# CMPE 202 Individual Project Part 2

## Kaustubh Kulkarni

## Student Id - 01451706

**Primary Problem**

To plan and design a powerful, scalable system which will return the appropriate credit card instance based on the data provided in different files. The system must categorize the data provided and check if the data belongs to any specific credit card and if it does, system should return the instance of that credit card to perform further operations on them.

**Secondary Problem**

The secondary problem that needs to be considered here is the system must be open for extension and closed for modification. That is the system should be able to accommodate new credit card types without having to change the existing code.

**Design Patterns and Consequence of Using these patterns**

To design the mentioned system following two design patterns can be used.

1. The chain of responsibility
2. Abstract Factory Pattern

**Chain of Responsibility Pattern**

1. The code designed with chain of responsibility pattern will determine if the given record is valid credit card type record and if it is valid credit card record, it will determine the type of credit card.
2. If the new category is added in credit card types, this pattern will accommodate the new type without changing the existing code.
3. This type increases request processing of new type very easily.

A picture containing map, text, screenshot

Description automatically generated

**Abstract Factory Pattern**

1. The code designed with factory pattern will return the instance of credit card which is identified by code designed with Chain of Responsibility pattern.
2. Abstraction is the main advantage of using this pattern. That is the code written to return the instance of asked type is isolated from client.

A screenshot of a map

Description automatically generated

A close up of text on a white background

Description automatically generated

**Consequence of Using these patterns**

If any new card factories must be included, then only code extension will work in this case. According to design pattern principles, the code should be open for extension and close for modification.

Chain of responsibility pattern is capable of finding weather the program can handle the type of input and factory pattern is returning the instance of appropriate Credit Card.

If chain of responsibility pattern can not handle the input then this situation is handled in default handler.

**Reader**

1. CreditCardReaderCSV, CreditCardReaderJSON, CreditCardReaderXML all these classes implement credetCardGeneratorReader.
2. The methods in these classes read the data from file and return the data in form of array list.

A screenshot of a cell phone

Description automatically generated

**Writer**

1. The CreditCardGeneratorWriterCSV, CreditCardGeneratorWriterJSON, CreditCardGeneratorWriterXML implements common interface CreditCardGeneratorWriter.
2. The method writeoutput writes the data in output file.

A screenshot of a cell phone

Description automatically generated

**General Program Flow**

A screenshot of a cell phone

Description automatically generated