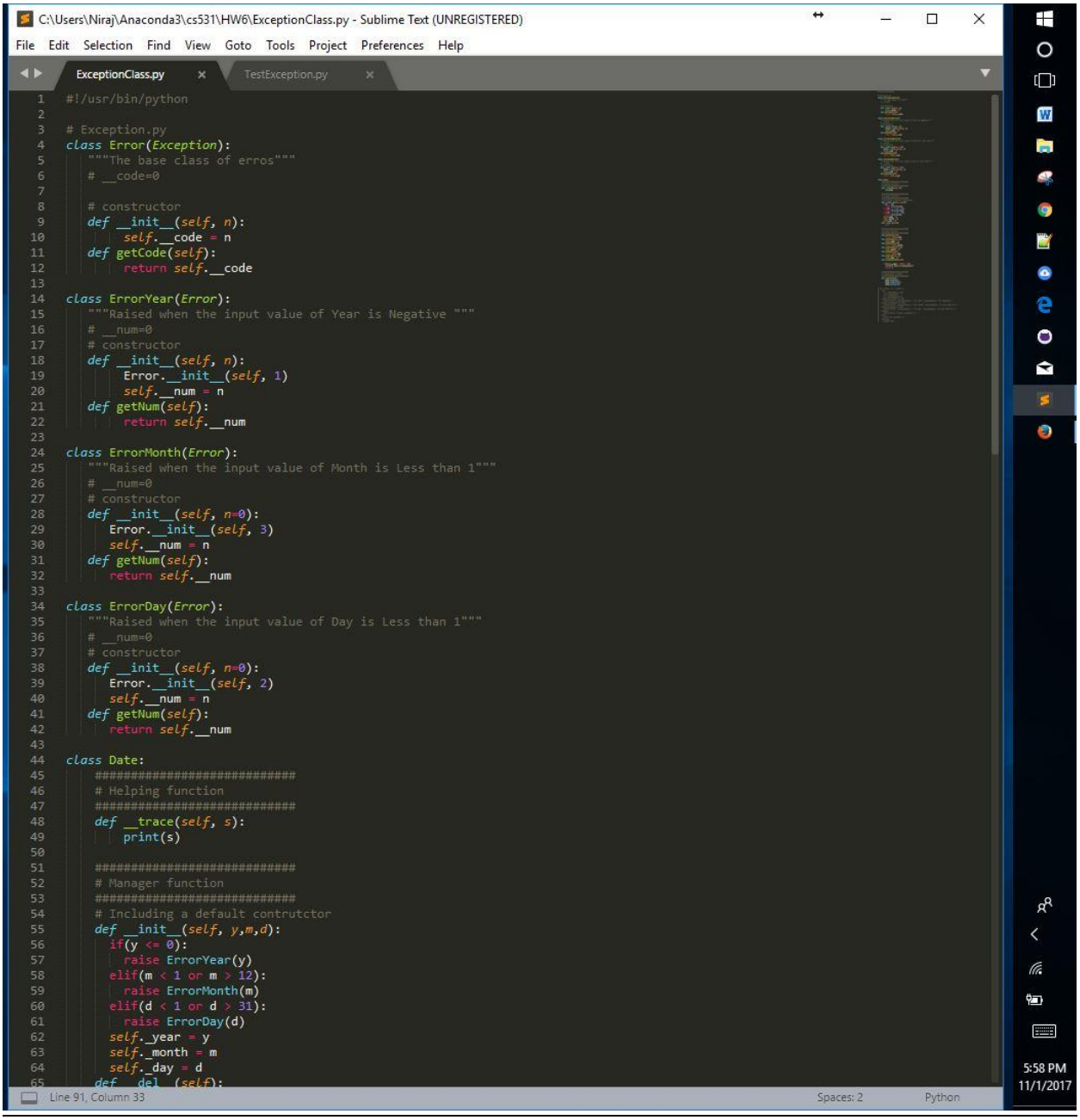
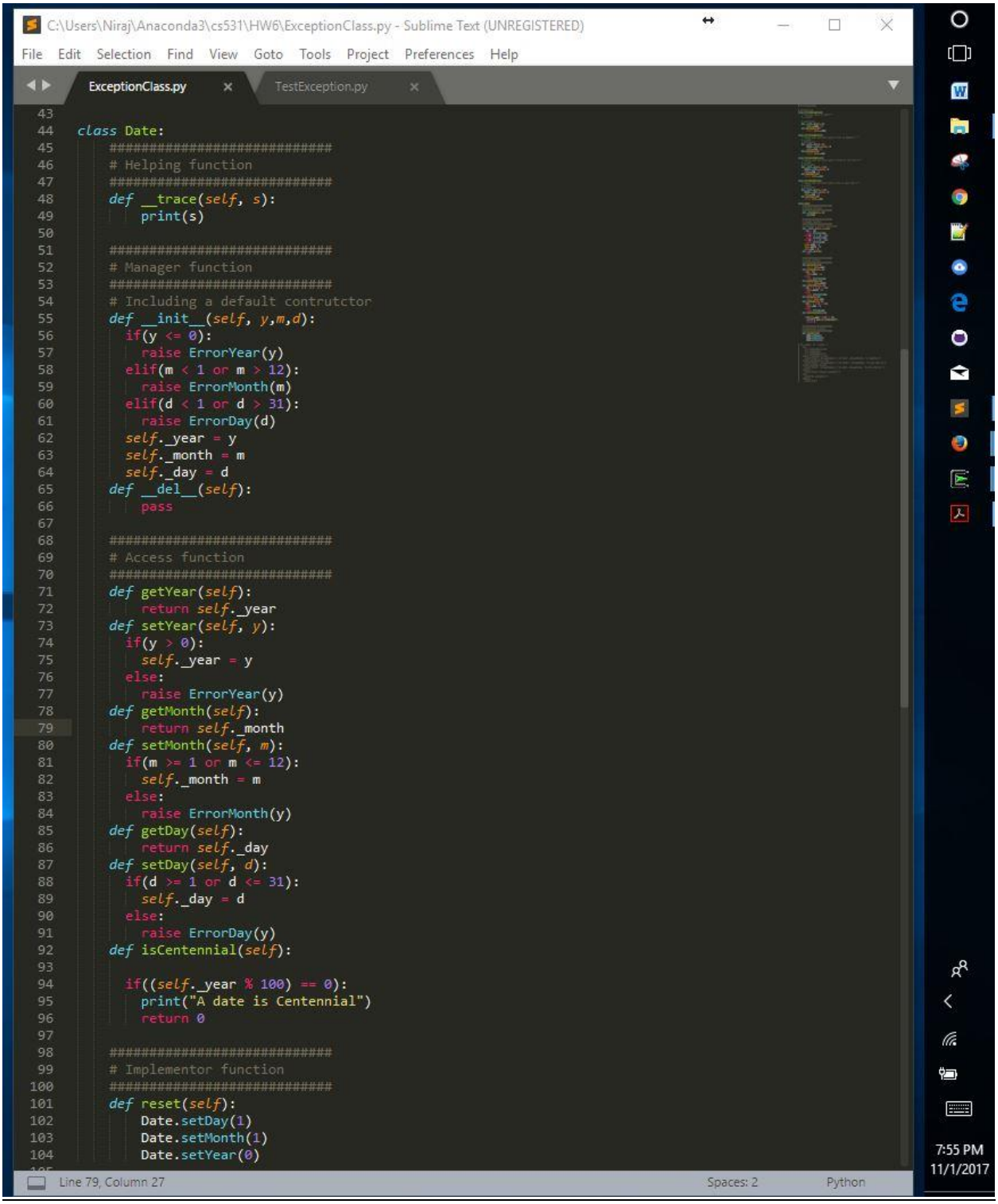
Name : Niraj Thanki SID : 19376 CLASS : CS531

