Byoung Mo Lee

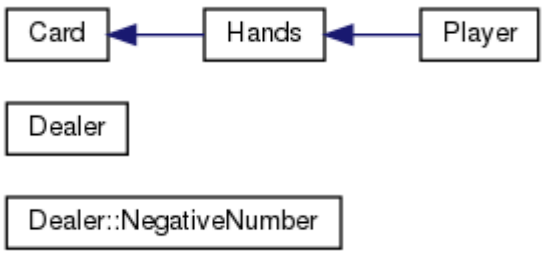
Cross Reference for Project 2

You are to fill-in with where located in code

Chapter	Section	Topic	Where Line #"s	Pts	Notes
13		Classes			
	1 to 3	Instance of a Class	Card.h #15-36 Dealer.h #19~53 Hands.h #19~38 Player.h #21~53	4	
	4	Private Data Members	Card.h #20-28 Dealer.h #19~27 Hands.h #31~40 Player.h #20~28	4	Never Public
	5	Specification vs. Implementation	Card.h vs. Card.cpp Dealer.h vs. Dealer.cpp Hands.h vs. Hands.cpp Player.h vs. Player.cpp	4	.h vs. .cpp files Always split
	6	Inline	Player.h #31,33,34	4	
	7, 8, 10	Constructors	Card.h #31 Dealer.h #31 Hands.cpp #14~17 Player.h #31	4	Overloading
	9	Destructors	Card.h #31 Dealer.h #32 Hands.h #21 Player.h #32	4	
	12	Arrays of Objects	Hands.h #35,#37	4	
	16	UML	doxygen	4	
14		More about Classes			
	1	Static		5	
	2	Friends		2	
	4	Copy Constructors	Hands.cpp #19	5	
	5	Operator Overloading	Cards.h #31	8	Overload 3 operators
	7	Aggregation		6	
15		Inheritance			
	1	Protected members	Cards.h #15~17	6	
	2 to 5	Base Class to Derived	Dealer.cpp #23	6	
	6	Polymorphic associations	main.cpp #85	6	
	7	Abstract Classes		6	
16		Advanced Classes			
	1	Exceptions	Dealer.h #26~36 Main #43~79	6	
	2 to 4	Templates	Templates.h	6	
	5	STL		6	
		Sum		100	

Class Hierarchy

Go to the textual class hierarchy



Class List

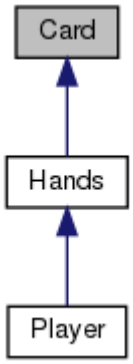
Here are the classes, structs, unions and interfaces with brief descriptions:

[detail level 1 2]

- C **Card**
- ▼ C **Dealer**
  - C **NegativeNumber**
- C **Hands**
- C **Player**

Card Class Reference

Inheritance diagram for Card:



[legend]

Public Member Functions

	Card (int f, int s)
void	**setCard** (int f, int s)
void	**setFace** (int f)
void	**setSuit** (int s)
void	**setACard** (**Card** &c)
int	**getFace** () const
int	**getSuit** () const
void	**displayCard** ()
void	**selectionSortByFace** (**Card** *c, int begin, int size)
void	**selectionSortBySuit** (**Card** *c, int begin, int size)
const **Card**	**operator=** (const **Card** &right)

Protected Attributes

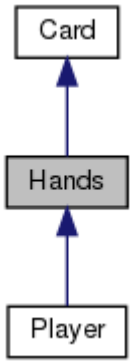
int	**face**
int	**suit**

The documentation for this class was generated from the following files:

- /home/pete/Dropbox/ScanSnap/School/Byoungmo/2019 Spring/CSC17A/git-repos/LeeByoungmo_CSC_17a_44083/Proj/Proj2/texasHoldem_Ver7/**Card.h**
- /home/pete/Dropbox/ScanSnap/School/Byoungmo/2019 Spring/CSC17A/git-repos/LeeByoungmo_CSC_17a_44083/Proj/Proj2/texasHoldem_Ver7/Card.cpp

