

CS105 : Computer Programming : PERL

Project 02

Assigned on Wednesday, Feb 21
Due on Monday, Feb 26 at 11:59PM

1 Objectives

Through this project, we will review the following contents that we have learned up to now: scope (of variables) and subroutine in chapter 4, filehandles in chapter 5, control structures, and file tests in chapter 11. From this project, we will be implementing all perl programs with subroutines. We start with the same program code that we implemented in “Project01”. However, we will be modifying the code so that it doesn’t generate any warning messages with “use warnings;” and “use strict;” directives.

2 Description

For this project, first download the same code (“computeFrequency.pl”) and the same data file (“sequence.txt”) that we used for Project01 from the class webpage. We will do all the followings with subroutines:

- Take both input and result files from the commandline
- Test both the files for existence
- Open input file
- Count frequency of each character
- Close input file
- Open result file
- Print result
- Close result file

3 Grading Criteria & Policy

We will modify the given code (i.e., “computeFrequency.pl”) and the modified code should be called “computeFrequency3.pl”. We will need the following subroutines. However, more subroutines can be defined if necessary.

1. “take_commandline()” (20 points): It takes commandline arguments as inputs and checks whether both the input and the result file names are specified. If both the file names are not specified, show the usage of your program as below and exit the program (Note that both the files needs to be specified),
> USAGE: ./computeFrequency3.pl sequence.txt result3.txt
2. “test_existence()” (20 points): It takes a file name and tests its existence. If it exists, prompt the following message on the screen and exit the program (For example, the filename “result3.txt” is typed.),
> Oops! A file called result3.txt already exists.
If not, prompt the following message on the screen and do nothing:
> A file called result3.txt doesn’t exist and it will be created.
3. “open_file()” (10 points): It takes a file name and access mode.
4. “count_frequency()” (10 points): It basically does count frequency of each character.
5. “close_file()” (10 points)
6. “print_result()” (10 points)
7. Correctness (10 points): (Usage of “use warnings;” and “use strict;” will be checked.)
8. Program style (10 points):

- Please make sure to include the followings the code (“computeFrequency3.pl”):

```
# Project = 02
# Description = computes frequency of a character (using subroutines)
# Program = computeFrequency3.pl
# Input = sequence.txt
# Output = result3.txt
# Usage = perl computeFrequency3.pl sequence.txt result3.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 with “use warnings;” and “use strict;” directives.
- You will need to submit one file for this project: “computeFrequency3.pl”. Submit the files as follows:
>turnin -submit hyukcho project02 computeFrequency3.pl

Good luck!