Exception.py




C:\Users\Niraj\Anaconda3\cs531\HW6\ExceptionClass.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
ExceptionClass.py
TestException.py
class Date:
############################
Helping function
############################
def __trace(self, s):
print(s)
############################
Manager function
############################
# Including a default contrutctor
def __init__(self, y,m,d):
if(y <= 0):
raise ErrorYear(y)
elif(m < 1 or m > 12):
raise ErrorMonth(m)
elif(d < 1 or d > 31):
raise ErrorDay(d)
self._year = y
self._month = m
self._day = d
def __del__(self):
pass
############################
Access function
############################
def getYear(self):
return self._year
def setYear(self, y):
if(y > 0):
self._year = y
else:
raise ErrorYear(y)
def getMonth(self):
return self._month
def setMonth(self, m):
if(m >= 1 or m <= 12):
self._month = m
else:
raise ErrorMonth(y)
def getDay(self):
return self._day
def setDay(self, d):
if(d >= 1 or d <= 31):
self._day = d
else:
raise ErrorDay(y)
def isCentennial(self):
if((self._year % 100) == 0):
print("A date is Centennial")
return 0
############################
Implementor function
############################
def reset(self):
Date.setDay(1)
Date.setMonth(1)
Date.setYear(0)
Line 79, Column 27
Spaces: 2
Python
7:55 PM
11/1/2017

Source Code:

```
#!/usr/bin/python


# Exception.py
class Error(Exception):
  """The base class of erros"""
  # __code=0


  # constructor
  def __init__(self, n):
      self.__code = n
  def getCode(self):
      return self.__code


class ErrorYear(Error):
  """Raised when the input value of Year is Negative """
  # __num=0
  # constructor
  def __init__(self, n):
      Error.__init__(self, 1)
      self.__num = n
  def getNum(self):
```