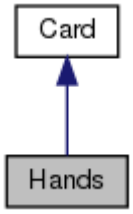
Hands Class Reference

Inheritance diagram for Hands:



[legend]

Collaboration diagram for Hands:



[legend]

Public Member Functions

	Hands (const **Hands** &orig)
void	**getInformation** (**Card** *)
int	**checkStraight** (const int *const *, int)
int	**isFlush** (**Card** *)
void	**checkFaceHands** (**Card** *, int)
void	**setHands** (**Card** *, int)
void	**displayFiveCards** (**Card** *, int)
Card *	**getFiveCards** (**Card** *c, int n)
int	**getHands** (**Card** *c, int n)

Public Member Functions inherited from Card

Additional Inherited Members

Protected Attributes inherited from Card

The documentation for this class was generated from the following files:

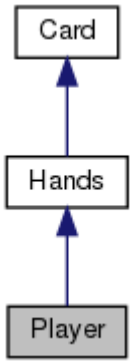
- /home/pete/Dropbox/ScanSnap/School/Byoungmo/2019 Spring/CSC17A/git-repos/LeeByoungmo_CSC_17a_44083/Proj/Proj2/texasHoldem_Ver7/**Hands.h**

- /home/pete/Dropbox/ScanSnap/School/Byoungmo/2019 Spring/CSC17A/git-repos/LeeByoungmo_CSC_17a_44083/Proj/Proj2/texasHoldem_Ver7/Hands.cpp

---

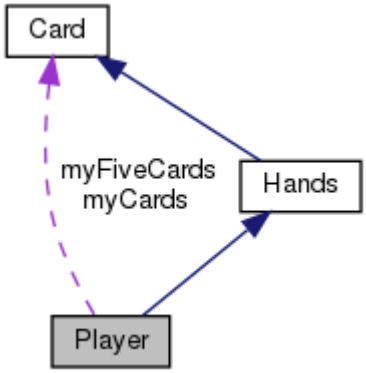
Player Class Reference

Inheritance diagram for Player:



[legend]

Collaboration diagram for Player:



[legend]

Public Member Functions

void	**setPlayer** (string n, int b, int o, int s)
void	**setMyCard** (**Card** *mC)
void	**setName** (string n)
void	**setBalance** (int b)
void	**setOrder** (int o)
void	**setStatus** (int s)
string	**getName** () const
int	**getBalance** () const
int	**getNumCards** () const
void	**updateMyHands** ()
int	**getMyHands** () const
Card *	**getMyFiveCards** () const
int	**getMyPoints** () const
int	**getOrder** () const
Card *	**getMycards** () const

int	**getStatus** () const
void	**putMoneyToPot** (int m)
void	**takeMoneyFromPot** (int m)
void	**resetMyCards** ()
void	**addCards** (Card *c, int n)
void	**displayMyFiveCards** ()

Public Member Functions inherited from Hands

Public Member Functions inherited from Card

Protected Attributes

string	**name**
int	**balance**
int	**order**
int	**status**
Card *	**myCards**
Card *	**myFiveCards**
int	**hands**
int	**myPoints**

Protected Attributes inherited from Card

The documentation for this class was generated from the following files:

- /home/pete/Dropbox/ScanSnap/School/Byoungmo/2019 Spring/CSC17A/git-repos/LeeByoungmo_CSC_17a_44083/Proj/Proj2/texasHoldem_Ver7/**Player.h**
- /home/pete/Dropbox/ScanSnap/School/Byoungmo/2019 Spring/CSC17A/git-repos/LeeByoungmo_CSC_17a_44083/Proj/Proj2/texasHoldem_Ver7/Player.cpp

