L2 – Python - Hands on

Name : yanamandra sarath kumar

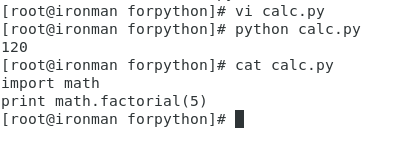
AD ID : SA374449

Project : CITI bank

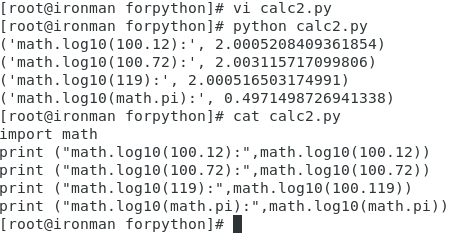
Mail ID : [sarathkumar.yanamandra@wipro.com](mailto:sarathkumar.yanamandra@wipro.com)

1.Create file called “calc.py" which has following functions a. function to find the factorial of a number b. function to find the log10 of a number c. function to convert degrees to radians d. The sin, cos and tan trigonometric functions

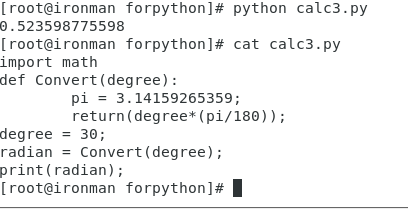
a) function to find the factorial of a number



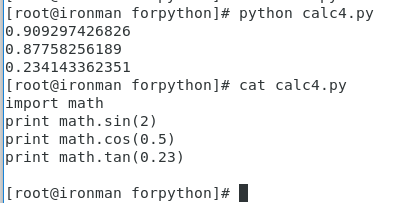
b) function to find the log10 of a number



c). function to convert degrees to radians



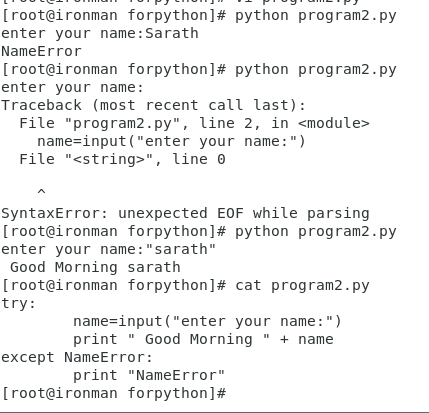
d). The sin, cos and tan trigonometric functions



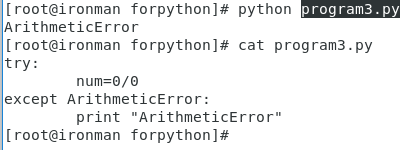
2. Write a program to handle the following exceptions

a) KeyboardInterrupt,

b) NameError

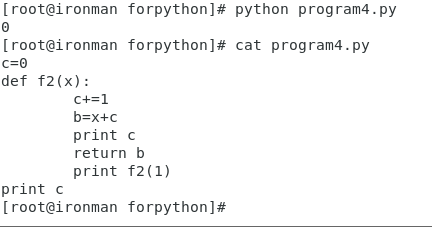


c) ArithmeticError



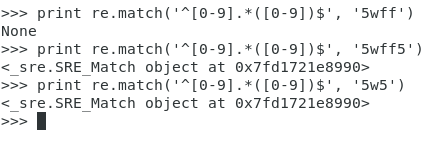
3. How do you handle exception for the following code?

c = 0 def f2(x): c+= 1 b = x + c print c return b print f2(1) print c

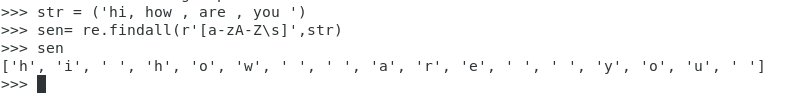


4. Translate each of the following English statements into a regular expression:

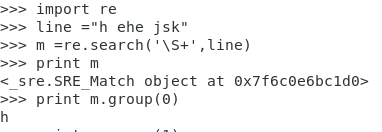
a. digit at the beginning of the string and a digit at the end of the string



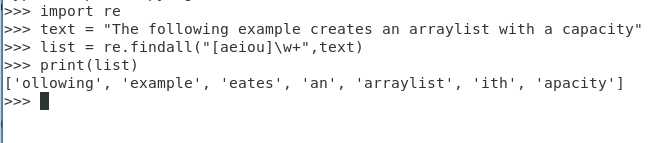
b. A string that contains only whitespace characters or word characters



c. A string containing no whitespace characters



5. Write a program using re module that loops through the lines of a file or standard input (where each line contains a single word) and prints all words containing two adjacent vowels



6.Implement a child class called mathnew and parent classes as sqroot, addition, subtraction, multiplication and division. Use the super () function to inherit the parent methods.

**Code:**

from math import \*

from functools import reduce

class sqroot:

def \_\_init\_\_(self,n1):

self.n1 = n1

def sqroot(self):

return(sqrt(self.n1))

class addition:

def \_\_init\_\_(self,n2):

self.n = n2

def add(self):

return(sum(self.n))

class subtraction:

def \_\_init\_\_(self,n3,n4):

self.n1 = n3

self.n2 = n4

def sub(self):

return((self.n1-self.n2))

class multiplication:

def \_\_init\_\_(self,n5):

self.n = n5

def product(self):

return((reduce(lambda x,y: x\*y, self.n)))

class division:

def \_\_init\_\_(self,n6,n7):

self.n1 = n6

self.n2 = n7

def div(self):

return("The quotient is :- {}".format(self.n1/self.n2))

class mathnew(sqroot,addition,subtraction,multiplication,division):

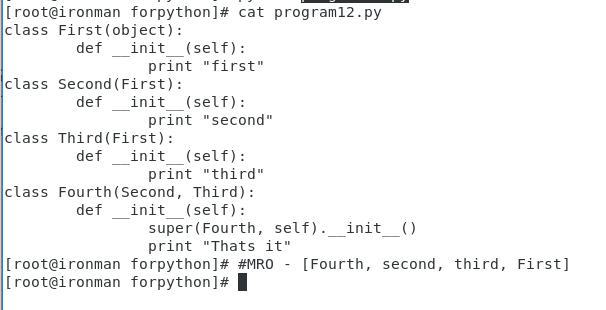
def \_\_init\_\_(self):

super().\_\_init\_\_()

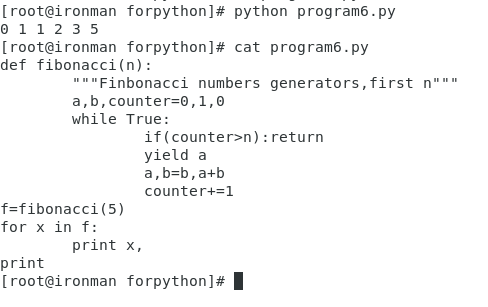
d1 = mathnew(4)

print(d1.sqroot(4))

7. Create a class called First and two classes called Second and Third which inherit from First. Create class called Fourth which inherits from Second and Third. Create a common method called method1 in all the classes and provide the Method Resolution Order



8. Implement a simple generator for Fibonacci series



9. Write an iterator class that iterators over a sequence of values in the reverse direction

10. Implement a decorator that quantifies and returns the execution time of any function

