Lab1

#! /usr/bin/python

#\_\_author\_\_ = "Tadhg Foley"

#\_\_credits\_\_ = ["Tadhg Foley"]

#\_\_license\_\_ = "GPL"

#\_\_version\_\_ = "1.0.0"

#\_\_maintainer\_\_ = "Tadhg Foley"

#\_\_email\_\_ = "tadhg.foley@mycit.ie"

#\_\_status\_\_ = "Sexy"

def dec\_to\_bin(s) : #Creates a Function

ret ="" #Variable for res

if s == 0 : #If Statement

return "0" #Returns 0

while int(s) > 0 : # while statement for while s is greater than 0

ret = str(int(s) % 2) + ret

s = int(s) >> 1

return ret #Returns ret ,

def check\_dec(s) : #Creates a function

for i in range(len(s)) : #bring back true if s is a decimal number false otherwise

if s[i] == int : #

return False #

return True #

decimal\_string = raw\_input("Put in Decimal Number ") #tells the user to enter a decimal

if check\_dec(decimal\_string) : #Checks decimal for if its true

print decimal\_string + " in binary is " + dec\_to\_bin(decimal\_string) #Displays the binary of the value

else :

print decimal\_string + "thats not a decimal number " #displays if not a decimal

Lab 2

#!/usr/bin/python

#\_\_author\_\_ = "Tadhg Foley"

#\_\_credits\_\_ = ["Tadhg Foley"]

#\_\_license\_\_ = "GPL"

#\_\_version\_\_ = "1.0.0"

#\_\_maintainer\_\_ = "Tadhg Foley"

#\_\_email\_\_ = "tadhg.foley@mycit.ie"

#\_\_status\_\_ = "Sexy"

def front\_z(words): #creates function

list\_m = [] #creates list that will contain words with z

list\_nom = [] #cretes other list

for word in words: #'for' loop that looks at words

if word.startswith('z'): #checks if word starts with 'z'

list\_m.append(word) #puts 'z' word into appropiate list

list\_m.sort() #sorts the 'list\_m' list

else:

list\_nom.append(word) #puts other words into the second list

list\_nom.sort() #sorts this list

return list\_m + list\_nom #combines the output of lists with

#list\_m first

print front\_z(['mix', 'apple', 'aardvark', 'zzzz', 'zebra'])

#prints the given words to use in function above

Lab 4

#!/usr/bin/python

# x = input from File

# y = output of file from

#\_\_author\_\_ = "Tadhg Foley"

#\_\_credits\_\_ = ["Tadhg Foley"]

#\_\_license\_\_ = "GPL"

#\_\_version\_\_ = "1.0.0"

#\_\_maintainer\_\_ = "Tadhg Foley"

#\_\_email\_\_ = "tadhg.foley@mycit.ie"

#\_\_status\_\_ = "Sexy"

import codecs #import codecs

import string #impot String

codecs.encode('foobar', 'rot\_13')

def make\_rot\_n(n): #create Function

lc = string.lowercase #To Lower Case

trans = string.maketrans(lc, lc[n:] + lc[:n])

return lambda s: string.translate(s, trans)

rot13 = make\_rot\_n(13)

rot13('foobar')

s = open(raw\_input("Enter the file name: "),'r') #Promptes User to enter file name

x = s.read() #Reads input

y = x.encode("rot13") #Stores x in rot13 formate and saves to Y

print "\nThis is the script decoded\n" #Prints to screen

print y #Displays to screen element Y

#Key for encoding

key = {'a':'n', 'b':'o', 'c':'p', 'd':'q', 'e':'r', 'f':'s', 'g':'t', 'h':'u',

'i':'v', 'j':'w', 'k':'x', 'l':'y', 'm':'z', 'n':'a', 'o':'b', 'p':'c',

'q':'d', 'r':'e', 's':'f', 't':'g', 'u':'h', 'v':'i', 'w':'j', 'x':'k',

'y':'l', 'z':'m', 'A':'N', 'B':'O', 'C':'P', 'D':'Q', 'E':'R', 'F':'S',

'G':'T', 'H':'U', 'I':'V', 'J':'W', 'K':'X', 'L':'Y', 'M':'Z', 'N':'A',

'O':'B', 'P':'C', 'Q':'D', 'R':'E', 'S':'F', 'T':'G', 'U':'H', 'V':'I',

'W':'J', 'X':'K', 'Y':'L', 'Z':'M'}

for c in (65, 97): #Algorthim to encode and and decode message using the rot13 method

for i in range(26):

key[chr(i+c)] = chr((i+13) % 26 + c)

print "This is the script encoded\n" #Prints message to screen

print x #Prints X to screen

print "".join([key.get(c, c) for c in s]) #Prints decoded machine.

