Regular Expression Lab

Part 1

1. Create an application that takes asks for the following input items:
   1. First Name
   2. Last Name
   3. Street Address
   4. State
   5. Zip
   6. Phone number
   7. Email address
2. Using regular expressions, validate the input according to the following constraints:
   1. First Name – First letter capitalized, up to 12 characters
   2. Last Name – First letter capitalized, up to 16 characters
   3. Street Address – Number followed by street name
   4. State – Two capital letters
   5. Zip – Five digits with an optional dash and four additional digits
   6. Phone number – 000-000-0000 or (000)-000-0000 or 0000000000
   7. Email address – [name@companyname.(com](mailto:name@companyname.(com), org, or net)
3. If any input fails validation, it should re-prompt until correct input is received

Part 2

1. Extend your application to accept a credit card number and an expiration date
2. Use regular expressions to validate the number format of both the date and the card number
3. Use the rules below to validate the number and determine the card issuer:
   1. MasterCard: Must have a prefix of 51 to 55, and must be 16 digits in length
   2. Visa: Must have a prefix of 4, and must be either 13 or 16 digits in length
   3. American Express: Must have a prefix of 34 or 37, and must be 15 digits in length
   4. Diners Club: Must have a prefix of 300 to 305, 36, or 38, and must be 14 digits in length
   5. Discover: Must have a prefix of 6011, and must be 16 digits in length
   6. JCB: Must have a prefix of 3, 1800, or 2131, and must be either 15 or 16 digits in length
4. If the card data fails validation, it should re-prompt until good data is received
5. Once the program receives a valid number, it should print the customer’s name and the card issuer