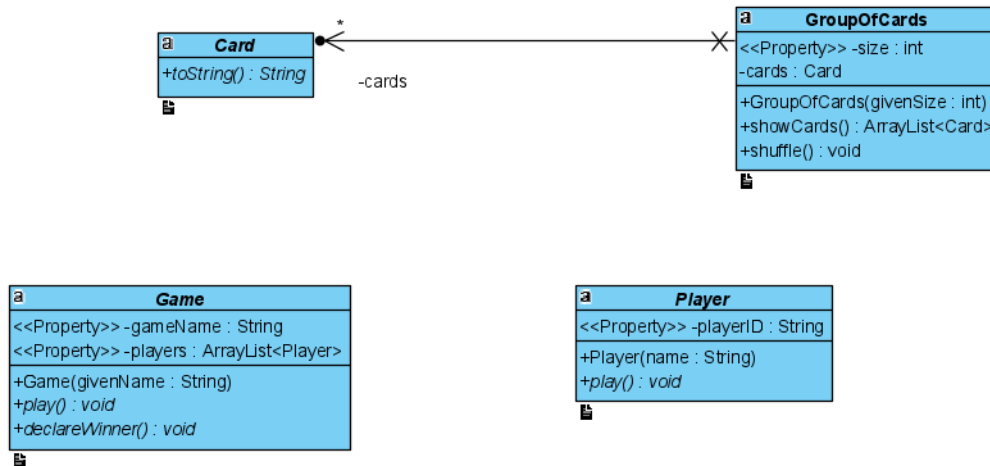
Scenario	Accepted Y/N + initial	We agree to do the following
Team member does not attend team meeting	Y R.K. Y J.G. Y S.P. Y N.P.	a) Team proceeds without him/her and will assign work to the absent member ✓ b) Team doesn't proceed and records team member's absence ____ c) Team proceeds for that meeting but "fires" member after ____ occurrences ____
A piece of production equipment fails such as a printer, disk drive, or laptop	Y J.G. Y R.K. Y S.P. Y N.P.	a) Backup copies will be made and kept in the college ✓ b) A locker or "share" directory will be used for joint access ____ c) A photocopy and duplicate disk of all deliverables will be made ____ d) Other:
An unforeseen constraint occurs after the deliverable has been allocated and scheduled (a surprise test or assignment)	Y J.G. Y R.K. Y S.P. Y N.P.	a) Team meets and reschedules deliverable ____ b) Team will cope with constraint ✓ c) Other:
Team cannot achieve consensus leaving one member feeling "railroaded",	Y N.P. Y J.G. Y R.K. Y S.P.	a) Team agrees to abide by majority vote ✓

<p>"ignored", or "frustrated" with a decision which affects all parties</p>		<p>b) Team flips coin ____</p> <p>c) Other:</p>
<p>Team members do not share expectations for grade desired</p>	<p>Y J.G.</p> <p>Y R.K.</p> <p>Y S.P.</p> <p>Y H.P.</p>	<p>a) Team will elect one person as "standards-bearer" who has the right to ask that work be redone ____</p> <p>b) Team votes on each submission's quality ____</p> <p>c) Team will ask for individual marking and will identify sections by author <u>✓</u></p> <p>d) Other:</p>

Scenario	Accepted Y/N + initial	We agree to do the following
Team member behaves in an unprofessional manner by being rude or uncooperative	Y Rk Y SP Y J.G. Y HP	a) Team attempts to resolve the issue by airing the problem at team meeting <input checked="" type="checkbox"/> b) Team requests meeting with professor to problem-solve ____ c) Team ignores behaviour ____ d) Team agrees to avoid use of all vocabulary inappropriate to the business setting ____
Team member assumes or requests that his/her name be signed to a submission but has not participated in production of the deliverable	Y RK Y SP Y J.G. Y HP	a) Team agrees that this is cheating and is unethical <input checked="" type="checkbox"/> b) Friends are friends and should help each other ____ c) Team will submit with signature but will advise professor who will take action ____
There is a dominant team member who is content to make all decisions on the team's behalf leaving some team members feeling like subordinates rather than equal members	Y RK Y SP Y NA Y J.G.	a) Team will actively solicit consensus on all decisions which affect project direction by asking for each member's decision and vote <input checked="" type="checkbox"/> b) Team will express subordination feelings and attempt to resolve issue ____ c) Other:
Team has a member who refuses to participate in decision making but complains to others that s/he wasn't consulted	Y RK Y S.P Y N.P Y J.G.	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 <input checked="" type="checkbox"/>

Deliverable 1

Starter Code Diagram (Visual Paradigm)



Project background and Description

The project consist of a code that is valid for all the card games which can be played by 52 cards and we will be modifying the code according to the game 'War'.

Here is the description of war game.

Number of Players: 2

Goal:

The goal of the game is that the player should win the whole deck of cards.

Deal:

The deck of 52 cards are divided equally among 2 players. In the beginning one player starts the game and its their choice to choose who will start the game first .

Play:

Every player needs to turn up a card simultaneously and the player with higher position card wins the two cards and put them face down on the base of his stack.

The war starts if the cards are in similar positions. Every player turns up one face down and one face up. The player with the higher position card with all the cards. Also, if the turn-up cards are again the equivalent, every player puts another card face down and another card with face up. The player with the higher card wins all the 10 cards, this game proceeds till the one player wins all the 52 cards. (*War – Card Game Rules*, 2015)

2. Project Scope:

Rakesh Kumar	Group of Cards
Jashan Goyal	Player
Henis Patel	Card
Sapan Ashokbhai Patel	Game

3. High Level Requirements:

- Ability for each player to register with the game
- Ability for the game to communicate a win or loss
- Ability for player to know the score
- There is a player subclass which constitutes and populates the player
- Inheritance
- Loose Coupling
- High Cohesion
- Delegation

4. Implementation Plan

- GitHub Link: <https://github.com/jashangoyal20/Deliverable1.git>
- Use of GitHub: On GitHub each developer can push their code to the git repository and others can check their code to make some changes or correction. Everyone will see the coding done by the others developer and help them to improve their part of code anytime.
- Structure:
 - Project Folder includes the 4 Java files
 - UML Diagram
- NetBeans and VP (Visual Paradigm) use as a Coding Standard

5. Design Considerations

➤ Encapsulation

```
15 public abstract class Game
16 {
17     private final String gameName; //the title of the game
18     private ArrayList <Player> players; // the players of the game
19
20     public Game (String givenName)
21     {
22         gameName = givenName;
23         players = new ArrayList ();
24     }
25
26     /**
27     * @return the gameName
28     */
29     public String getGameName ()
```

```
    public int getSize () {
        return size;
    }

    /**
    * @param givenSize the max size for the group of cards
    */
    public void setSize (int givenSize) {
        size = givenSize;
    }

} //end class
```

➤ Array list

```

    }

    /**
     * @return the players of this game
     */
    public ArrayList <Player> getPlayers()
    {
        return players;
    }

    /**
     * @param players the players of this game
     */
    public void setPlayers(ArrayList <Player> players)
    {
        this.players = players;
    }

```

- Abstraction Classes can be seen in the given starter code. These are Player, Game, Cards.

References

1. *War – Card Game Rules*. (2015). Bicycle Playing Cards. <https://bicyclecards.com/how-to-play/war/>