

# Course Project Phase 2



This is the second phase of the course project which we have already started. In this phase, we will improve our program from phase 1. You will do this in two stages: first, you will update the classes in `objects.py` and second, you will add implementation to some of the methods of the partially implemented Blackjack class in the file `blackjack.py`. The rules of the game we are implementing are described at the end in the Appendix. Please read those to get a better idea of the game.

*Don't be discouraged by the length of this document or the rules of the game listed in the Appendix. Just follow the instructions and you will have no problem getting the game to work.*

## Step 1: Update classes in `objects.py`:

Update the classes in the `objects.py` in the following way:

### 1. **Card Class:**

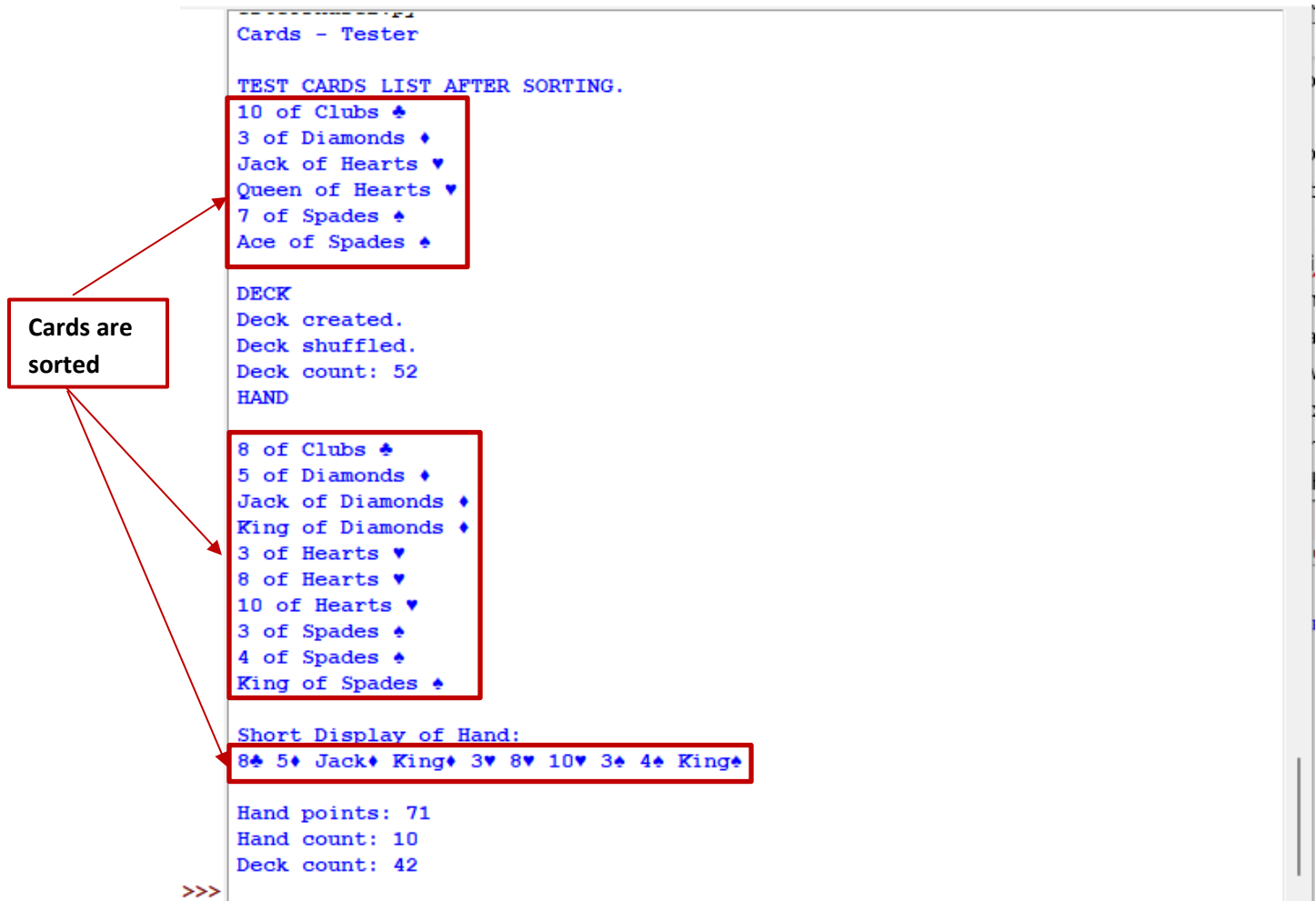
- Make the **Card** class comparable so that when a hand is displayed, the cards in the hand can be shown in a sorted order. When comparing two cards, their suit attributes are compared first and then the rank attributes. Note that the suits follow this order: Spades > Hearts > Diamonds > Clubs. (Luckily, these are alphabetically ordered as well!) The ranks follow this order: Ace > King > Queen > Jack > 10 > 9 > 8 > 7 > 6 > 5 > 4 > 3 > 2.
- Make the **card** class printable. You should replace the **displayCard** method with one of the special methods.