Dealer Class Reference

Classes

class	**NegativeNumber**

Public Member Functions

	Dealer (int n)
void	**shuffle** ()
void	**setPlayers** (int n)
void	**setCards** (**Card** *c)
void	**setMyCards** (**Card** *mc)
void	**setRound** (int r)
void	**initiateRound** ()
int	**decideWinner** ()
void	**completeRound** ()
void	**dealPreflop** ()
void	**dealFlop** ()
void	**dealTurn** ()
void	**dealRiver** ()
int	**getPotAmount** () const
int	**getInitContAmount** () const
int	**getRound** () const
Player *	**getPlayers** () const
Card *	**getCards** () const
int	**getNumPlayers** () const
void	**displayCards** (**Card** *cards)
void	**display** ()

The documentation for this class was generated from the following files:

- /home/pete/Dropbox/ScanSnap/School/Byoungmo/2019 Spring/CSC17A/git-repos/LeeByoungmo_CSC_17a_44083/Proj/Proj2/texasHoldem_Ver7/**Dealer.h**
- /home/pete/Dropbox/ScanSnap/School/Byoungmo/2019 Spring/CSC17A/git-repos/LeeByoungmo_CSC_17a_44083/Proj/Proj2/texasHoldem_Ver7/Dealer.cpp

Hands.h

```
/*
 * File:   Hands.h
 * Author: Byoung Mo Lee
 * Created on May 18, 2019 20:43 PM
 * Purpose: Hands Class for Texas Holdem
 *
 */


#ifndef HANDS_H
#define HANDS_H

#include "Card.h"



class Hands : public Card{
public:
    Hands();
    Hands(const Hands& orig);
    virtual ~Hands() {}
    void getInformation(Card* );
    int checkStraight(const int* const*, int);
    int isFlush(Card* );
    void checkFaceHands(Card*,int);
    void setHands(Card*,int );
    void displayFiveCards(Card*, int);
    Card* getFiveCards(Card* c,int n) {setHands(c, n);return fiveCards;}
    int getHands(Card* c,int n) {setHands(c,n);return hands;}
    //void displayInformation();
private:
    Card* fiveCards;  //select 5 of 7
    int hands;
    struct{
        int** faceSum;
        int faceRow;
        int** suitSum;
        int suitRow;
    };
};

#endif /* HANDS_H */

// hands=0: Highcard
// hands=1: One Pair
// hands=2: Two Pair
// hands=3: Three of a card
// hands=4: Straight
// hands=5: Flush
// hands=6: Full house
// hands=7: Four of a card
// hands=8: Straight Flush
```

Dealer.h

```
/*
 * File:   Dealer.h
 * Author: Byoung Mo Lee
 * Created on May 17, 2019 09:08 AM
 * Purpose: Texas Holdem
 *
 */
#ifndef DEALER_H
#define DEALER_H

#include <string>
#include "Card.h"
#include "Player.h"


using namespace std;

class Dealer{
private:
    int potAmount;
    int initialContributeAmount;
    int round;
    int numPlayers;
    Player* players;
    Card* cards;
    Card* myCards;
    void DeckOfCards();


public:
    class NegativeNumber {};
    Dealer();
    ~Dealer() {}
    Dealer(int n);
    void shuffle();
    void setPlayers(int n);
    void setCards(Card* c) {cards=new Card [52]; cards=c;}
    void setMyCards(Card* mc) {myCards=mc;}
    void setRound(int r) {round=r;}
    void initiateRound();
    int decideWinner();
    void completeRound();
    void dealPreflop();
    void dealFlop();
    void dealTurn();
    void dealRiver();
    int getPotAmount() const {return potAmount;}
    int getInitContAmount() const {return initialContributeAmount;}
    int getRound() const {return round;}
    Player* getPlayers() const {return players;}
    Card* getCards() const {return cards;}
    int getNumPlayers() const {return numPlayers;}
    void displayCards(Card* cards);
    void display();

};

//int Dealer::round=0;

#endif /* DEALER_H */
```

