

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 x TestException.py x

1 #!/usr/bin/python
2
3 # Exception.py
4 class Error(Exception):
5     """The base class of errors"""
6     # __code=0
7
8     # constructor
9     def __init__(self, n):
10         self.__code = n
11     def getCode(self):
12         return self.__code
13
14 class ErrorYear(Error):
15     """Raised when the input value of Year is Negative """
16     # __num=0
17     # constructor
18     def __init__(self, n):
19         Error.__init__(self, 1)
20         self.__num = n
21     def getNum(self):
22         return self.__num
23
24 class ErrorMonth(Error):
25     """Raised when the input value of Month is Less than 1"""
26     # __num=0
27     # constructor
28     def __init__(self, n=0):
29         Error.__init__(self, 3)
30         self.__num = n
31     def getNum(self):
32         return self.__num
33
34 class ErrorDay(Error):
35     """Raised when the input value of Day is Less than 1"""
36     # __num=0
37     # constructor
38     def __init__(self, n=0):
39         Error.__init__(self, 2)
40         self.__num = n
41     def getNum(self):
42         return self.__num
43
44 class Date:
45     #####
46     # Helping function
47     #####
48     def __trace(self, s):
49         print(s)
50
51     #####
52     # Manager function
53     #####
54     # Including a default contructor
55     def __init__(self, y,m,d):
56         if(y <= 0):
57             raise ErrorYear(y)
58         elif(m < 1 or m > 12):
59             raise ErrorMonth(m)
60         elif(d < 1 or d > 31):
61             raise ErrorDay(d)
62         self.__year = y
63         self.__month = m
64         self.__day = d
65     def __del__(self):
```

Line 91, Column 33 Spaces: 2 Python 5:58 PM 11/1/2017

```
C:\Users\Niraj\Anaconda3\cs531\HW6\ExceptionClass.py - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help

ExceptionClass.py x TestException.py x

43
44 class Date:
45     #####
46     # Helping function
47     #####
48     def __trace(self, s):
49         print(s)
50
51     #####
52     # Manager function
53     #####
54     # Including a default contrutctor
55     def __init__(self, y,m,d):
56         if(y <= 0):
57             raise ErrorYear(y)
58         elif(m < 1 or m > 12):
59             raise ErrorMonth(m)
60         elif(d < 1 or d > 31):
61             raise ErrorDay(d)
62         self._year = y
63         self._month = m
64         self._day = d
65     def __del__(self):
66         pass
67
68     #####
69     # Access function
70     #####
71     def getYear(self):
72         return self._year
73     def setYear(self, y):
74         if(y > 0):
75             self._year = y
76         else:
77             raise ErrorYear(y)
78     def getMonth(self):
79         return self._month
80     def setMonth(self, m):
81         if(m >= 1 or m <= 12):
82             self._month = m
83         else:
84             raise ErrorMonth(y)
85     def getDay(self):
86         return self._day
87     def setDay(self, d):
88         if(d >= 1 or d <= 31):
89             self._day = d
90         else:
91             raise ErrorDay(y)
92     def isCentennial(self):
93
94         if((self._year % 100) == 0):
95             print("A date is Centennial")
96             return 0
97
98     #####
99     # Implementor function
100     #####
101     def reset(self):
102         Date.setDay(1)
103         Date.setMonth(1)
104         Date.setYear(0)
105
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
```