### 2. **Hand Class:**

- Add a method called **shortDisplay** to the Hand class that returns a shorter string with the rank and suit symbol of the sorted cards in the hand, such as "10♣ 3♦ Queen♥ Ace♠"
- Make the **Hand** class printable. Replace the **displayHand** method with one of the special methods and have it return a combined string that will print the hand with cards displayed on multiple lines.
- Modify the **points** method of the **Hand** class to implement how points are counted in Blackjack (as given in section 3c of the Appendix). As per this rule, value of an Ace can be taken as 1 or 11 to not have the hand value go over 21.
- Add a readonly property called **isBusted** that is True when the hand points are greater than 21.
- Add a readonly property called **hasBlackjack** that is True when the hand has exactly 2 cards and the points are 21.

### 3. Deck and Hand Classes:

- a. Make these classes iterable, by adding a Generator function.
  - b. Replace the count methods of these classes with a special method that allows the client code to pass objects of these classes to the built-in `len` function.
4. **main() function:** Note that with these changes, the test code in main function of your existing objects.py file will break partially. Replace the main function in your object file with the main function from the attached **objectsTesterPhase2.py** file. And test it out. Here's a sample output of the updated object.py file:



```
Cards - Tester

TEST CARDS LIST AFTER SORTING.
10 of Clubs ♣
3 of Diamonds ♦
Jack of Hearts ♥
Queen of Hearts ♥
7 of Spades ♠
Ace of Spades ♠

DECK
Deck created.
Deck shuffled.
Deck count: 52
HAND
8 of Clubs ♣
5 of Diamonds ♦
Jack of Diamonds ♦
King of Diamonds ♦
3 of Hearts ♥
8 of Hearts ♥
10 of Hearts ♥
3 of Spades ♠
4 of Spades ♠
King of Spades ♠

Short Display of Hand:
8♣ 5♦ Jack♥ King♥ 3♥ 8♥ 10♥ 3♠ 4♠ King♠

Hand points: 71
Hand count: 10
Deck count: 42

>>>
```

As you can see in the sample output, the cards in the testcardsList list and the hand are shown in sorted order.

## Step 2: Add code to blackjack.py file:

### Implement Blackjack class:

You are given a file **blackjack.py** which contains a **Blackjack** class definition and a **main** function that uses this class to play the game. In Phase 2, you will be implementing some of the methods of the **Blackjack** class:

1. **\_\_init\_\_(self, startingBalance)**: The constructor that accepts a starting balance to start the game with. It initializes all the attributes correctly.
2. **displayCards(self, hand, title)**: This method first prints the given title, prints the given hand and prints the points of that hand.
3. **getBet(self)**: Method to update **self.bet** by prompting the user for the bet amount, making sure bet is a positive number and is less than or equal to **self.money**.
4. **takePlayerTurn(self)**: Method to simulate player playing one turn by dealing a card to the player's hand.
5. **setupRound(self)**: Method to setup a round by doing these steps:
  - a. Initialize **self.deck** to a new Deck object and shuffle it
  - b. Initialize **self.dealerHand** and **self.playerHand** to new Hand objects
  - c. Deal two cards to the **playerHand**, and one card to the **dealerHand**
  - d. Finally, print **dealerHand** and **playerHand** using **displayCards** method

We will implement more methods (such as to play dealers hand, decide a winner) in Phase 3).

