Coding Exercise: Simulate Blackjack

Objective

Your goal is to create a text-based (console) version of Blackjack using **Python**. The game will be played by one Player against the Dealer (the computer).

Technical Requirements

- **Language:** The entire exercise must be coded in **Python**.
- **Package Structure:** The final project must be structured as a Python package.
- **Installability:** The package should be "pip installable"
- **Importability:** Once installed, a user should be able to import and run your game

---

The Basic Core Game

Objective

Get a single, playable round working. Focus on the core logic: dealing, hitting, standing, and determining a winner.

Core Rules to Implement

The Deck

Create a standard 52-card deck. For this level, you can simplify it. A list of numbers is fine (e.g., `[2, 3, 4, 5, 6, 7, 8, 9, 10, 10, 10, 10, 11] * 4`).

Values

- Cards 2-10 are worth their face value.
- Jacks, Queens, and Kings (J, Q, K) are all worth 10.
- Aces (A) are worth 11... for now.

The Game Flow

Follow this sequence to structure your game round:

1. **Shuffle:** Shuffle your deck
2. **Initial Deal:**
    - Deal two cards to the Player (face up).
    - Deal two cards to the Dealer (one face up, one "face down").

3. **Calculate Totals:** Show the Player their total and the Dealer's visible total (the value of their single face-up card).
4. **Player's Turn:**
    - Ask the Player if they want to "Hit" (take another card) or "Stand" (keep their current hand).
    - **If Hit:** Deal one card. If their new total is over 21, they **bust** (lose) immediately. The game ends.
    - **If Stand:** Their turn ends, and play proceeds to the Dealer.
    - Continue this loop (Hit/Stand) until the Player stands or busts.
5. **Dealer's Turn:** (Only occurs if the Player did not bust)
    - Reveal the Dealer's face-down card and show their full initial total.
    - The Dealer must **hit** if their total is 16 or less.
    - The Dealer must **stand** if their total is 17 or more.
    - If the Dealer hits and their total goes over 21, they **bust**, and the Player wins.
6. **Determine Winner:** (Only occurs if neither player busted)
    - Compare the Player's total to the Dealer's total.
    - The hand with the higher total wins.
    - If the totals are the same, it's a **push** (a tie).

Expectations

The primary goal of this exercise is to evaluate your comfort and proficiency with the **Python** programming language, particularly your grasp of **Object-Oriented Programming (OOP)** principles as you progress through the levels. You are expected to deliver code that is functional and **runnable**. Alongside your solution, please provide a clear **explanation or justification** for your design choices.