```
#####
```

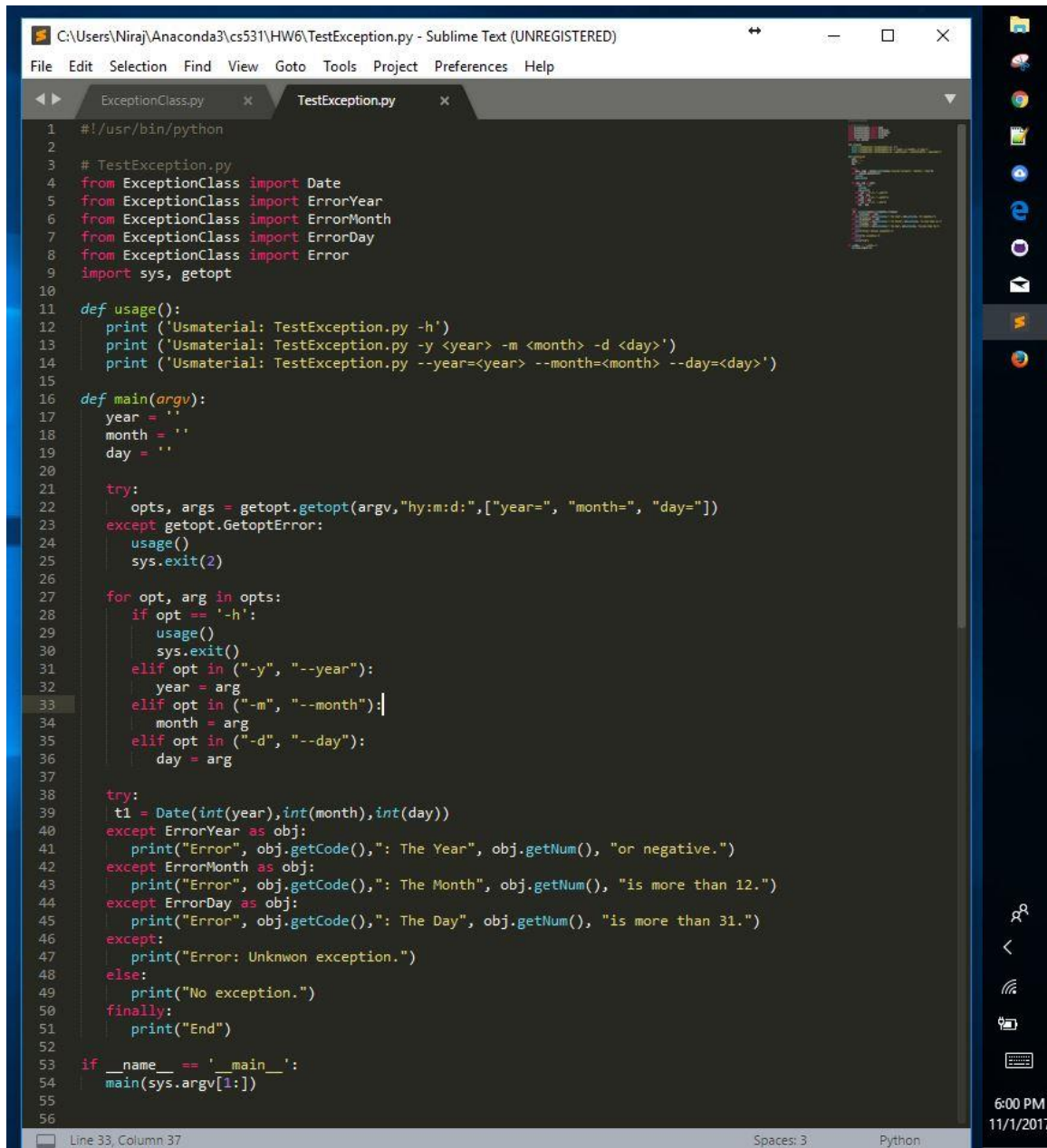
```
def reset(self):
```

```
    Date.setDay(1)
```

```
    Date.setMonth(1)
```

```
    Date.setYear(0)
```

TestException.py



```
1 #!/usr/bin/python
2
3 # TestException.py
4 from ExceptionClass import Date
5 from ExceptionClass import ErrorYear
6 from ExceptionClass import ErrorMonth
7 from ExceptionClass import ErrorDay
8 from ExceptionClass import Error
9 import sys, getopt
10
11 def usage():
12     print('Usmaterial: TestException.py -h')
13     print('Usmaterial: TestException.py -y <year> -m <month> -d <day>')
14     print('Usmaterial: TestException.py --year=<year> --month=<month> --day=<day>')
15
16 def main(argv):
17     year = ''
18     month = ''
19     day = ''
20
21     try:
22         opts, args = getopt.getopt(argv,"hy:m:d:",["year=", "month=", "day="])
23     except getopt.GetoptError:
24         usage()
25         sys.exit(2)
26
27     for opt, arg in opts:
28         if opt == '-h':
29             usage()
30             sys.exit()