### Add code to main function:

The **main** function in the **blackjack.py** file uses the **Blackjack** class to play the game. For the phase 2, we will implement the game only partially (only up to step 3c from the rules in the Appendix). That's ok. We will complete the game in the next phase. Add code to the main function:

- a. Call the **getBet** method of blackjack object
- b. Call the **setupRound** method of blackjack object to get the game started
- c. Play player hand:
  - i. Prompt the user whether to Hit (h) or Stand (s)
  - ii. If user picks stand, print the player points and end the player play.
  - iii. If user picks hit,
    1. deal a card to the **playerHand**.
    2. check if the player hand is busted with the latest card added. If so, print the player points and end the player play
    3. otherwise, repeat from step i.

Three sample runs of **blackjack.py** file for this phase are attached:

- blackjack-phase2-sample1.jpg
- blackjack-phase2-sample2.jpg
- blackjack-phase2-sample-3.jpg

Please submit edited versions of objects.py and blackjack.py with a block-comment at the top.

## Appendix: Game Rules

Here's the description of a simplified version of the game of Blackjack which we are developing. (Look up online the rules of the actual game if you wish)

1. The basic objective of the game is that the player wants to have a hand value that is closer to 21 than the dealer's, without going over 21.
2. The game begins by the player having a starting balance.
3. In each round of the game,
  - a. The player first places a bet. The amount must be less than or equal to the player's current balance.
  - b. Next, the dealer sets up the round by:
    - i. Starting with a fresh new deck, new playerHand and a new dealerHand.
    - ii. Dealing two cards to the player, and one card to himself.
  - c. Next, the player plays her hand by repeatedly indicating whether to draw another card ("hit") or stop at the current total ("stand"). Player bases this decision based on the value of the hand using this rule: Cards 2-10 have face value; King, Queen, Jack are worth 10; and Aces are either a 1 or an 11 — whichever makes the hand value not go over 21. E.g., if player has an Ace and a King, the Ace should be counted as 11 so that hand value is 21. However, if the player has two Aces — one of them should be counted as 1 so that the hand value is  $11+1 = 12$  (instead of  $11+11=22$  which will go over 21).
  - d. If the player draws a card (by indicating "hit") and that card makes player's hand value go over 21, player hand is a bust. That is an automatic loser. Otherwise, the dealer continues to deal a card until the player indicates a "stand" or the player hand busts.
  - e. Once the player plays her hand (either by ending the current round by indicating "stand" or "bust"), the dealer plays his hand: first he draws one more card for himself, and then must draw cards until he has a total of 17 or more. The dealer has no choice in how to play the hand. He must continue taking cards until his total is at least 17.
  - f. Once both the player and the dealer have played hands, the winner is decided, and the bet is settled:
    - i. If player hand busts, the player is the loser, and she loses her bet amount.

- ii. If the dealer busts by going over 21, the player wins her bet.
  - iii. If both the player and the dealer didn't bust, then if the dealer's hand total is higher than the player's, the player lose the bet. Otherwise, if the player hand total is higher than the dealer's, the dealer loses and pays the player her bet amount.
  - iv. If the player and the dealer tie, with the same exact total, it is called a "push", and the player does not win or lose the bet.
4. Declaring a hand Blackjack: There is an additional rule which declares a hand a natural or a Blackjack. If a player's first two cards are an ace and a "ten-card" (a face card or 10), giving her a count of 21 in two cards, this is a natural or "Blackjack" and this is an automatic win for the player and the player gets ***one and a half times the amount of her bet.***