

# CS105 : Computer Programming : PERL

## Project 04

Assigned on Wednesday, Mar 07  
Due on Saturday, March 19 at 11:59PM

### 1 Objectives

Through this project, we want to practise the usage of regular expressions in various situations. In this project, you are asked to implement some of functions with regular expressions. Basically, you start with the solution (i.e., *addressbook.pl*) of project03 and modify functions that contain “eq” or “==” so that regular expressions are used for pattern matching instead of “eq” or “==”. In addition, you are asked to implement new functions for processing required patterns. As before, you will use the two pragmas, “use warnings;” and “use strict;”.

### 2 Description

Implement all the following requirements:

1. The script will be called *addressbook\_regex.pl*.
2. As before, the script (i.e., *addressbook\_regex.pl*) will contain a global my hash variable called *addressbook*. Each record consists of name, address, and phone number. Also, note that no other global variables will be used for this project.
3. Note that you start with the solution (i.e., *addressbook.pl*) of project03.
4. Make sure your program should work just like *addressbook.pl*, even though you modify and add some functions in order to check some patterns using regular expressions.
5. *addressbook\_regex.pl* should not contain either “eq” or “==”, but contain appropriate regular expressions for checking patterns.

### 3 Grading Criteria & Policy

Note that the grade will be based on how well the required functionality for each subroutine is fulfilled. Therefore, please try to follow the description as much as possible to get full credits.

1. “while (1) { }” (**20 points**): Modify the “if ()” and “elsif ()” so that the appropriate patterns can be processed. Try to implement your code as short as possible. For example, the given code uses “lc()” many times, but you can get rid of it using regular expression.
2. “get\_content()” (**60 points**): The given code doesn’t check validity of data. So, in this project, you are asked to type again until unless input is valid.
  - name: The name should not contain any numbers or functuation marks (except period(.)).
  - address: It should be the format of “cityName, stateName ZIPcode”, where cityName is an usual city name, stateName should be abbreviation (i.e. two-word XX) of one of 50 states, and ZIPcode should be of five-digit number (i.e., DDDDD).
  - phone: It should be the format of either “DDD-DDD-DDD” or “(DDD)DDD-DDDD”. Note that if the input is the format of “DDD-DDD-DDD”, convert it into the format of “(DDD)DDD-DDDD”.

To check the validity, implement the three new functions: “check\_name()”, “check\_address()”, and “check\_phone()”.

3. “delete\_all()” (**20 points**): Modify the “if ()” with appropriate regular expression. Try to implement it as short as possible.
  - Please make sure to include the following header in the code (“addressbook\_regex.pl”):

```
# Project = 04
# Description = Address book using regular expressions
# Program = addressbook_regex.pl
# Input = address.txt
# Output =
# Usage = perl addressbook_regex.pl (or perl addressbook_regex.pl address.txt)
# Name = Your name
# UT EID = Your eid
# Comments (or README) = Describe some thing that you wish the grader to know.
# For example, What do you like/dislike about the project? What was your challenge
# for this project?
# Perl code will be followed.
```

- This project is worth a total of 100 points. Make sure your program has no warnings and errors.
- You will need to submit one file for this project: “addressbook\_regex.pl”. Submit the files as follows:
 

```
>turnin -submit hyukcho project04 addressbook_regex.pl
```

Good luck!