# Card Validator

## Background:

The provided Python code is designed to validate credit or debit card numbers using the Luhn algorithm. This algorithm is widely used for validating identification numbers, especially credit card numbers. It checks whether the given card number follows a specific mathematical formula, thereby helping to detect common errors such as mistyped or incorrect digits.

## Explanation of Main Function:

The core functionality of the program resides in the `validate` method of the `CardValidator` class. This method takes a card number as input and performs the following steps:

1. \*\*Digit Extraction\*\*: The method extracts only the digits from the input, discarding any non-numeric characters.  
2. \*\*Check Digit Separation\*\*: The last digit of the card number is separated as the check digit. This is the digit against which the validation is performed.  
3. \*\*Digit Reversal\*\*: The remaining digits are reversed to simplify processing according to the Luhn algorithm.  
4. \*\*Doubling Rule\*\*: Every second digit (based on the new order) is doubled. If the result is greater than 9, 9 is subtracted from it to keep the number within a single digit.  
5. \*\*Summation\*\*: The sum of all transformed digits is calculated and added to the check digit.  
6. \*\*Validation Check\*\*: If the total sum is divisible by 10, the card number is considered valid.

## Program Execution:

When the script is executed directly, the program prompts the user to enter a card number. It then calls the `validate` method and prints whether the card is valid or invalid based on the result. This interaction is handled within the `if \_\_name\_\_ == "\_\_main\_\_":` block, which ensures that the input and output are only processed when the file is run as the main program, not when imported as a module.