

PERL Lab Manual

Introduction to Lab Exercises

The lab exercises are designed to make you understand how to write basic PERL script. They are organized in a way that takes you step by step until you become familiar with the language.

Over this period, you will gain the level of scripting skill, syntax proficiency, and understanding that can only be achieved through meaningful practice and effort.

There are 5 labs and every lab has 3 files:

- Lab_Template
- Lab_Exercise
- Readme.txt

PERL version is v5.8.8 and supports Linux architecture i386 and i486.

How to identify a PERL script ?

A file is identified as a PERL script by 2 ways :

- 1) File will have .pl as the file extension .
- 2) The first line of the file will have the following non-executable statement as:
#!/path to perl

Steps below to create a PERL script and execute the same :

- 1) Create a file "**f1**" using command **#vi f1**
- 2) Make sure you add the line **#!/usr/bin/perl** at the start line of the script where **/usr/bin/perl** is the path to "**perl**" binary.
- 3) After writing the body of the script, save the file f1 by pressing the **ESC key** followed by: **w**
- 4) Then quit and move to the shell prompt by pressing the **ESC key** followed by: **q**
- 5) Then check the file permission on the Perl script using the command: **ls -l**
- 6) The default permission on the file will be: **-rw- --- ---**
- 7) Change the mode the file by using the following command :

#chmod +x f1

8) Execute the script by using the following command : #./f1

LAB1:

Write a simple PERL script that should print the string "Hello World".

Exercise:

Write a Perl script that prints your name and your area of interest in VLSI Domain and run the script.

LAB2 :

Write a script that will calculate the circumference of a circle.

Exercise :

Create a copy of the perl script "circle" under lab2 and rename it to circle_1. Then modify the new script "circle_1" such that it prompts for and accept a radius from the person running the program and then calculates the circumference.

LAB3:

Write a script that prompts for and reads two numbers and prints out the product of the two numbers multiplied together in separate lines.

Exercise:

Write a script that prompts for and reads a string and a number (on separate lines of input) and prints out the string the number of times indicated by the number on separate lines. (Hint: Use the "x" operator) .If the user enters "fred" and "3", the output should be three lines, each saying "fred".

LAB4:

Write a script that reads a list of strings on separate lines until end of input and prints out the lists in reverse order.

Exercise:

Write a script that reads a list of numbers (on separate lines) until end of input and then prints for each number the corresponding person's name from the list shown below (fred, wilson, kelly, lewis, andrew). For example :if the input numbers were 1,2,4,3 then the output names would be fred, wilson, lewis, Kelly

LAB5:

Write a subroutine, called "total", which returns the 4 times value of the same number for first 10 integers. The subroutine shouldn't perform any I/O. It should simply process its parameters and return a value to its caller. Try this out in this sample program, which merely exercises the subroutine to see that it works.

Exercise:

Calculate Simple interest for year 2000. Assume Principal = 10000, time = 5 years, rate = 10%. (Hint) $SI = PTR/100$.