

# Assignment

*This assignment is the examination of this course. This means that if you finish this assignment properly you will pass the course. When you've done the assignment you can hand it in at [courses.vsw.se/assignments.php](https://courses.vsw.se/assignments.php). When it has been handed in it will be corrected and marked. If everything was fine with your assignment you'll pass the course. If something was wrong you'll get it back together with some feedback about it. Fix whatever that was wrong and hand it in again. You can't really fail a course since you can just hand in new versions of an assignment. Note however that if you send in too many assignments your assignment won't be prioritized when marking it. You should therefore make sure to always double check your code before sending it in.*

You are currently taking a break from whatever you need to take a break from and to relax you play some cards with some friends.

Write a Card class which stores the suit and the value of the card, the suit should be stored as an Enumeration. You should also have an abstract Player class that keeps track of a player's cards as well as their score(how many times they have won). You also need a Deck class that should be able to shuffle as well as reset(refill it) the deck. Both the Player and the Deck class should implement the interface ICardHolder which should contain methods to get the amount of cards and a card at a specific index. This should be used to create a method that prints out the content of an ICardHolder (the cards in a player's hand or the cards in the deck).

The game itself is very simple, each player has a hand of 5 cards. The player that starts, plays a card. The other players also play a card of their choice. The player who had the card with the highest value of the suit of the first card gets a point. The played cards are put back into the deck and the deck is reshuffled. Everyone picks a new card and then the player who got a point starts the next round.

Finally write a class ControlledPlayer and one called BotPlayer where the former gets user input to play where the latter just randomizes which card to play.

The program should be easy to follow, be expandable, well documented, be able to prevent errors and using classes, interfaces and enumerations properly.

## Runtime example

When your program is running it could look like the following. Note that it doesn't have to be exactly the same but it should be able to do similar things. Also note that not all parts of the program is shown in the example. Lines starting with a > means that what comes after it is input by the user and therefore not a part of the program itself.

Round number 1  
Steve has 0 points  
Bot 1 has 0 points

*Bot 2 has 0 points*

*Bot 3 has 0 points*

*Your cards are the following*

- 1. Ace of Diamonds*
- 2. Seven of Clubs*
- 3. Five of Clubs*
- 4. King of Spades*
- 5. Queen of Hearts*

*It's your time to play, please pick a card.*

> 1

*You played Ace of Diamonds*

*Bot 1 played Seven of Diamonds*

*Bot 2 played Ace of Clubs*

*Bot 3 played King of Hearts*

*You won the round. Everyone collects one card.*

*Round number 1*

*Steve has 1 points*

*Bot 1 has 0 points*

*Bot 2 has 0 points*

*Bot 3 has 0 points*

*Your cards are the following*

- 1. Two of Hearts*
- 2. Seven of Clubs*
- 3. Five of Clubs*
- 4. King of Spades*
- 5. Queen of Hearts*

*It's your time to play, please pick a card.*

> 4

*You played King of Spades*

*Bot 1 played Ace of Spades*

*Bot 2 played Ace of Hearts*

*Bot 3 played Three of Spades*

*Bot 1 won. Everyone collects one card.*

*Round number 31*

*Steve has 1 points*

*Bot 1 has 1 points*

*Bot 2 has 0 points*

*Bot 3 has 0 points*

*Your cards are the following*

*1. Two of Hearts*

*2. Seven of Clubs*

*3. Five of Clubs*

*4. Ten of Clubs*

*5. Queen of Hearts*

*Bot 1 played Six of Clubs*

*Bot 2 played Nine of Clubs*

*Bot 3 played Nine of Hearts*

*It's your time to play, please pick a card.*

> 4

*You played Ten of Clubs*

*You won the round. Everyone collects one card.*

*Round number 4*

*Steve has 2 points*

*Bot 1 has 1 points*

*Bot 2 has 0 points*

*Bot 3 has 0 points*