CSCI 3430.1 - Principles of Programming Languages

Solution for Assignment 9

Background

My python environment is Python 2.7 in Windows environment

All of the questions in this assignment are developed in separate python source code. t1.py, t2.py, t3.py, t4.py and t5.py

In this solution, I post the screen shot the source code and sample output for each questions. The details of source code can be found in attached.

Question 1:

```
def fOnly( mylist ):mylist = [b for b in mylist if isinstance(b, float)];print mylist;return;
mylist = [1.0, 2, 3.0, "cat", "a", 4, 5];
fOnly(mylist);
```

Sample output:

```
C:\Python27>python.exe t1.py
[1.0, 3.0]
```

Question 2:

```
def notprime(x):
    for i in range(2,x):
        if x%i==0:
            return True
        if i==x-1:
            return False

def fNonPrimes(maxnumber):
    output=filter(notprime,range(2,maxnumber+1))
    print(list(output))

fNonPrimes(10)
```

Sample output:

```
C:\Python27>python.exe t2.py
[4, 6, 8, 9, 10]
```

Question 3:

```
limit = raw_input("Please enter the speed limt:")
speed = raw_input("Please enter your current speed:")
l = 0
s = 0
try:
    f = float(limit)
except ValueError:
    print 'string is not a number for limit speed'
    exit()

try:
    l = int(limit)
except ValueError:
    print 'for limit speed cannot be float'
    exit()

try:
    f = float(speed)
except ValueError:
    print 'string is not a number for current speed'
    exit()

try:
    s = int(speed)

except ValueError:
    print 'current speed cannot be float'
    exit()

if l < 0:
    print 'speed limit is less than zero'
    exit()
if s < 0:
    print 'current speed is less than zero'
    exit()

if s <= l:
    print '$0 because your current speed equal or below the speed limit'
    exit()
else:
    if s < l+5:
        print '$200 becuase your current speed is less than 5 km per hour above the speed limit'
        exit()
    else:
        if s < l+20 :
            f = ( s - l -5 ) * 500
            print 'speeding fine is $' + str(f)
            exit()
        else:
            if s < 400:
                f = ( s - l - 20 ) * 1000
                print 'speeding fine is $' + str(f)
                print 'Because $1000 for every additional km per hour above over 20 km per hour above the speed limit'
            else:
                print 'speeding fine is 0 becuase yuou are in a bufatti so they can not catcy you!'
```

Sample output:

```
C:\Python27>python.exe t3.py
Please enter the speed limt:10
Please enter your current speed:9
$0 because your current speed equal or below the speed limit
```

```
C:\Python27>python.exe t3.py
Please enter the speed limt:10
Please enter your current speed:20
speeding fine is $2500
```

```
C:\Python27>python.exe t3.py
Please enter the speed limt:10
Please enter your current speed:500
speeding fine is 0 becuase yuou are in a bufatti so they can not catcy you!
```

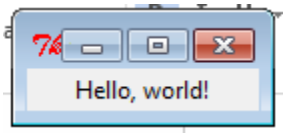
Question 4:

```
import Tkinter
root = Tkinter.Tk()
w = Tkinter.Label(root, text="Hello, world!")
w.pack()
root.mainloop()
```

In this question, I use Tkinter

Sample output:

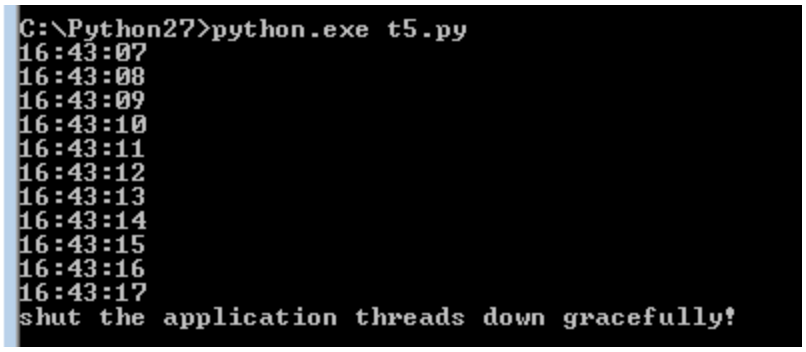
```
C:\Python27>python.exe t4.py
```



Question 5:

```
import thread
import time
import datetime
from time import gmtime, strftime
import threading

#use the lock function to mutex the variable
lock = threading.Lock()
global index
index=0

#writes the time of day as an HH:MM:SS string into a global variable 100 times per second
def thread1():
    while True:
        time.sleep(0.1)
        global currentTime
        lock.acquire()
        try:
            currentTime = strftime("%H:%M:%S", gmtime())
        finally:
            lock.release()

def thread2():
    while True:
        #read the time of day string from global variable twice per second
        time.sleep(0.5)
        global index
        index = index +1
        #really displays to screen only once per second
        if index % 2 == 0:
            continue
        lock.acquire()
        try:
            print currentTime
        finally:
            lock.release()


t1 = threading.Thread(target=thread1)
t1.setDaemon(True)
t2 = threading.Thread(target=thread2)
t2.setDaemon(True)
t1.start()
t2.start()

#killing both threads before the main program thread exits
try:
    while 1:
        time.sleep(0.01)
except KeyboardInterrupt:
    t1.kill_received = True
    t2.kill_received = True
    print "shut the application threads down gracefully!"
```

Sample output:


C:\Python27>python.exe t5.py
16:43:07
16:43:08
16:43:09
16:43:10
16:43:11
16:43:12
16:43:13
16:43:14
16:43:15
16:43:16
16:43:17
shut the application threads down gracefully!