Orange County Community College

Computer Science and Technology Department

csc 138 - Scripting

Laboratory Exercise 6

Name: ____Alex DeGhetto_____

Date: _____

Objectives:

- Demonstrate the ability to write and call functions in Python.
- Create and utilize Python modules.
- Demonstrate an understanding file IO
- Understand the import directive and Python's module structure
- Use import to include modules in code

Procedures:

1. Write a program that uses a function to calculate the volume of a box. Prompt the user for the length, width, and height of the box, then call your function and print the result. In your main program, allow the user to enter data for several different boxes, calling your volume function on each set of values. Use a sentinel value or similar method to terminate user input. Attach the source code for your program to this lab sheet.

```
#!/usr/bin/python3
#setting avariable to run script on loop
runScript = True
#making function for volume
def vol(l, w, h):
  vol = float(l * w * h)
  return vol
  #starting script
while(runScript):
  #prompting user for values of length, width, height
  print("Enter the Height, Width, and Length")
  l = float(input("Length: "))
  w = float(input("Width: "))
  h = float(input("Height: "))
  print("Volume:", vol(l, w, h))
  #asking user if they would like to find another volume
  userInput = input("Would you like to find the volume of another object? (y/n): ")
  #if statements for continuing or ending script
  if userInput == "y" or userInput == "Y":
    runScript = True
  else:
    if userInput == "n" or userInput == "N":
      runScript = False
```

```
  else:
    print("You did not enter y/n")
    runScript = False
```

2. Create a program that counts the number of digits in an integer using mathematical operations (do not convert the integer to a string!). Prompt the user for a number and read it into an integer variable. Pass the integer into a function that determines how many digits the integer has. For example, if the user enters the number 43432, the function should return 5. Attach the source code for your program to this lab sheet.

```
#!/usr/bin/python3
#variable to run script on loop
runScript = True
#making function for len of int
def cnt(userInput):
 num = len(userInput)
 return num
#starting script
while(runScript):
 #asking user for integer
 num = input("Input an integer: ")
 print("Length of Integer: ", cnt(num))
 #asking user if they would like to enter another int
 userInput = input("Would you like to enter another integer? (y/n): ")
 if userInput == "y" or userInput == "Y":
  runScript = True
 else:
  if userInput == "n" or userInput == "N":
   runScript = False
  else:
   print("You did not enter y/n")
   runScript = False
```

3. Modify the program you wrote above to reverse the integer entered. For example, if the user typed the number 2468, the function should return the integer 8642. Attach the source code for your modified program to this lab sheet.

```
#!/usr/bin/python3
#variable to run script on loop
revr_num = 0
runScript = True
#making function for  of int
def rvs(num):
 global revr_num   # We can use it out of the function
 if (num > 0):
  Remainder = num % 10
```

```
    revr_num = (revr_num * 10) + Remainder
    rvs(num // 10)
    return revr_num
#starting script
while(runScript):
  #asking user for integer
  num = int(input("Input an integer: "))
  print("Reversed Number: ", rvs(num))
  #asking user if they would like to enter another int
  userInput = input("Would you like to enter another integer? (y/n): ")
  if userInput == "y" or userInput == "Y":
    runScript = True
  else:
    if userInput == "n" or userInput == "N":
      runScript = False
    else:
      print("You did not enter y/n")
      runScript = False
  revr_num = 0
  number = 0
  Remainder = 0
```

4. Write a program that calculates the summation of integers from one to N. Allow the user to input an integer, then pass that number to a function that will calculate the sum of all integers up to N. For example, if the user inputs the number 5, then the function will return the sum 1+2+3+4+5. Allow the user to enter multiple integers, calling the function for each number. Use a sentinel value or similar method to terminate user input. Attach the source code for your program to this lab sheet.

```
runScript = True
def sumOf(userInput):
  x = 0
  while x <= userInput:
    print(x)
    x += 1
  #sum = (userInput * (userInput + 1) // 2)
  #print("Sum of number: ", sum )
  return sum
while(runScript):
  n = int(input("Enter a number: "))
  sumOf(n)
  userInput = input("Would you like to enter another integer? (y/n): ")
  if userInput == "y" or userInput == "Y":
    runScript = True
  else:
    if userInput == "n" or userInput == "N":
      runScript = False
    else:
      print("You did not enter y/n")
```

```
runScript = False
```

5. Research the *Babylonian method* for determining the square root of a real number. Consult the following web sites (or similar pages):

- http://mathlesstraveled.com/2009/05/18/square-roots-with-pencil-and-paper-the-babylonian-method/
- http://en.wikipedia.org/wiki/Methods_of_computing_square_roots
- http://www.mathpages.com/home/kmath190.htm

Use this algorithm to write a function that takes a number as its argument and returns the square root of that number. Write a complete Python program that calls this function. Attach the source code for your program to this lab sheet.

```
import math
runScript = True
def square(userInput):
  square = math.sqrt(userInput)
  print(f"The Square Root of {userInput} is {square}")
  return square

while(runScript):
  num = int(input("Enter A Number You Want The Square Root To: "))
  square(num)
  userInput = input("Would you like to enter another integer? (y/n): ")
  if userInput == "y" or userInput == "Y":
    runScript = True
  else:
    if userInput == "n" or userInput == "N":
      runScript = False
    else:
      print("You did not enter y/n")
      runScript = False
```

6. Create a Python module that contains the functions you created in numbers 1-5 above. Attach the source code for your module to this lab sheet. What is the name of your Python module?