Lab 5

#! /bin/python

#in home dir

#\_\_author\_\_ = "Tadhg Foley"

#\_\_credits\_\_ = ["Tadhg Foley"]

#\_\_license\_\_ = "GPL"

#\_\_version\_\_ = "1.0.0"

#\_\_maintainer\_\_ = "Tadhg Foley"

#\_\_email\_\_ = "tadhg.foley@mycit.ie"

#\_\_status\_\_ = "Sexy"

import sys, os, os.path ,time, subprocess # imports the sys os,os path time ans sub process

created =0 # String for creation time

asciiTime =0 #Sting for ascii time

err =0 # String for storing the error count

if len(sys.argv) == 1 or len(sys.argv) > 2:

print "Please enter one Argument" #Prompts for argument

exit() #EXITS

elif sys.argv[1].isdigit() == False:

print "Only accepting positive Integer Values" #Integear

exit() #EXITS

elif int(sys.argv[1]) > 120:

print "needs to be within range of 120" #Accepts

exit() #EXITS

else:

t = int(sys.argv[1]) #Gets time of the system avege

rtime = time.time() - (t \* 60) #Get the time t and muilitpys it by 60

startDir = os.getcwd()

directories = ["./"] #Goes though all dictuiores on the system

while len(directories) > 0:

directory = directories.pop()

for name in os.listdir(directory):

fullpath = os.path.join(directory, name)

try:

created = os.stat(name).st\_mtime

except OSError:

err = err+1

asciiTime = time.asctime(time.gmtime(created)) #Get the time in ascii

if os.path.isfile(fullpath) and created >= rtime:

print fullpath, "is a file (created", asciiTime,")" #Displays file name,path and time created

if os.path.isdir(fullpath) and created >= rtime:

directories.append(fullpath)

print fullpath ,"is a dir (created", asciiTime,")" #Display the dictoery lostion snf time created

print "\n There were " + str(err) + " files that were not readable" #Displays the unthinkable and un reachable file

Lab 6

#!/usr/bin/python

#\_\_author\_\_ = "Tadhg Foley"

#\_\_credits\_\_ = ["Tadhg Foley"]

#\_\_license\_\_ = "GPL"

#\_\_version\_\_ = "1.0.0"

#\_\_maintainer\_\_ = "Tadhg Foley"

#\_\_email\_\_ = "tadhg.foley@mycit.ie"

#\_\_status\_\_ = "Sexy"

#Name the first class

class person(object):

def \_\_init\_\_(self,PSN,name,birthday): #this is the constructor

+

self.PSN = PSN #Sets Psn as Psn

self.Name = name #Sets name as name

self.birthday = birthday #Sets birthday as

def \_\_str\_\_(self): # Returns following information(getter)

return "Name: "+self.Name+", PSN: "+self.PSN+", birthday: "+str(self.birthday[0])+"/"+str(self.birthday[1])+"/"+str(self.birthday[2])

class student(person): #Class for student

def \_\_init\_\_(self,PSN,name,birthday,Stnumber,Departmentname): #constructor for Student

super(student, self).\_\_init\_\_(name,PSN,birthday) #Is extending to class person

self.Stnumber=Stnumber #Stores Stnumber as Stnumber

self.Departmentname=Departmentname #Store Departmentname as Department

def \_\_str\_\_(self): #Create a function and returns details (getter)

return "Name: "+self.Name+", PSN: "+self.PSN+", birthday: "+str(self.birthday[0])+"/"+str(self.birthday[1])+"/"+str(self.birthday[2])+", Student number: "+str(self.Stnumber)+", Department: "+self.Departmentname

def department(self): #Function for creating department

return self.Departmentname #Returns local

class employee(person): #Creates the class Employee

def \_\_init\_\_(self,PSN,name,birthday,Workingdepartment,wages): #Creates the constureor

super(employee, self).\_\_init\_\_(name,PSN,birthday) #Super to employee

self.Workingdepartment=Workingdepartment #Stores workingdepart as workingdepart

self.wages=wages #Stores wages as wages

def \_\_str\_\_(self): #New created and returns information

return "Name: "+self.Name+", PSN: "+self.PSN+", birthday: "+str(self.birthday[0])+"/"+str(self.birthday[1])+"/"+str(self.birthday[2])+", wages: "+str(self.wages)+"$, Department: "+self.Workingdepartment

def department(self): #New class for returning a working value

return self.Workingdepartment

#this is the information stored in the class

person1=student('bill o shea','PSN1',(28,12,1992),58550,"Engineering") #person 1

person2=employee('sean o shea','PSN2',(4,8,1982),"Engineering",850) #Person 2

# the information be printed out here

print (person1) #Displays information from person 1

print (person2) #Displays information from person 2

if person1.department()==person2.department()

print ("The two people are from the dept of"+ person1.department()) #displays if students are from same department

else:

print ("The people are from different dept") #displays if students are from same department