Programming and Scripting
Lab Topic 09-Errors

Introduction.

Not a big lab for this topic,
Activity 1 is the quiz,
Activity 2 is to write the code to check an input

quiz:

1.

a. Look at or run the program below, What will happen when the program runs?

```
a = b
```

Why is it that the error mentions b and not a?

b. Look at the program below, what will be outputted to the console when this program is run?

```
try:
    #bad code
    a = b
#catch the NameError and call it ne
except NameError as ne:
    # this prints the message in the error
    # ie "name 'b' is not defined"
    print(ne)
else:
    #this is only run if not error occured
    print("How now")
finally:
    #this is anways run
    print("brown cow")
```

c. Look at the program below, what will the contents of the file be after this program is run.

```
try:
    #this code does not through a name error
    b = 2
    a = b
    # but does throw a TypeError here
    b[10] = 'this is not good'
#catch name errors
except NameError as ne:
    print("a Name error occured")
    print(ne)

#catch all other errors (this is NOT good programming
practice)
# it makes your code very hard to debug
# where did it go wrong? what was the type of error?
except Exception as e:
    print("whoops something went wrong")
    print (e)
```

Lab: Check input.

2. Write a program that takes in a number as an input and subtracts 10% and prints the result,
   the program should throw a value error of the input is less than 0

Answer

```
#
# Read in a number from the keyboard and subtract 10% from
the number
# throw an exception if the input is less then 0
percent = 0.90 # 1 - 0.10
number = float(input("enter number :"))
# The easiest way is to do an assert,
# which is fine, but it does not give much information to
# work out what happen
# assert (number >= 0) # will throw an AssertError
# another way would be to raise an exception
# this way you can put in more information
if (number < 0 ):
    raise ValueError("input should be less than 0
({})".format(number))

newNumber = number * percent
print (" {} minus 10% is {} ".format(number,newNumber))
```

Answers to quiz

a. The program outputs an error (with traceback)

```
Traceback (most recent call last):
  File "lab09.01-quiz-a.py", line 1, in <module>
    a = b
NameError: name 'b' is not defined
```

The program is trying to make a new variable 'a' equal to 'b' but has not been defined

b. In this example we catch the NameError and print its message, so there is not trace back, the code in the else is not executed (because the code in the except was), the code in the finally is executed (because that will always be executed)

```
name 'a' is not defined
brown cow
```

NOTE: this time is is 'a' that is not defined (before the code of 'b' is reached), this is because we are trying to check the value of 'a', how can you check something if you don't know what it is.

c. A NameError is not raised this time, (It is a TypeError that is raised), so the code in the except NameError is not run, and the error is caught by the except Exception (which catches all errrors).
We print out "whoops…" and the message in the error

```
whoops something went wrong
'int' object does not support item assignment
```