Code Louisville Capstone Project

Software Development I

Project Idea: Blackjack/21 utilizing Deck of Cards API. The project will aim for a minimum viable product, that does not include playing card images, or the player's ability to split/double-down. If time allows the full vision of this product will be realized.

Game Summary: In Blackjack, a player and a dealer are each dealt a hand of 2 cards, with one of the dealer's cards face-down (hidden). The player then chooses to HIT (draw a card), STAY (stop drawing cards and end turn), or SPLIT/DOUBLE-DOWN (edge cases where a player has 2 of the same card, or chooses to double their bet when holding a certain value of cards. When the player's turn has ended (after they 'stay'), the dealer takes their turn and operates on house rules (stay at 17 & hit at 16, etc.). After the dealer finishes their turn, the value of the hands is compared and the player closest to, or at, 21 is the winner. Both players getting 21 results in a tie and neither party wins.

Nouns: player, dealer, card(s), hand, house rules

Verbs: choose, hit, stay, split/double-down

Included Features:

- Implement a "master loop" console application where the user can repeatedly enter commands/perform actions, including choosing to exit the program
- Connect to an external/3rd party API and read data into your app (Deck of Cards API)
- Create an additional class which inherits one or more properties from its parent (? Dealer is a player)
- Read data from an external file, such as text, JSON, CSV, etc. and use that data in your application (? player scores file)
- Create 3 or more unit tests for your application (?)
- Implement a log that records errors, invalid inputs, or other important events and writes them to a text file (?)

Classes:

- Player (HAS A hand)
- Dealer (IS A player; USES A house rules)
- Hand (HAS A card(s))
- House Rules - ? not a class but constants ?
- Card (unnecessary class since API handles this?
    - Use API data to set instance/object data

        ▪ Do this with a constructor?

Game Flow:

1) Run File
   a) Title Screen: Blackjack!   High Score: "Joe Somebody" : 400;
   b) Choose Beginning: New or Returning Player?
      i) If returning should list names? Or match with what player enters?
   c) Initialize Deck (API call)
   d) Separate initiating function for the players to give them 2 draws?
   e) Initiate Player hand and dealer hand (one card face down/hidden)
      i) Total = X; this given for each player and keeps running total. For dealer this total will show but will only include the value of the face-up card.
   f) Player Turn: Choose Hit, Stay, or Split/Double-down***`Loop while total < 21`***
      i) HIT – draw a card (API), add to the player hand and this will be added to total
         (1) Player choice again
      ii) STAY – end player turn, begin dealer turn
      iii) SPLIT – create 2nd hand for player
         (1) Begin to loop through player's choice for - each hand in turn OR one at a time
            (a) **Check rules on exact pacing**
      iv) Double-Down – increase the "bet" by 2x
            (a) **Check rules on exact calculation**
   g) Dealer Turn: Use house rules on hand total to decide dealer choice
      i) Dealer can **ONLY** hit/stay
   h) End Dealer Turn / Compare Hand Scores
   i) Win Condition and Message
      i) "You win!", "Draw", "You lost!"
   j) Calculate overall score/bet? Add to running total.
      i) Write this score to (read-only? Need to research writing to files…) file
         (1) player.Score = 400;
   k) End Conditions: Play Again? Change player? Exit?
      i) Can also implement a 'Help' display
         (1) Lists the commands available
2) Loop to [Run File]

Extras:

1) Player can change house rules.
2) Implement other simple card games to the game as extras
         (1) High card (can also be used as a chooser for 'who-goes-first', etc.)

        (2) War (extended high card game)
    ii) More than one player (blackjack “table”)

Challenges:

- Magic Numbers – set constants for win score (21), house rules, etc.….?
- Which nouns need to be a class?
- Use API correctly; Instance of deck or deck is handled exclusively by API?

Ideas:

- Console.Title to set window name
- Console.SetWindowSize(numberColumns, numberRows) to set window size (good for making sure cards display right on all screens).
- Console.Beep() method for indicating win/loss?
- Console.Clear() refresh/clean up console window (can make console seem paginated?)
- Console.ReadKey() / ConsoleKey.[RightArrow/UpArrow/DownArrow/LeftArrow/Escape] – set up a menu navigable by arrow keys (?)
- Player turn: loop while total < 21 && !stay/bust?; check for total == 21, then if bool BUST == true, each time a player draws a card, <u>then</u> offer choice to hit/stay
    - Total check happens at while loop beginning… for instant loop-end nest loops? …One each, bust & total <21… 3 exit conditions
        - Or, 3 exit conditions to trigger a loop-ending object that includes a pass-thru?
            - Pass-thru: “bust” message, end of turn/initiate dealer turn, user ‘stays’
- Typewriter effect? String.split(“ “) then use a loop to append a pause after each character for the words being typed; OR remove whitespace then append a pause after each character