31         elif opt in ("-y", "--year"):
32             year = arg
33         elif opt in ("-m", "--month"):
34             month = arg
35         elif opt in ("-d", "--day"):
36             day = arg
37
38     try:
39         t1 = Date(int(year),int(month),int(day))
40     except ErrorYear as obj:
41         print("Error", obj.getCode(),": The Year", obj.getNum(), "or negative.")
42     except ErrorMonth as obj:
43         print("Error", obj.getCode(),": The Month", obj.getNum(), "is more than 12.")
44     except ErrorDay as obj:
45         print("Error", obj.getCode(),": The Day", obj.getNum(), "is more than 31.")
46     except:
47         print("Error: Unknown exception.")
48     else:
49         print("No exception.")
50     finally:
51         print("End")
52
53 if __name__ == '__main__':
54     main(sys.argv[1:])
55
56
```

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.")
```

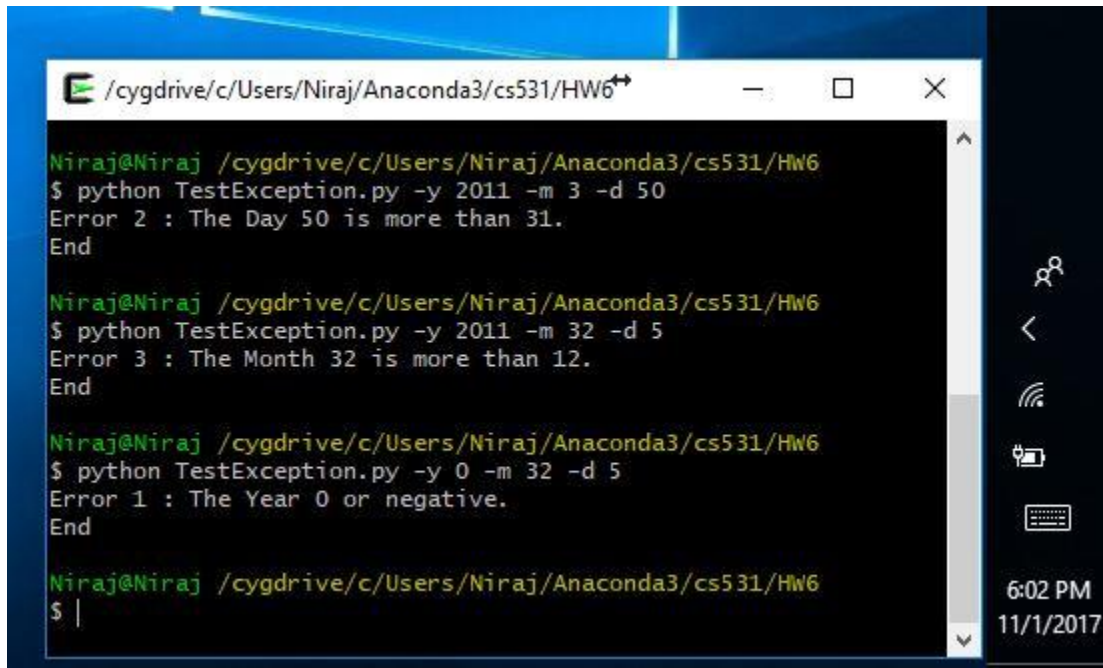
```
finally:
```

```
    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