Assignment 6

File handling and exception handling

LILLEBAELT ACADEMY OF
PROFESSIONAL HIGHER EDUCATION

Author
Martin Grønholdt
mart80c7@edu.eal.dk

Sunday 18 December 2016

Table of Contents

Introduction

The programs in this hand-in uses most of the things learned so far. Some programs have not been split into functions, since they were so simple that it would be an exercise in over complication.

All files for this hand in are available at:
https://github.com/deadbok/eal_programming/tree/master/Assignment 4B

Error handling

All programs, requesting user input, handle bad input by asking the user, to use only the correct data type, where after it exits.

```
Enter the amount of a purchase: 2hjjhg

Please use only numbers.
```

Example output of a program when the user enters an incorrect value

All programs using file I/O will show an error message if something goes wring during file access.

6. Average of Numbers

This program calculate the average of a list of integers read from the file “numbers.txt”.

prog6.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
"""
Name: prog6.py
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (2016-12-18)

A program that calculates the avarage of a list of comma seperated integers.
"""


def main():
    '''
    Program main entry point.
    '''
    try:
        # Open 'numbers.txt' for reading.
        with open('numbers.txt', 'r') as numbers_file:
            # Get the number form the file.
            numbers = numbers_file.readline()
            # Show them.
            print('Numbers in numbers.txt: ' + numbers)
            # Convert them to a list of integers using list comprehensions.
            numbers = [int(x) for x in numbers.split(',')]
            # Print the total sum.
            print('The average of the numbers:' +
                  ' {:.2f}'.format(sum(numbers) / len(numbers)))
    except IOError as ex:
        # Complain when something goes wrong with the file access.
        print('Exception: {}'.format(str(ex)))
        print('Error reading "numbers.txt".')
        exit(1)
    except ValueError:
        print('File should contain only comma separated numbers.')
        exit(1)


# Run this when invoked directly
if __name__ == '__main__':
    main()
```

Result

All results are using the file “numbers.txt” file generated and show in the results of assignment 7.

```
Numbers in numbers.txt:
12,41,61,20,51,43,33,85,91,46,12,92,13,11,46,94,46,53,93,13

The average of the numbers: 47.80
```

Output of the program when run from the command line.

```
Exception: [Errno 2] No such file or directory: 'numbers.txt'
Error reading "numbers.txt.bak".
```

Output of the program when the file is missing.

```
Numbers in numbers.txt:
12,41,61,20,51,43,33,85,91,46,12,92,13,11,46,94,46,53,93,13,error

File should contain only comma separated numbers.
```

Output of the program when the file has malformed contents.

7. Random Number File Writer

This program calculate writes a number of comma separated integers to a file.

prog7.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
"""
Name: prog7.py
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (2016-12-18)

A program that writes random integers to a file.
"""
from random import randrange
import os


def main():
    '''
    Program main entry point.
    '''
    # Get a file name and the number of integers to write to it.
    try:
        filename = input('Enter a file name to write the random integers to: ')
        count = int(input('Enter number of random integers to generate: '))
        # Complain when something unexpected was entered.
    except ValueError:
        print('Please use only numbers.')
        exit(1)
    except Exception as ex:
        print('Exception: {}'.format(str(ex)))
        print('\nSomething went wrong, try another name.')
        exit(1)

    # Check and warn if the file exists.
    if os.path.exists(filename):
        print('WARNING: Path exists are you sure you want to continue ' +
              'overwriting (y to continue)? ', end='')
        if input() != 'y':
            exit(0)

    try:
        # Open the file.
        with open(filename, 'w') as number_file:
            # Write the correct number of entries.
            for i in range(0, count):
                number = randrange(0, 100)
                if i > 0:
                    # Generate a Comma Separated File.
                    number_file.write(',')

                number_file.write(str(number))
                print('{}, '.format(number), end='')
        print('done.')
    except IOError:
```

```
        print('Error writing file.')
        exit(1)


# Run this when invoked directly
if __name__ == '__main__':
    main()
```

Result

```
Enter a file name to write the random integers to: numbers.txt
Enter number of random integers to generate: 20
12, 41, 61, 20, 51, 43, 33, 85, 91, 46, 12, 92, 13, 11, 46, 94, 46, 53, 93, 13,
done.
```

Console output of the program.

```
Enter a file name to write the random integers to: numbers.txt
Enter number of random integers to generate: 20
WARNING: Path exists are you sure you want to continue overwriting (y to
continue)? y
12, 41, 61, 20, 51, 43, 33, 85, 91, 46, 12, 92, 13, 11, 46, 94, 46, 53, 93, 13,
done.
```

Console output of the program when the target file exists.

```
12,41,61,20,51,43,33,85,91,46,12,92,13,11,46,94,46,53,93,13
```

numbers.txt output by the program.

8. Random Number File Reader

This program read the numbers back, when written in a format like prog7.py.

prog8.py

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
# The above lines tell the shell to use python as interpreter when the
# script is called directly, and that this file uses utf-8 encoding,
# because of the country specific letter in my surname.
"""
Name: prog8.py
Author: Martin Bo Kristensen Grønholdt.
Version: 1.0 (2016-12-18)

A program that prints the numbers from a file a with list of comma separated
integers.
"""


def main():
    '''
    Program main entry point.
    '''
    # Get a file name.
    try:
        filename = input('Enter a file name to read the random integers from: ')
        # Complain when something unexpected was entered.
    except Exception as ex:
        print('Exception: {}'.format(str(ex)))
        print('\nSomething went wrong, try another name.')
        exit(1)

    try:
        # Open 'numbers.txt' for reading.
        with open(filename, 'r') as numbers_file:
            # Get the number form the file.
            numbers = numbers_file.readline()
            # Show them.
            print('Numbers in numbers.txt: ' + numbers)
            # Convert them to a list of integers using list comprehensions.
            numbers = [int(x) for x in numbers.split(',')]
            # Print number of items.
            print('Numbers: {}'.format(len(numbers)))
            # Print the total sum.
            print('The sum of the numbers: {:.2f}'.format(sum(numbers)))
    except IOError as ex:
        # Complain when something goes wrong with the file access.
        print('Exception: {}'.format(str(ex)))
        print('Error reading ' + filename + '.')
        exit(1)
    except ValueError:
        print('File should contain only comma separated numbers.')
        exit(1)
```

```
# Run this when invoked directly
if __name__ == '__main__':
    main()
```

Result

```
Enter a file name to read the random integers from: numbers.txt
Numbers in numbers.txt:
12,41,61,20,51,43,33,85,91,46,12,92,13,11,46,94,46,53,93,13

Numbers: 20
The sum of the numbers: 956.00
```

Output of the program.

```
Enter a file name to read the random integers from: numers.txt
Exception: [Errno 2] No such file or directory: 'numers.txt'
Error reading numers.txt.
```

Output when a wrong file name is entered.

9. Drawing pattern

Ooops I did that in assignment 6.

Conclusion

The reading and writing of files have been accomplished as well as proper exception handling.
All source files for this assignment are available at
https://github.com/deadbok/eal_programming/tree/master/Assignment%206.