CSCD 437
Lab 6
Regular Expressions

SPECIFICATIONS

Part 1 – regexone.com

Go to regexone.com and complete the all the lessons through Lesson 10: Starting and Ending. As you finish each lesson take a screen grab to prove you completed the lesson. Create a single PDF containing all members screen captures. The order of the PDF must be in alphabetical by last name. Presuming Stu Steiner and Tom Capaul are on team 16, the order would be Tom Capaul's screen captures and then Stu Steiner's screen captures. Clearly denote the individual team member's screen captures.

Name the PDF regexoneteam number.pdf (Example: regexoneteam16.pdf)

Part 2 – Java Regular Expressions

Write Java code using my provided Java file that contains main to implement regular expressions. The regular expressions you must write are specified below.

1. Easy Social Security Number (can be with dashes, whitespace, or no spaces at all)

2. Medium Easy Social Security Number (can be with dashes, whitespace, or no spaces at all) NOTE: dashes and whitespace will only happen between 3 and 4 and/or between 5 and 6
   - Some Valid Examples
     - 123456789
     - 123-45-6789
     - 123 45 6789
     - 123-45 6789
     - 123 45-6789
     - 12345-6789
   - Some NOT Valid Examples
     - 12-345-6789
     - 123--45-6789
     - 123.45.6789

3. New Social Security Number (can be with dashes, whitespace, or no spaces at all)
   NOTE: dashes and whitespace will only happen between 3 and 4 and/or between 5 and 6
   The Rules:
   - SSA will include all possible SSNs with the number "7" in position 1. Currently, SSNs that start with "7" are for certain states and other specific groups.
   - SSA will issue SSNs with the number "8" in position 1
   - SSA will not issue SSNs beginning with the number "9" unless the position 4 digit is a 9
   - SSA will not issue SSNs beginning with the number "666" in positions 1-3.
   - SSA will not issue SSNs beginning with the number "000" in positions 1-3.
   - SSA will not issue SSNs with the number "00" in positions 4-5.
   - SSA will not issue SSNs with the number "0000" in positions 6-9
   - The SSN must be exactly 9 digits
   - A null value is invalid

4. US Phone number – country code area code exchange code subscriber number
   - Parentheses are optional.
   - Dash is optional

        - Between area code and number
        - Between exchange code and subscriber number
    - +1 for country code is optional (must contain the + to be valid)
        - Space is the only valid optional character between country code and area code

5. E-mail address – Must contain @ and more than 1 character after the at symbol. A dot is optional in a string before the at symbol, but it is not allowed as the character right before the at symbol. A dot can't be used more than once consecutively. (Example: stu.steiner@ewu.edu is valid but stu.steiner.@ewu.edu is not valid)

6. Name on a class roster - Last name, First name, MI (presuming zero or more middle initials)

7. Date in MM-DD-YYYY format
    - You must account for:
        - Separators can be either the dash (-) or the forward slash (/) but not both
        - An optional leading zero.
        - A date that is a leap year date

8. House address - Street number, street name, abbreviation for road, street, boulevard or avenue (full version of those items should also be accepted)

9. City followed by state followed by zip as it should appear on a letter. A comma is optional between city and state. Capital letters are optional, except state must be 2 capital letters.

10. Military time, including seconds

11. US Currency down to the penny (ex: $123,456,789.23)

12. URL, optionally including http:// or https://, uppercase and lowercase should be accepted

13. A password that contains at least 10 characters and includes at least one upper case character, one lower case character, one digit, one punctuation mark, and does not have more than 3 consecutive lowercase characters

14. All words containing an even number of alphabetic characters, ending in "er"

NOTES

- I have provided CSCD437RegExMain.java and CSCD437RegExMethods.java.
    - The code in CSCD437RegExMethods is stubbed out but will compile.
    - The code is not packaged so don't add package statements
    - Write your code for the appropriate regular expression in the method provided.
    - You may write helper methods, but they must be declared as private.
    - You can't change my methods in any fashion
    - Your method will print valid if the string it is passed passes the regular expression or not valid if it does not.
    - You must print out the string in the method, so it is easy to see what is being tested
    - I have provided a everything but the regular expression code. You don't need to worry about extra whitespace etc.
- **Your testing MUST be thorough and robust -- you will be heavily penalized if this is not the case. You should have at least 5 tests per problem.**
- Make sure you document any cases you could not get to work.

- You will only submit your Java code with the results of your testing as a comment at the bottom of the file.
- ONCE AGAIN, BE SURE AND DOCUMENT ANYTHING YOU COULD NOT GET TO WORK, OR ANY SHORTCOMINGS FOR A GIVEN TASK.

WHAT TO TURN IN:

- A zip file containing:
  - regexoneteam number.pdf
  - CSCD437RegExMethods.java
  - CSCD437RegExMain.java
  - Ensure your team member names are at the top of the file.
  - I should be able to download your CSCD437RegExMethods.java and compile it with a different CSCD437RegExMain.java file and then run the code.

Name your zip team number lab6.zip (Example: team16lab6.zip)