**EXPERIMENT NO – 08**

**Title:**

Ruby programs on variable declarations, control structure and looping.

#VARIABLE DECLARATIONS

#1-local variables (starts with "\_" or smallcase latter)

var1=10

var2=20

\_add=var1+var2

print("addition is=",\_add)

a = 124.24

b = 127.56

print("\naddition =",a+b)

#2-class variables

=begin

A class variable name starts with "@@" sign. They need to be initialized before use.

A class variable belongs to the whole class and can be accessible from anywhere inside the class.

If the value will be changed at one instance, it will be changed at every instance.

=end

#3-instance variables

=begin

An instance variable name starts with a "@" sign. It belongs to one instance of the class and can be accessed from

any instance of the class within a method. They only have limited access to a particular instance of a class.

They don't need to be initialize. An uninitialized instance variable will have a nil value.

=end

#4-global variables

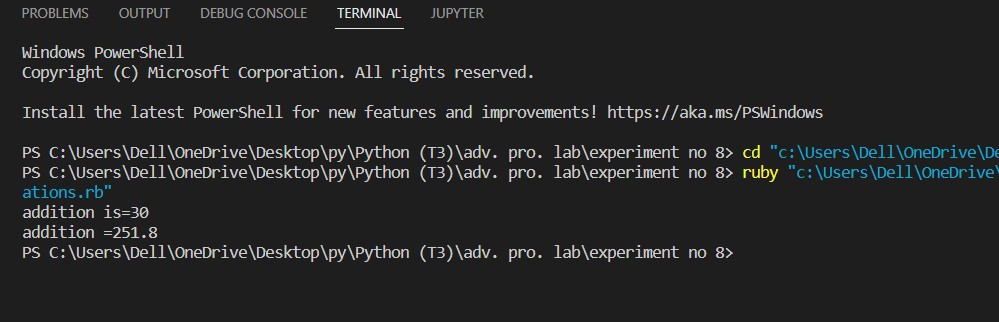
=begin

A global variable name starts with a "$" sign. Its scope is globally, means it can be accessed from any where in a program.

An uninitialized global variable will have a nil value. It is advised not to use them as they make programs cryptic and complex.

=end

OUTPUT:



1==> #if

age = 26

if (age >= 18)

  puts "You are eligible to get your driver's license."

end

# if-else

age = 19

if (age >= 18)

  puts "you can vote"

else

  puts "you are not eligible for vote"

end

#if-elseif-else

x = 1

if x > 2

   puts "x is greater than 2"

elsif x <= 2 and x!=0

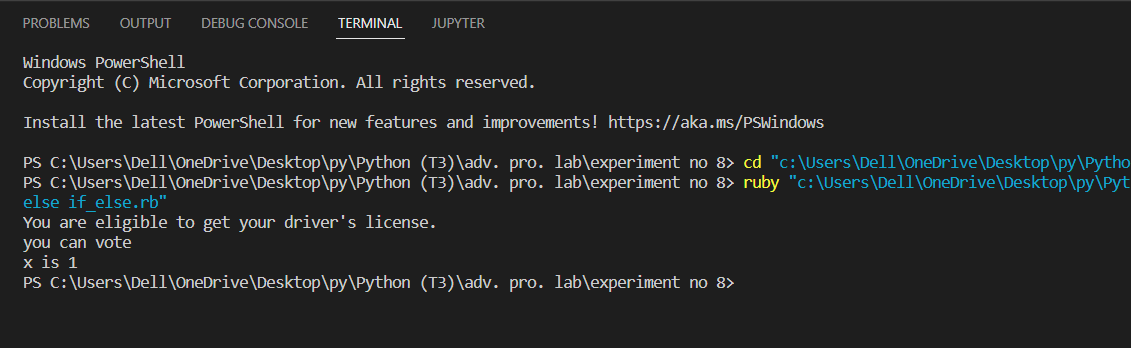
   puts "x is 1"

else

   puts "value of x is",x

end

output:

****