Player.h

```
/*
 * File:   Player.h
 * Author: Byoung Mo Lee
 * Created on May 17, 2019 10:16 AM
 * Purpose: Player Class for Texas Holdem
 *
 */

#ifndef PLAYER_H
#define PLAYER_H

#include <iostream>
#include <string>
#include "Card.h"
#include "Hands.h"

using namespace std;

class Player : public Hands  {
protected:
    string name;
    int balance;
    int order;      //if(order==0) Big blind, if(order==1) small blind
    int status;
    Card* myCards;
    Card* myFiveCards;
    int hands;
    int myPoints;

public:
    Player() {setPlayer("",10000,0,1);myCards=new Card[7];}
    ~Player() {}
    void setPlayer(string n,int b, int o, int s) {name=n;balance=b;order=o;status=s;
    void setMyCard(Card* mC) {myCards=new Card[7];myCards=mC;}
    void setName(string n) {name=n;}
    void setBalance(int b) {balance=b;}
    void setOrder(int o) {order=o;}
    void setStatus(int s) {status=s;}
    string getName() const {return name;}
    int getBalance() const {return balance;}
    int getNumCards() const;
    void updateMyHands();
    int getMyHands() const;
    Card* getMyFiveCards() const;
    int getMyPoints() const;
    int getOrder() const {return order;}
    Card* getMycards() const {return myCards;}
    int getStatus() const {return status;}
    void putMoneyToPot(int m) {balance-=m;}
    void takeMoneyFromPot(int m) {balance+=m;}
    void resetMyCards() {delete [] myCards;myCards= new Card [7];}
    void addCards(Card *c, int n); //mC=players.myCards, c=decked Cards, n=number
of addition
    void displayMyFiveCards();

};

#endif /* PLAYER_H */
```

Templates.h

```
/*
 * File:   Templates.h
 * Author: Byoung Mo Lee
 * Created on May 17, 2019 09:50 AM
 * Purpose: Templates
 *
 */

#ifndef TEMPLATES_H
#define TEMPLATES_H

#include "Card.h"

using namespace std;


template <class T>
void swapVars(T &var1, T &var2){
    T temp;

    temp=var1;
    var1=var2;
    var2=temp;
}

template <class T>
void selectionSortVar(T* var,int begin,int size){
    T maxValue;
    int minIndex;
    int end=begin+size;
    for(int start=begin;start<end-1;start++){
        minIndex=start;
        maxValue=var[start];
        for(int index=start+1;index<end;index++){
            if(var[index]>maxValue){
                maxValue=var[index];
                minIndex=index;
            }
        }
        swap(var[minIndex],var[start]);
    }
}

void selectionSort(int** array,int col,int begin, int size){
    int maxIndex, maxValue;
    int end=begin+size;
    for(int start=begin;start<end-1;start++){
        maxIndex=start;
        maxValue=array[start][col];
        for(int index=start+1;index<end;index++){
            if(array[index][col]>maxValue){
                maxValue=array[index][col];
                maxIndex=index;
            }
        }
        swapVars(array[maxIndex][1],array[start][1]);
        swapVars(array[maxIndex][0],array[start][0]);
    }
}

#endif /* TEMPLATES_H */
```

Card.h

```
/*
 * File:   Card.h
 * Author: Byoung Mo Lee
 * Created on May 16, 2019 23:52 PM
 * Purpose: Texas Holdem
 *
 */

#ifndef CARD_H
#define CARD_H

#include <memory>

class Card{
protected:
    int face; //2~J(11),Q(12),K(13),A(14)
    int suit;  //0-Spades, 1-Hearts, 2-Diamonds, 3-Clubs
public:
    Card() {face=-1;suit=-1;}
    Card(int f,int s) {face=f;suit=s;}

    void setCard(int f,int s) {face=f;suit=s;}
    void setFace(int f) {face=f;}
    void setSuit(int s) {suit=s;}
    void setACard(Card &c) {setCard(c.face,c.suit);}
    int getFace() const {return face;}
    int getSuit() const {return suit;}
    void displayCard();  //defined in Card.cpp
    void selectionSortByFace(Card* c, int begin, int size);
    void selectionSortBySuit(Card* c, int begin, int size);
    const Card operator=(const Card &right){
        if(this != &right){
            face=right.face;
            suit=right.suit;
        }
        return *this;
    }
};


#endif /* CARD_H */
```