Assignment: Poker Hand Evaluation (High Card and Pair)

Objective: The objective of this assignment is to create a simplified poker hand evaluation program that focuses on creating a hand of cards and evaluating it for high cards and pairs.

Task: You are tasked with developing a Java program that simulates a poker hand evaluation. Your program should perform the following tasks:

1. Create a `Deck` class that represents a standard deck of 52 cards, each with a rank (2 to Ace) and a suit (Hearts, Diamonds, Clubs, Spades).
2. Create a `Hand` class that represents a player's hand. The hand should be able to hold exactly 5 cards.
3. Implement a method in the `Hand` class called `evaluate` that evaluates the hand for high cards and pairs. For this assignment, we will define high cards as cards with ranks of 10, Jack, Queen, King, or Ace. A pair is defined as two cards with the same rank. You need to find all pairs.
4. In the `Main` class, create a deck, draw 5 cards from the deck, and add them to a hand.
5. Call the `evaluate` method on the hand to display the cards in the hand and determine if it contains high cards and/or a pair.

Example Output: Below is an example of the expected console output for your program:

```
--- Poker Hand Evaluation ---
Your Hand:
7 of Hearts
Ace of Spades
King of Diamonds
2 of Clubs
Ace of Clubs

Evaluation Result:
High Cards: Ace of Spades, King of Diamonds, Ace of Clubs
Pair: Ace
```

or

```
--- Poker Hand Evaluation ---
Your Hand:
NINE of HEARTS
TWO of SPADES
TWO of HEARTS
SIX of HEARTS
NINE of DIAMONDS

Evaluation Result:
```

```
High Cards: None
Pair: NINE, TWO
```

Grading Criteria:

- Properly implemented `Deck`, `Hand`, and `Main` classes.
- The `evaluate` method correctly identifies high cards and pairs.
- The program produces output that displays the hand's cards and evaluation results.

Bonus Challenges (Optional):

1. Implement additional hand evaluation criteria, such as two pairs or three of a kind.
2. Extend the program to allow multiple players to have hands and evaluate each player's hand.
3. Create a graphical user interface (GUI) for the program to display cards and evaluation results.

Submission: Submit your Java source code files (`.java`) along with any additional documentation or comments that explain your code. Follow the instructions on Canvas for CodeGrade submission.

Hints:

- to get hold of all values in an `enum`, use the method values() on the `enum` you've created
- in order to compare enum values, use to method ordinal() on the values to get hold of their positions in the enum
- if you want to create a string from a list, you could use the method String.join(). For example: String.join(",", List.of("One", "Two", "Three")) will create the string "One,Two,Three".
- to check whether a list is empty or not, use list.isEmpty()
- to check if a list contains a certain object, use the list.contains() method which returns true of false. For example List.of("One", "Two", "Three").contains("One") returns true.