

CS 60

Programming Assignment #3

Due Date: May 23, 2010 (6:00 PM). 75 Points.

Remember: Points will be deducted for homework turned in after the Due Date.

Deadline: May 26, 2010 (6:00 PM)

Remember: No homework will be accepted after the Deadline

Remember turnin your code and student.id file to hw07@cs60 (see the class web page).

You must work on this assignment independently.

For this assignment you will write a C++ program that will read two hands of five cards each from a standard deck of playing cards. More specifically, a deck of cards consists of 52 cards. Each card has a "rank" and a "suit." The allowable ranks are the characters: 2, 3, 4, 5, 6, 7, 8, 9, T, J, Q, K, A (and this is their order - note that this is not alphabetical order). The allowable suits are: C (Clubs), D(Diamonds), H (Hearts), S (Spades). Note that the suits are in alphabetical order.

The input to the program consists of 10 lines each with two characters followed by an end-of-line. There cannot be any blanks or extra characters in any of these lines. The first character in a line represents the rank of a card and the second character represents the suit of the card. The first five lines represent the five cards for the first hand and the last five lines represent the the five cards in the second hand.

Your program will begin by reading both hands and checking if there are any illegal characters. If this is the case your program should print out an appropriate message and then exit.

Your program should print all the cards in both of the hands as they are given in the input (see the sample output in the next few pages). Then your program will sort the cards in increasing order with respect to the rank of the cards. I.e., first all 2s, then all 3s, ..., then all Kings and then all Aces. In case there are two or more cards with the same rank they will be sorted in increasing order with respect to the suit of the cards. I.e., all cards with the same rank will be arranged so that the club one appears first, then the diamond, then the heart and then the spade. Thus 3 of H is before 9 of H and 3 of H is before 3 of S. But 3 of H is after 2 of H and 3 of H is after 3 of C. Likewise, J of H is before A of C and after 9 of S.

Then your program should print for each hand (sorted in increasing order (with respect to the ordering defined above)). The possible type of poker hands are as follows and you should print out the highest type for each hand.

- (1) Nothing - None of the cases below occurs.
- (2) One Pair - Two cards of the same rank.
- (3) Two Pairs - Two cards of one rank plus two cards of another rank.
- (4) Three of a Kind - Three cards of the same rank.
- (5) Straight - A hand with five consecutive cards like 2,3,4,5,6. An Ace can be high or low, i.e., an ace (A) can be part of the straight A,2,3,4,5, or the straight T,J,Q,K,A.
- (6) Flush - Five cards of the same suit.
- (7) Full House - Any three cards of one rank plus any two cards of another rank.

- (8) Four of a Kind - Four cards of the same rank.
- (9) Straight Flush - A straight with all five cards of the same suit.
- (10) Royal Flush - This is a straight flush with an Ace as the high card (i.e., T,J,Q,K,A).

Once your program figures the highest poker hand for each of the two hands, it should print out which of the hands wins or print that there is a tie. Note that a highest poker hand of type i beats a highest poker hand of type j when $j < i$. What if the two highest poker hands are of the same type? In this case we use the following tie breaking rules, and if ties persist then it is a tie. Each entry corresponds to the case when the two hands have the highest poker hand of the following type.

- (1) Nothing - Ties are broken by the highest rank card; if ties persist, then by the second highest ranking card, and so on. If the rank of all the cards match then it is a tie.
- (2) One Pair - Ties are broken by the rank of the pair; if ties persist by then by the highest rank card; if ties persist then by the second highest card and so on. If the rank of all the card match then it is a tie.
- (3) Two Pairs - Ties are broken by the highest rank pair; if ties persist by the lowest rank pair; if ties persist by the rank of the highest card that is not part of any pair; if ties persist then it is a tie.
- (4) Three of a Kind - Ties are broken by the rank of the the triplet; Note that since we are using one deck of card we will not have further ties.
- (5) Straight - Ties are broken by the rank of the highest rank card in the straight (i.e., the highest ranking card in straight A,2,3,4,5 is 5, and in straight 10,J,Q,K,A is A); if a tie persists then it is a tie.
- (6) Flush - In this case it is a tie (contrary poker rules, but it simplifies the code).
- (7) Full House - Ties are broken by the rank of the card in the three cards that have the same rank. Note that since we are using one deck of card we will not have further ties.
- (8) Four of a Kind - Ties are broken by the rank of the card in the four cards that have the same rank. Note that since we are using one deck of card we will not have further ties.
- (9) Straight Flush - Ties are broken by the rank of the highest rank card in the straight (see the tie breaking rule for the straight); if a tie persists then it is a tie.
- (10) Royal Flush - In this case it is a tie.

The third component of this program is to select 5 out of the 10 cards (in both the hands) that will represent the best possible hand (according to the rules specified above including the tie breaking rules). Then you need to print the best hand in sorted order (according to our rules above) and identifying the type of hand ((1) - (9)). There are several ways to do this. One way is to design the member functions to identify (1) - (9) above when a hand has five or more cards and then select the best cards for the tie

breaking rule. The other approach is to design the member functions to identify (1) - (9) above when a hand has five cards and use those member functions for every subset of 5 cards from the 10 cards and keeping the best hand (under tie breaking rules too). Either approach is fine or you may come up with another approach.

Program Design: For this program you will use two classes `Hand` and `Card` given in the sample code below. First you read each of the hands and print them. At the same time you store them in two objects of class `Hand`, in sorted order. Then the sorted hand should be passed in turn to a sequence of member functions, which will determine the poker hand. Then you compare and print the best hand. Then you will do the third part of selecting and printing the best hand of 5 cards taken out of the 10 cards.

For class `Hand` you need to overload the `>` operator and the equality operator (`==`). You also need to overload `cin` and `cout`. You will also need to overload the plus operator (`+`) to mean find and return the best hand out of the 10 cards in both hands.

Each class must have its own `.H` file and its own `.C` file. A make file should also be included. When we type `make` the executable called `executeit` must be created. Turnin all of this via turnin to `hw07@cs60` including the student.id file.

```
class Card;

class Hand {
private:
    Card*  nextcard;
public:
}

class Card {
private:
    char    rank;
    char    suit;
    Card*   nextcard;
public:
}
```

```
Input
3H
2D
AH
AD
JC
JD
4H
KH
KC
```

Sample Output

The input cards for Player 1 are:

- 3 of H
- 2 of D
- A of H
- A of D
- J of C

The input cards for Player 2 are:

- J of D
- 4 of H
- K of H
- K of C
- Q of D

The sorted cards for Player 1 are:

- 2 of D
- 3 of H
- J of C
- A of D
- A of H

The sorted cards for Player 2 are:

- 4 of H
- J of D
- Q of D
- K of C
- K of H

Player 1 has One Pair

Player 2 has One Pair

Player 1 wins

The best hand of five cards out of the 10 cards is:

- Q of D
- K of C
- K of H
- A of D
- A of H

Best five cards have Two Pairs

Input

JD

JH

QH

QC

3D

2H

2D

KH

KD

KC

Sample Output

The input cards for Player 1 are:

J of D

J of H

Q of H

Q of C

3 of D

The input cards for Player 2 are:

2 of H

2 of D

K of H

K of D

K of C

The sorted cards for Player 1 are:

3 of D

J of D

J of H

Q of C

Q of H

The sorted cards for Player 2 are:

2 of D

2 of H
K of C
K of D
K of H

Player 1 has Two Pairs
Player 2 has Full House

Player 2 wins

The best hand of five cards out of the 10 cards is:

Q of C
Q of H
K of C
K of D
K of H

Best five cards have Full House
