NTUST OOP Midterm Problem Design

Subject: Card Rank Analyzer

Contributor: 陳柏睿

Main testing concept:
Familiar with sorting, complex logic, and string handling.

Basics	Functions
■ C++ BASICS	□ SEPARATE COMPILATION AND NAMESPACES
□ FLOW OF CONTROL	■ STREAMS AND FILE I/O
■ FUNCTION BASICS	□ RECURSION
□ PARAMETERS AND OVERLOADING	□ INHERITANCE
■ ARRAYS	□ POLYMORPHISM AND VIRTUAL FUNCTIONS
■ STRUCTURES AND CLASSES	□ TEMPLATES
□ CONSTRUCTORS AND OTHER TOOLS	□ LINKED DATA STRUCTURES
□ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES	□ EXCEPTION HANDLING
■ STRINGS	□ STANDARD TEMPLATE LIBRARY
□ POINTERS AND DYNAMIC ARRAYS	□ PATTERNS AND UML

Description
You are given 5 playing cards, where each card consists of a suit and a rank. Your task is to determine the strongest possible hand ranking from these five cards.

Hand Rankings (Strongest to Weakest)
Straight Flush – Five consecutive ranked cards of the same suit.
Four of a Kind – Four cards with the same rank.
Full House – Three cards of the same rank and a pair of another rank.
Flush – Five cards of the same suit (not in sequence).
Straight – Five consecutive ranked cards of different suits.
Three of a Kind – Three cards of the same rank with two unrelated cards.
Two Pair – Two pairs of the same rank with one unrelated card.
One Pair – A single pair of the same rank with three unrelated cards.
High Card – If none of the above applies, the highest-ranked card determines the hand strength.

About Ace:
The Ace (A) can be used as the highest card (A, K, Q, J, 10) or the lowest card (A, 2, 3, 4, 5) but cannot be in the middle (e.g., Q, K, A, 2, 3 is invalid for a straight).

Input Format
Each hand consists of 5 cards, each containing a single playing card represented by a suit (single letter) and rank (string) separated by a space:
Suits:
S (Spades)
H (Hearts)
D (Diamonds)
C (Clubs)
Ranks: 2 to 10, J, Q, K, A
Each hand is terminated by a command “parse”.
The program should output the result upon encountering the command “parse”.

Note:
The input cards are not necessarily given in order, you may need to sort them for easier analysis.
There will be no duplicate cards, so you don’t have to deal with that.

Output Format
Print the name of the strongest possible hand, such as "Straight Flush" or "Four of a Kind".
If a hand contains an incorrect number of cards, an invalid suit, or an invalid rank, output "Invalid Input" instead of determining the hand ranking.

Sample Input	Sample Output
S 10 S J S Q S K S A parse C A D 2 H 3 S 4 C 5 parse D 3 H 3 S 3 C K H K parse H 2 C 5 D 9 S J H K parse E W D 14 H 2 C 5 D 9 parse	Straight Flush Straight Full House High Card Invalid Input

□ **Easy, Only basic programming syntax and structure are required.**
■ **Medium, Multiple programming grammars, and structures are required.**
□ **Hard, Need to use multiple program structures or complex data types.**

Expected solving time:
60min.

Other Notes: