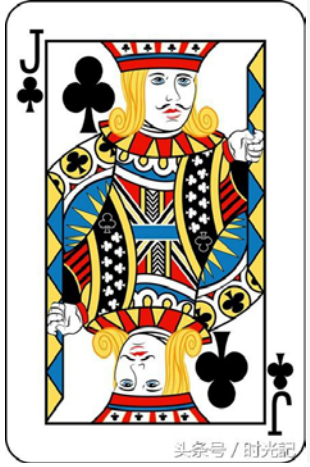
HW #6 (Blackjack21點)

- **In this project, your task is to write a Blackjack game in C++ or Python or Java (if you like to have graphics).**

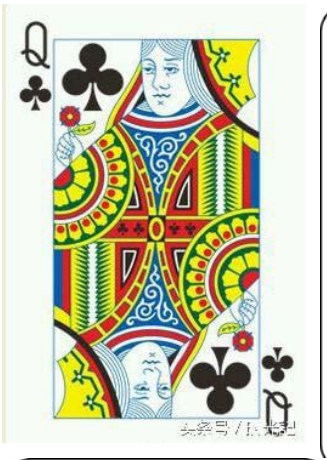
- **Blackjack is played with an ordinary deck**一幅撲克牌 **of 52 cards. Each card has a rank (Ace through King, or equivalently 1 to 13) and suit (Hearts**紅心**, Diamonds**方塊**, Clubs**梅花**, Spades**黑桃**).**





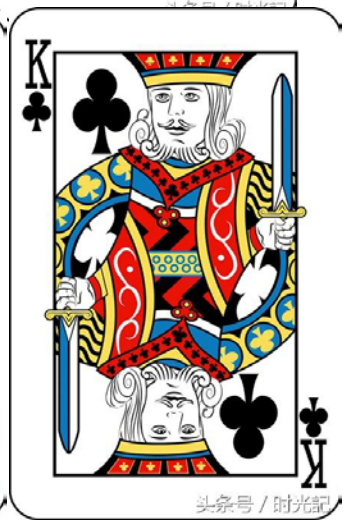

騎士，11 點






皇后，
12 點






國王，13 點

HW #6 (2)

- **In the game, human players and various other computer players are pitted使競爭 against a computer dealer. (The players do not play against each other; each player is simultaneously playing a one-to-one game against the dealer.)**

- **A round of Blackjack is played according to these rules described as follows.**

https://zh.wikipedia.org/zh-tw/%E4%BA%8C%E5%8D%81%E4%B8%80%E9%BB%9E

HW #6 (3)

- **The dealer gives each player two cards: one face down and one face up. Players can look at their own face-down card, but not other player's.**

- **Each player's goal is to get more points than the dealer, but not to exceed 21 points. A player's points are determined by summing the contributions of their cards: 1) face cards (rank = 11, 12, and 13) are worth 10 points;**

HW #6 (4)

2) an ace (rank = 1) is worth either 1 point or 11 points; 3) all other cards are worth their face value.

- After dealing out the starting cards, each player gets a turn to accumulate a hand as close as they can to 21 without going over. A player can draw additional cards one by one from the deck until they no longer want any more. If at any point the player's hand exceeds 21, they are "busted"爆裂 and lose.

HW #6 (5)

- **After all of the players have had their turn and drawn as many cards as they want, the dealer draws his cards.**

- **After all cards are drawn, the face-down cards are revealed and the dealer announces the winners. All ties打成平局 go to the dealer.**

HW #6 (6)

- **Of course it would not be interesting without a little money riding on the outcome. At the beginning of a round, before any cards are dealt, each player is asked how much they would like to bet賭注 on this round.**

- **If they win (i.e., beat the dealer), the player will double their money, if they lose, they lost the entire bet. The minimum bet is $1 at our table.**

HW #6 (7)

- **Our casino is very generous with credit, so players can bet more money than they have and go way down deep into the hole if they like.**
- **Okay, time for betting!**

  Bob, how much would you like to bet? 15

  Bob bets $15

  Meek bets $2

  Random bets $47

HW #6 (8)

- **The initial starting cards are:**

  Bob’s current hand: [??][10D]

  Meek’s current hand: [??][9H]

  Random’s current hand: [??][5D]

  Dealer’s current hand: [??][3C]

HW #6 (9)

- **Bob’s turn:**

  Bob’s current hand: [9C][10D] (19 points)

  Would you like to draw another card (Y or N): N

  Bob chooses to stay.
- **Meek’s turn:**

  Meek’s current hand: [8H][9H] (17 points)

  Meek chooses to stay.

HW #6 (10)

- **Random's turn:**

  Random's current hand: [8H][5D] (13 points)

  Random chooses to draw.

  Random's current hand: [8H][5D][4H] (17 points)

  Random chooses to draw.

  Random busted at 27!

HW #6 (11)

- **Dealer's turn:**

  Dealer's current hand: [3S][3C] (6 points)

  Dealer chooses to draw.

  Dealer's current hand: [3S][3C][4C] (10 points)

  Dealer chooses to draw.

  Dealer's current hand: [3S][3C][4C][8D] (18 points)

  Dealer chooses to stay.