(SAVED AS mymodule.py)

```
import math

def square(userInput):
  square = math.sqrt(userInput)
  print(f"The Square Root of {userInput} is {square}")
  return square

def vol(l, w, h):
  vol = float(l * w * h)
  return vol

def sumOf(userInput):
  x = 0
  while x <= userInput:
    print(x)
    x += 1
  #sum = (userInput * (userInput + 1) // 2)
  #print("Sum of number: ", sum )
```

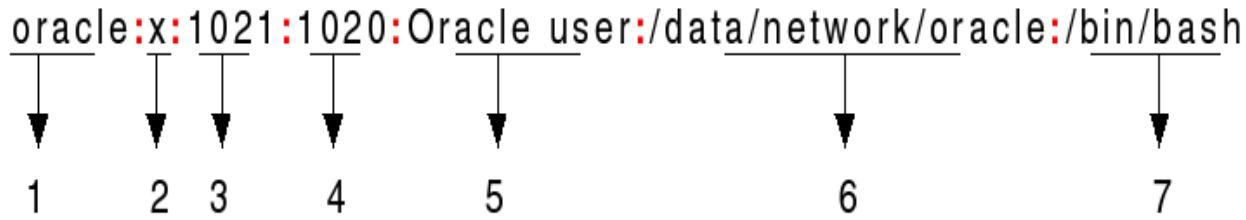
```
 return sum

revr_num = 0
runScript = True
#making function for  of int
def rvs(num):
 global revr_num   # We can use it out of the function
 if (num > 0):
  Remainder = num % 10
  revr_num = (revr_num * 10) + Remainder
  rvs(num // 10)
  return revr_num
def cnt(userInput):
 num = len(userInput)
 return num

def yesOrNo(userInput):
 if (userInput == "n" or userInput == "N"):
    return False
 else:
  if (userInput != "y" and userInput != "Y"):
   print("You didnt enter a y or an n")
  return True
```

File IO

1. create a program that will open the /etc/passwd file on your system for reading (**or create a file with the attached content at the end**)



1. **Username**: It is used when user logs in. It should be between 1 and 32 characters in length.
2. **Password**: An x character indicates that encrypted password is stored in /etc/shadow file. Please note that you need to use the passwd command to computes the hash of a password typed at the CLI or to store/update the hash of the password in /etc/shadow file.
3. **User ID (UID)**: Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups.
4. **Group ID (GID)**: The primary group ID (stored in /etc/group file)
5. **User ID Info**: The comment field. It allow you to add extra information about the users such as user's full name, phone number etc. This field use by finger command.
6. **Home directory**: The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /
7. **Command/shell**: The absolute path of a command or shell (/bin/bash). Typically, this is a shell. Please note that it does not have to be a shell.

a) Create a dictionary from this file storing the username as a key and userId(UID) as values

b) Write this dictionary to a new file in your home directory storing the userid(UID) as comma separated values.

Ex. root,0
daemon,1

```
x = open("./pass.txt", "r")
y = x.read().split("\n")
x.close()
dict = {}
for var in y:
  newVar= var.split(":")
  dict.update({newVar[0]:newVar[2]})
file = open("../text.csv", "w")
finalValue = ""
for keys, values in dict.items():
  finalValue += keys + "," + values + '\n'
file.write(finalValue)
file.close()
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
```

messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
uuidd:x:105:111::/run/uuidd:/usr/sbin/nologin
avahi-autoipd:x:106:112:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
dnsmasq:x:108:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
rtkit:x:109:114:RealtimeKit,,,:/proc:/usr/sbin/nologin
speech-dispatcher:x:110:29:Speech Dispatcher,,,:/var/run/speech-dispatcher:/bin/false
whoopsie:x:111:117::/nonexistent:/bin/false
kernoops:x:112:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:113:119::/var/lib/saned:/usr/sbin/nologin
pulse:x:114:120:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
avahi:x:115:122:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
colord:x:116:123:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
hplip:x:117:7:HPLIP system user,,,:/var/run/hplip:/bin/false
geoclue:x:118:124::/var/lib/geoclue:/usr/sbin/nologin
gnome-initial-setup:x:119:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:120:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
occc:x:1000:1000:occc,,,:/home/occc:/bin/bash
mysql:x:121:127:MySQL Server,,,:/nonexistent:/bin/false
sshd:x:122:65534::/run/sshd:/usr/sbin/nologin
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
cups-pk-helper:x:123:116:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nol
ogin
munge:x:124:129::/nonexistent:/usr/sbin/nologin
statd:x:125:65534::/var/lib/nfs:/usr/sbin/nologin