Fulltime Project – Card Game

The objective of this project is to create a simple card game based on the following specification. The game will consist of rounds, where each player draws a single card from a shuffled deck. The players' cards will be compared to determine the winner of the round (whose score will be increased) and/or any penalized players (whose score will be decreased). A scoreboard should be displayed at the end of each round (to track player progress). Rounds will continue until an overall game winner is determined. Below is a detailed outline of the project requirements.

Interface – The program **MUST** utilize a text-based interface (i.e. console/terminal). Graphical "Event Driven" applications will not be accepted. Player input should be read via the keyboard, and all information output to the display as text. Please strive to make your interface clean, readable, and user friendly.

Players – The game must allow for two through four players, the number of which should be selectable at the beginning of a game. Each player should at a minimum have a unique identifier as well as an associated score (which starts at zero, and may not drop below zero). Players should be notified when it is their turn, and be required to press a key to draw their card. Players should always take their turn in the same order every round.

Deck – The deck will consist of 56 cards, 52 of which are standard face + suit cards, and four of which are special penalty cards. Access to cards within the deck can be implemented in any way, so long as that access conforms to the requirement that the deck be in a "Shuffled" state at the beginning of each round. When a player selects a card from the deck, their card should be displayed on screen for all to see.

Cards – Cards are ranked based on their face value first and their suit second. Face values range from *Two* to *Ten*, followed by *Jack*, *Queen*, *King* and finally *Ace* (just like a standard deck of cards). Suits are ranked based on the following hierarchy: *Spade* > *Heart* > *Diamond* > *Club*. Penalty cards can be represented in any way, but do not have a rank like the other cards (they will have special scoring associated with them).

Scoring – At the end of a round, the player with the highest ranked card should be displayed as the winner and will have their score increased by two. Any players that drew a penalty card should have their scores reduced by one. This should be indicated via a scoreboard, showing the running totals for all players. In order to win the game, a player must reach a score of 21 or greater, but shall only be considered the champion if they lead by two points (i.e. if one player has a score of 21, but another has a score of 20, the game must continue).

Your code will be reviewed; treat this as a professional project, and consider aspects of extensible design and potential code reuse while developing. Please supply any and all files (documentation, make files, source code, binaries etc.) in a single compressed archive, with a filename containing both your last name and first name, to:

Projects@MaxetaTech.com (Questions regarding project requirements may be sent to: JHorvath@MaxetaTech.com)

Note: Development may be done in any language. Take as much time as necessary (within reason) in order to fulfill the specification. However please specify any language version, compiler version, and or IDE used to help rebuild the project appropriately.