HW #6 (12)

- **Let’s see how it turned out:**

  Yowzah! Bob wins $15

  Ouch! Meek loses $2

  Ouch! Random loses $47

HW #6 (13)

- **The standings so far:**

  Bob $115

  Meek $98

  Random $53

  Dealer $10034

  Another round? (Y or N):

HW #6 (14)

- **Each player has a name, a starting bankroll資金, and some strategy (how they decide how much to bet and how to determine when to draw and when to stay).**
- **As a player wins and loses money, their bankroll grows and shrinks. Of course, the end goal is to end up with lots of cash.**

- **Here are the types of Blackjack players we have:**

HW #6 (15)

- *Dealer:* **There is always exactly one dealer player in a game. The dealer always plays by the house rule, which mandates that they draw while their hand is worth 16 or fewer points, and stop when they reach 17 or more. The dealer's "bankroll" is actually the house money. The dealer never bets any of the house money, but it pays out the house money to any player which beats the dealer and keeps the money bet by any player that loses.**

HW #6 (16)

- *Human:* **A human player is one that makes choices by asking the user questions and getting answers using the standard I/O functions. For example, the name and starting bankroll of a human player are obtained by asking the user to type them in. At the beginning of the round, the user is prompted for the human's bet. On each turn of a human player the user is asked whether the player choose to draw or stay. There can be zero, one, or many human players in a game.**

HW #6 (17)

- *Meek:* **The meek溫順的 computer player is a rather meek soul. He usually bets just $2 every round. However, he goes a little crazy when he starts winning. Each time he wins 3 rounds in a row, he doubles his current betting amount. So after a string of 3 wins, he would start betting $4 a round. After 3 more wins, he would start betting $8. However, the first time he busts (i.e., his card total goes over 21), he goes back to his $2 bet. His card-choosing strategy is a little**

HW #6 (18)

bizarre怪誕的**, he draws a card whenever he has an even number of points, he stays whenever he has an odd amount. However if he has the seven of clubs in his hand, no matter what his total, he draws.**

- *Random:* **The random computer player is an unpredictable fellow. He randomly bets some amount between $1 and half of his bankroll each round. When it comes time for choosing whether to draw or stay, he also**

HW #6 (19)

makes random decisions. If he has 9 or less points, he always draws. If he has 10-12 points, he draws with random probability 80%, if he has 13-15 points he draws 70% of the time, if he has 16-18 points, he draws with 50% of the time, and if he has 19 points or over, he never draws.

- **The different players have a lot of behavior in common – the challenge of the assignment is structuring your player class hierarchy so as to avoid duplicate code.**

HW #6 (20)

- **There can be any number of players in a game. Before starting to play the game, you should ask the user how many players to create and of which kind they are, also each player gets a name and an initial starting bankroll.**

- **A dealer must also be created and he starts with the house's money account that always has an initial balance of 10,000.**

HW #6 (21)

- **Once all the players have been set up, your program should go into a loop playing rounds of Blackjack. At the end of each round, you should report the standings and ask the user if they would like to play another round (keeping the same players).**

- **You continue playing rounds until the user indicates he no longer wants to continue.**

HW #6 (22)

- **You have got card objects, deck objects, player objects, game objects, etc, all roaming about in this "object soup" and you will need to establish lines of communications that allow them to send messages to one another.**

- **This sort of situation comes up all the time in OOP – you have several cooperating objects, and they need pointers to each other so they can exchanges messages.**

HW #6 (23)

- **If an object only needs to refer to an object for a short time, such as just within a particular method, you might consider passing that object as an argument to that method.**
- **If an object needs to message to another object repeatedly, you will probably want to store a pointer to this object as a data member of the object that needs to message it.**

HW #6 (24)

- **Even though the Dealer would normally run the game, own the deck, dispend分發 cards, etc, you might find it cleaner to have a Game object which handles those functions and restrict the Dealer object to just handling the "player" aspect of the Dealer, (managing a hand of cards and deciding whether to draw or not).**

HW #6 (25)

- **Dividing up the tasks into two separate objects will simplify your design and helps the Dealer fit more naturally into the Player design structure.**

- **The Game object will take care of tracking the list of players, deciding whose turn it is, controlling the deck, and handling other game-related functionality.**

HW #6 (26)

- **Since you may have many players in a game and you do not want to run out of cards or have players count cards, your deck should be a "Vegas" deck of cards which is a bunch of normal-sized decks all mixed together.**

- **Create your deck with a 52-card pack for every 3 players (e.g., 9 players means a deck of 156 cards.)**

HW #6 (27)

- **In order to make sure you do not run out of cards in the middle of a round, you should check before the round begins. You can figure that you want enough remaining cards in the deck to cover at least 5 cards per player before you begin a round. If you do not have enough, shuffle the deck and start from the top.**

第一版

```
class Card {
    int rank;
    enum suit {spade, heart, diamond, club};
    int value; };

typedef Card Deck [52];

==========
class Player {
    string name;
    int bankroll;
    int bet;
    virtual strategy() = 0; };

class Human: public Player { };
class Computer: public Player { };
```

```
Computer Dealer;
    // bankroll = 10,000
    // strategy()
        // 1. draw while hand is worth 16 or fewer points
        // 2. stop when reach 17 or more

Computer Meek;
    // strategy();

Computer Random;
    // strategy();

main () {
    // How many players? each kind? each name? each bankroll?
    // Loop for each round
}
```