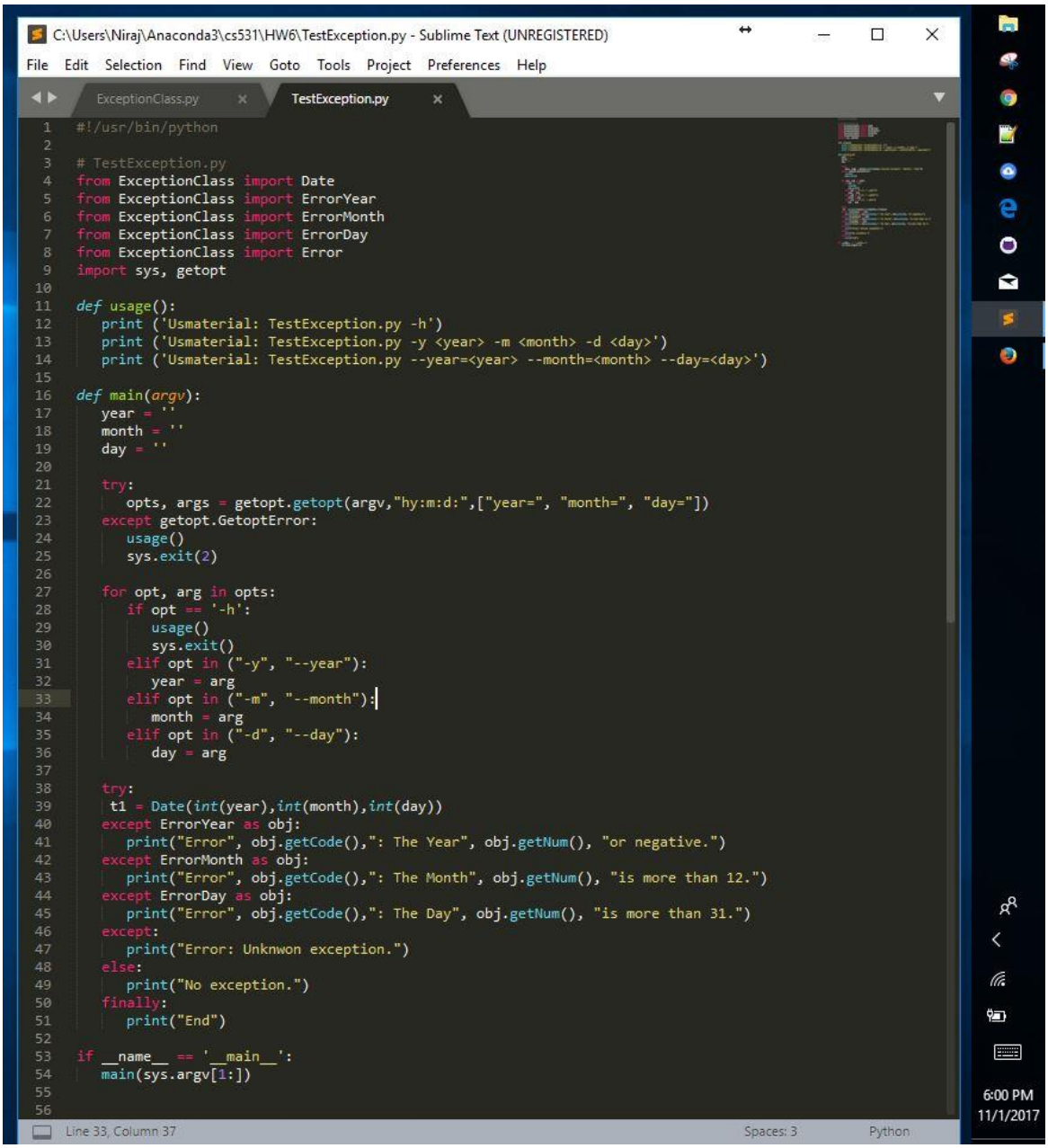
```
        return self.__num


class ErrorMonth(Error):
   """Raised when the input value of Month is Less than 1"""
   # __num=0
   # constructor
   def __init__(self, n=0):
      Error.__init__(self, 3)
      self.__num = n
   def getNum(self):
      return self.__num


class ErrorDay(Error):
   """Raised when the input value of Day is Less than 1"""
   # __num=0
   # constructor
   def __init__(self, n=0):
      Error.__init__(self, 2)
      self.__num = n
   def getNum(self):
      return self.__num


class Date:
    ############################
    # Helping function
```

```
############################
def __trace(self, s):
   print(s)

############################
# Manager function
############################
# Including a default contrutctor
def __init__(self, y,m,d):
  if(y <= 0):
    raise ErrorYear(y)
  elif(m < 1 or m > 12):
    raise ErrorMonth(m)
  elif(d < 1 or d > 31):
    raise ErrorDay(d)
  self._year = y
  self._month = m
  self._day = d
def __del__(self):
   pass

############################
# Access function
############################
def getYear(self):
```

```
    return self._year
def setYear(self, y):
  if(y > 0):
    self._year = y
  else:
    raise ErrorYear(y)
def getMonth(self):
    return self._month
def setMonth(self, m):
  if(m >= 1 or m <= 12):
    self._month = m
  else:
    raise ErrorMonth(y)
def getDay(self):
    return self._day
def setDay(self, d):
  if(d >= 1 or d <= 31):
    self._day = d
  else:
    raise ErrorDay(y)
def isCentennial(self):


  if((self._year % 100) == 0):
    print("A date is Centennial")
    return 0
```

```
############################
# Implementor function
############################
def reset(self):
    Date.setDay(1)
    Date.setMonth(1)
    Date.setYear(0)
```

TestException.py



Source Code:

#!/usr/bin/python

TestException.py

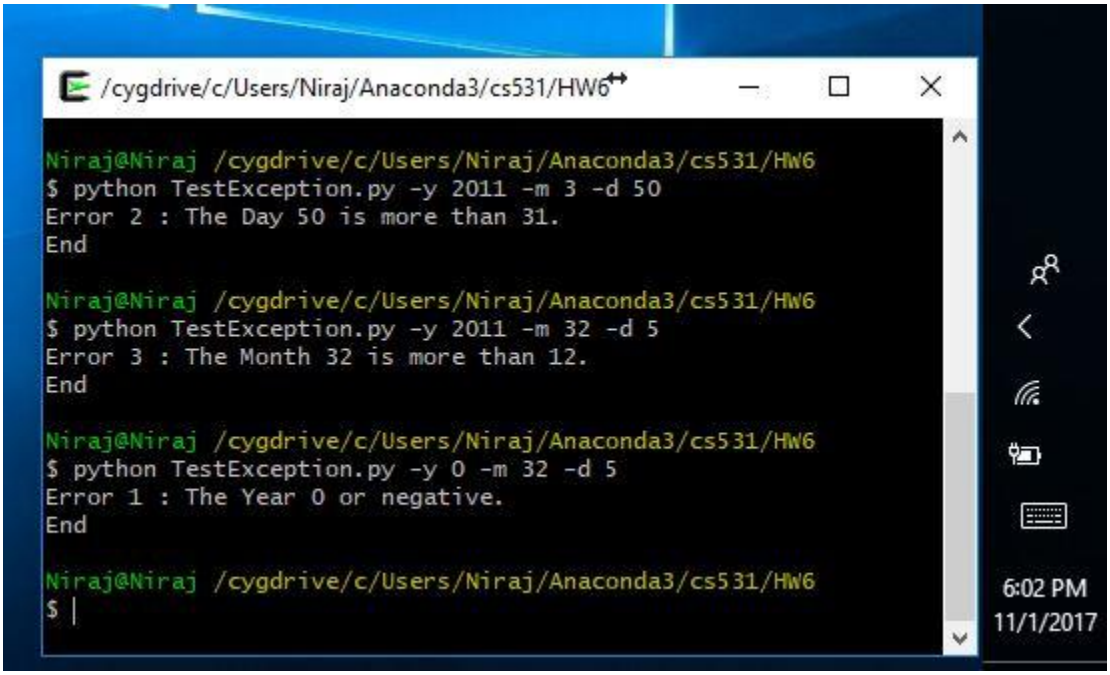
from ExceptionClass import Date

from ExceptionClass import ErrorYear

```
from ExceptionClass import ErrorMonth
from ExceptionClass import ErrorDay
from ExceptionClass import Error
import sys, getopt

def usage():
  print ('Usmaterial: TestException.py -h')
  print ('Usmaterial: TestException.py -y <year> -m <month> -d <day>')
  print ('Usmaterial: TestException.py --year=<year> --month=<month> --day=<day>')

def main(argv):
  year = ''
  month = ''
  day = ''

  try:
    opts, args = getopt.getopt(argv,"hy:m:d:",["year=", "month=", "day="])
  except getopt.GetoptError:
    usage()
    sys.exit(2)

  for opt, arg in opts:
    if opt == '-h':
      usage()
      sys.exit()
```

```
    elif opt in ("-y", "--year"):
      year = arg
    elif opt in ("-m", "--month"):
      month = arg
    elif opt in ("-d", "--day"):
      day = arg

  try:
   t1 = Date(int(year),int(month),int(day))
  except ErrorYear as obj:
    print("Error", obj.getCode(),": The Year", obj.getNum(), "or negative.")
  except ErrorMonth as obj:
    print("Error", obj.getCode(),": The Month", obj.getNum(), "is more than 12.")
  except ErrorDay as obj:
    print("Error", obj.getCode(),": The Day", obj.getNum(), "is more than 31.")
  except:
    print("Error: Unknwon exception.")
  else:
    print("No exception.")
  finally:
    print("End")

if __name__ == '__main__':
  main(sys.argv[1:])
```

Output :


/cygdrive/c/Users/Niraj/Anaconda3/cs531/HW6
Niraj@Niraj /cygdrive/c/Users/Niraj/Anaconda3/cs531/HW6
$ python TestException.py -y 2011 -m 3 -d 50
Error 2 : The Day 50 is more than 31.
End
Niraj@Niraj /cygdrive/c/Users/Niraj/Anaconda3/cs531/HW6
$ python TestException.py -y 2011 -m 32 -d 5
Error 3 : The Month 32 is more than 12.
End
Niraj@Niraj /cygdrive/c/Users/Niraj/Anaconda3/cs531/HW6
$ python TestException.py -y 0 -m 32 -d 5
Error 1 : The Year 0 or negative.
End
Niraj@Niraj /cygdrive/c/Users/Niraj/Anaconda3/cs531/HW6
$
6:02 PM
11/1/2017