```
import random
import re
import pickle

class Casino:
    def __init__(self):
        self.card_info = self.load_card_info()  # Load saved card info if it
exists
        self.start_casino()

    def load_card_info(self):

        try:
            with open('card_info.pkl', 'rb') as file:
                return pickle.load(file)
        except FileNotFoundError:
            return None

    def save_card_info(self, card_number):

        with open('card_info.pkl', 'wb') as file:
            pickle.dump(card_number, file)

    def start_casino(self):
        # Ensure card information is provided
        self.get_credit_card_info()

        while True:
            self.game = input("What do you want to play?\n A. Blackjack\n B.
Poker\n C. Roulette\n D. Slot Machines\n E. Exit\n").strip().lower()
            if self.game == 'a':
                self.Blackjack()
            elif self.game == 'b':
                self.Poker()
            elif self.game == 'c':
                self.Roulette()
            elif self.game == 'd':
                self.Slot_Machines()
            elif self.game == 'e':
                print("Thanks for playing! Goodbye!")
                break
            else:
                print("Invalid choice. Please select a valid game.")
```

```
            self.play_again()

    def get_credit_card_info(self):

        if self.card_info:
            print(f"Welcome back! Your stored card number ends with
{self.card_info[-4:]}.")
            choice = input("Would you like to use the stored card? (yes/no):
").strip().lower()
            if choice == 'yes':
                return

        while True:
            card_number = input("Please enter your credit card number (format:
XXXX-XXXX-XXXX-XXXX): ")
            if self.validate_card_number(card_number):
                self.card_info = card_number
                save_choice = input("Would you like to save your card info for
next time? (yes/no): ").strip().lower()
                if save_choice == 'yes':
                    self.save_card_info(card_number)
                break
            else:
                print("Invalid card number format. Please try again.")

    @staticmethod
    def validate_card_number(card_number):

        pattern = r'^\d{4}-\d{4}-\d{4}-\d{4}$'
        return bool(re.match(pattern, card_number))

    def play_again(self):
        while True:
            choice = input("Do you want to play again, switch games, or exit?
(play/switch/exit): ").strip().lower()
            if choice == 'play':
                return
            elif choice == 'switch':
                break
            elif choice == 'exit':
                print("Thanks for playing! Goodbye!")
```

```
                exit()
            else:
                print("Invalid input. Please type 'play', 'switch', or 'exit'.")

    def Blackjack(self):
        print("Welcome to Blackjack!")
        player_hand = [random.choice(self.get_deck()),
random.choice(self.get_deck())]
        dealer_hand = [random.choice(self.get_deck()),
random.choice(self.get_deck())]

        print(f"Your hand: {player_hand}")
        print(f"Dealer's first card: {dealer_hand[0]}")




        print("Blackjack game finished.")

    def Poker(self):
        print("Welcome to Poker!")
        num_players = int(input("How many players would you like to play against?
"))
        hands = {f'Player {i+1}': [random.choice(self.get_deck()),
random.choice(self.get_deck())] for i in range(num_players)}

        for player, cards in hands.items():
            print(f"{player}'s hand: {cards}")




        print("Poker game finished.")

    def Roulette(self):
        print("Welcome to Roulette!")
        bet = input("What do you want to bet on? (number/color): ")
        winning_number = random.randint(0, 36)
        winning_color = 'red' if winning_number % 2 == 0 else 'black'

        print(f"The ball landed on {winning_number} {winning_color}!")
```

```
        print("Roulette game finished.")

    def Slot_Machines(self):
        print("Welcome to Slot Machines!")
        symbols = ["🍒", "🍋", "🍊", "🍉", "🍇"]
        spin_result = [random.choice(symbols) for _ in range(3)]

        print(f"Slot spin result: {spin_result}")




        print("Slot Machines game finished.")

    def get_deck(self):
        """Create a standard deck of playing cards."""
        suits = ['♠', '♥', '♦', '♣']
        ranks = ['A', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'J', 'Q',
'K']
        return [f"{rank}{suit}" for suit in suits for rank in ranks]

# Run the casino program
casino = Casino()
```