

# CS105 : Computer Programming : PERL

## Project 03

Assigned on Wednesday, Feb 21  
Due on Friday, March 02 at 11:59PM

### 1 Objectives

Through this project, we want to review most of the basic concepts in Perl (except regular expressions) and make use of the basic programming techniques introduced in the textbook. As before, we will be implementing all perl programs with subroutines. Also, we will use the two pragmas, “use warnings;” and “use strict;”.

### 2 Description

We will implement all the following requirements:

1. The script will be called *addressbook.pl*.
2. The script (i.e., *addressbook.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. It will contain the following functionalities (Details of each subroutine will be described in sequel): *add\_one*, *add\_all*, *delete\_one*, *delete\_all*, *view\_one*, *view\_all*, and *exit*
4. If the script is executed from the commandline as  
>perl addressbook.pl  
then, show the following menu on the screen (Hint: you can create the menu using a *here document*):

Please Make a choice:

- (1) Add One entry (A0)
- (2) Add All entries (AA)
- (3) Delete One entry (DO)
- (4) Delete All entries (DA)
- (5) View One entry (VO)
- (6) View All entries (VA)
- (7) Exit/Quit (E/Q)

Your choice:

To be specific, the user will select one of the menu items (either one number among 1 to 7 or the corresponding acronym on the right). Until the user selects number 7 (or E/Q, not case-sensitive), the program will go into a loop to redisplay the menu and wait for the user to select another transaction.

5. If the script is executed from the commandline as  
`>perl addressbook.pl address.txt`  
then, it first stores the global my hash variable, *addressbook*, with the contents of the given data file (i.e., address.txt in this example). Note that you will call the function called “add\_all()”, which is also used for (2) Add All entries (AA) in the menu. After then, the menu needs to be displayed.
6. After every transaction, the following information needs to be printed out on the screen:  
Currently, the address book contains *n* entry (or entries).  
Note that the form (i.e., either singular or plural form) of “entry” should be in accordance with the number *n*.

### 3 Grading Criteria & Policy

Here are the detail description of each subroutine in the menu. 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. “process\_file()” (**20 points**) takes a filename as a parameter and stores the global my hash variable *addressbook* with its contents. Each line in a given file contains the following information (i.e., one record):  
name:address:phone  
In other words, name, address, and phone are separated with colon(:). Basically, you should do the following: (1) first open the file (use “open()” and “die()”); (2) get each line (use “while loop” and “chomp()”); (3) get name, address, and phone from the line (use “split()” or other function that you like); (4) add name, address, and phone to *addressbook*; (5) close the file handle; and (6) meanwhile, you need to keep track of the number of records stored and print it out on the screen.  
Note that “process\_file()” will be called when a data filename is specified at the commandline or when menu 2 (or AA) is selected.
2. “menu()” (**20 points**) prints out the menu on the screen. For this, you can use “print()” or *here document*.
3. Implement the main menu control structure (**20 points**) processes the all the menu selection, braches to each menu, and redisplay the menu. For example, “if” and “elsif” can be used within a infinite “while” loop.
4. “add\_one()” (**20 points**) gets a record (i.e., name, address, and phone) from STDIN and add the content into the global hash variable *addressbook*. Note that the following message should be print out on the screen if the name already exists in the hash:

Bill already exists in the address book!

Note that the record will be overwritten in case it already exists.

5. “add\_all()” (**20 points**) takes a file name to read from STDIN and store the contents into *addressbook*. It should do the following: (1) print “Enter filename: “ on the screen; (2) get a filename from STDIN; and (3) pass the filename to “process\_file()”. and access mode.
6. “delete\_one()” (**20 points**) gets a name to delete from STDIN and delete it from *addressbook* if it already exists. Otherwise, print out the following message on the screen:  
Bill is not in the address book!
7. “delete\_all()” (**20 points**) clears the contents of *addressbook*. Do the following: (1) ask “Do you really want to delete all names? (Y[es]/N[o]): “ and (2) delete all the contents if the answer is “Yes” or “Y”. Otherwise, do nothing.
8. “view\_one()” (**20 points**) shows the address and the phone for a requested name. To do this, do the following: (1) ask “Enter name to view: “; (2) and (2) print out the contents as follows if it already exists:  
Name: Bill  
Address: Austin, TX  
Phone: (512)111-2222  
Otherwise, print out “Bill is not in the address book!”.
9. “view\_all()” (**20 points**) shows the all records in *addressbook* in name key order. Print out each record as in “view\_one()”.
10. Program style (**20 points**):

- Please make sure to include the following header in the code (“addressbook.pl”):

```
# Project = 03
# Description = Address book
# Program = addressbook.pl
# Input = address.txt
# Output =
# Usage = perl addressbook.pl (or perl addressbook.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 200 points. Make sure your program has no warnings and errors.

- You will need to submit one file for this project: “addressbook.pl”. Submit the files as follows:  
>turnin -submit hyukcho project03 addressbook.pl

Good luck!