Name: Kerolos Youssef Ghobrial

# LAB 1: Perl

1-

```perl
# number 1
print "Hello World!\n";
```

```
Hello World!
```

2-

```perl
# number 2
foreach my $arg (@ARGV) {
    print "Argument: $arg\n";
}
print "Argument: @ARGV \n";
```

3-

```perl
# number 3
$rad = 12.5;
$circle_cir = 2 * 3.14 * $rad;
print "Circle Circumference = ",$circle_cir,"\n";
```

```
Circle Circumference = 78.5
```

4-

```perl
# number 4
print ("Enter the radius of Circle: \n");
$rad= <STDIN>;
$Circumference = 2 * 3.14 * $rad;
print "Circle Circumference = ", $Circumference, "\n";
```

```
Enter the radius of Circle:
5
Circle Circumference = 31.4
```

5-

```perl
# number 5
print ("Enter the First Number: \n");
my $num1 = <STDIN>;
print ("Enter the second Number: \n");
my $num2 = <STDIN>;
my $sum = $num1 * $num2;
print "sum of numbers = ", $sum, "\n";
```

```
Enter the First Number:
5
Enter the second Number:
6
sum of numbers = 30
```

6-

```perl
# number 6
print "Write a string \n";
$a = <STDIN>;
chomp $a;
print "Write a number \n";
$b = <STDIN>;
chomp $b;
for ($i = 1; $i <= $b; $i++) {
print "$a \n";
}
```

```
Write a string
kero
Write a number
5
kero
kero
kero
kero
kero
```

7-

```perl
# number 7
print "Write a list \n";
@c = <STDIN>;
@C = reverse(@c);
print "reversed list is: @C \n";
```

```
Write a list
kero
iti
linux
perl
reversed list is: perl
 linux
 iti
 kero
```

8-

```
# number 8
print "Write a list \n";
@c = <STDIN>;
print "Write a number \n";
$b = <STDIN>;
$a = $c[$b];
print "string at number $b is $a \n";
```

```
Write a list
kero
iti
linux
perl
Write a number
1
string at number 1
 is iti
```

9-

```
# number 9
print "Write a list \n";
@c = <STDIN>;
print "random string is:\n $c[rand @c]";
```

```
Write a list
kero
iti
linux
perl
random string is:
 perl
```

10-

```
print "what is the temperature outside? \n";
$temp = <STDIN>;

if ($temp > 35) {
    print "too hot \n";
 }
else {
      print "too cold \n";
 }
```

```
what is the temperature outside?
43
too hot
```

11-

```
print "what is the temperature outside? \n";
$temp = <STDIN>;

if ($temp >= 35) {
    print "too hot \n";
 }
 elsif ($temp <= 28) {
      print "too cold \n";
 }
 else {
     print "just right! \n";   #if temp > 28 & temp < 35
 }
```

```
what is the temperature outside?
35
too hot
```

12-

```
$total = 0;
while (1) {
print "Enter a number (or 999 to finish): ";
$input = <STDIN>;

last if $input == 999; # Exit the loop when 999 is entered

$total += $input; # Add the number to the total
}

print "Total of all numbers (excluding 999) = " ,$total,"\n";
```

```
Enter a number (or 999 to finish): 99
Enter a number (or 999 to finish): 135
Enter a number (or 999 to finish): 999
Total of all numbers (excluding 999) = 234
```

13-

```
# number 13
@string_list;
print "Enter strings (one per line, press Enter to finish):\n";

# Read strings and store them in an array
while ($input_string = <STDIN>) {
last if $input_string eq " ";
push @string_list, $input_string;
 }

# Print the strings in reverse order
print "Reversed list of strings:\n";
 for ($i = $#string_list; $i >= 0; $i--)
    {
        print "$string_list[$i]\n";
    }
```

```
Enter strings (one per line, press Enter to finish):
kero
iti
linux
perl

Reversed list of strings:


perl

linux

iti

kero
```

14-

```perl
# number 14
for ($a = 0; $a <= 32;$a++)
{
    $b = $a**2;
 printf "%5g %8g \n",$a,$b;
}
```

```
    0        0
    1        1
    2        4
    3        9
    4       16
    5       25
    6       36
    7       49
    8       64
    9       81
   10      100
   11      121
   12      144
   13      169
   14      196
   15      225
   16      256
   17      289
   18      324
   19      361
   20      400
   21      441
   22      484
   23      529
   24      576
   25      625
   26      676
   27      729
   28      784
   29      841
   30      900
   31      961
   32     1024
```

15-

```
# number 15
 %map= (
      "red" => "apple" ,
      "green" => "leaves" ,
      "blue" => "ocean"
      );
 print "Enter a string: ";
 $str = <STDIN>;      #read input from user
 chomp ($str) ;
 print "The value for $str is $map{$str}\n";
```

```
Enter a string: green
The value for green is leaves
```

16-

```
# number 16

$c = "i am kerolos and i am a fresh graduate";
@a = split / /, $c;
foreach $a(@a)
{
    $count{$a}++;
}
foreach $a(sort keys %count)
{
    print  $a, " ", $count{$a},"\n";
}
@b = sort (@a);
print "sorted words:\n @b";
```

```
a 1
am 2
and 1
fresh 1
graduate 1
i 2
kerolos 1
sorted words:
 a am am and fresh graduate i i kerolos
```

17-

```
# number 17
 opendir($dir, ".") or die "Cannot open directory: $!";

# Read and print the directory contents
 while ( $entry = readdir($dir)) {
     next if ($entry eq "." || $entry eq ".."); # Skip '.' and '..' entries
     print "$entry\n";
 }

 closedir($dir);
```

18-

```
# number 18
 $c = kero_youssef;   #correct username
 $d = kj1567813556;      #correct password
 print "your username is: ";
 $a = <STDIN>;
 print "your password is : ";
 $b = <STDIN>;
 if ($a == $c && $b == $d)
 {
     print "Welcome,kero!"
 }
 else
 {
     print "your username or password may not be correct, try again";
 }
```

```
your username is: kero_youssef
your password is : kj1567813556
Welcome,kero!
```

19-

```
# number 19
use strict;
use warnings;
use POSIX qw(strftime);

my ($sec, $min, $hour, $mday, $mon, $year, $wday) = localtime();

# Adjust the month and year values
 $mon++;      # Add 1 to the month since it's 0-based (0 = January, 1 = February, and so on)
 $year += 1900;  # Add 1900 to the year to get the current year

# Get the day of the week
 my $day_name = strftime("%a", localtime()); # (%A displays Sunday) , (%a displays Sun)

# Array of month names
 my @month_names = qw(Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec);

# Get the month name
 my $month_name = $month_names[$mon - 1];

# Print the current date and time
 print "Current date and time: $day_name $month_name $mday $hour:$min:$sec $year\n";
```

```
Current date and time: Sun Nov 5 20:50:1 2023
```