Fundamentals of Computer Programming Building a Programming Portfolio

Week 5

You should be able to complete the following programs by the end of the week. By now you should understand why you should be saving your work to GitHub or similar. Possible solutions will be uploaded to the main module GitHub repository every week. If you follow that repo you should be able to receive notifications.

After this week you should be able to run your programs from the command-line, without having to use an IDE.

Note: Most of these programs process command-line arguments. In every case your program should not crash if no arguments are provided. In most cases it should just exit with some suitable error message.

1. Using command-line arguments involves the `sys` module. Review the docs for this module and using the information in there write a short program that when run from the command-line reports what operating system platform is being used.

   Code-

```
Week 5 > exercise 1.py
1   import sys
2   print("You are now using", sys.platform)
```

   Output-

```
C:\Users\diksh\OneDrive\Desktop\FOCP\
You are now using win32
PS C:\Users\diksh\OneDrive\Desktop\FOCP>
```

2. Write a program that, when run from the command line, reports how many arguments were provided. (Remember that the program name itself is *not* an argument).

   Code-

```
Week 5 > exercise 2.py
1   from sys import argv
2   print(len(argv)-1)
```

   Output-

```
0
PS C:\Users\diksh\OneDrive\Desktop\FOCP>
```

3. Write a program that takes a bunch of command-line arguments, and then prints

out the shortest. If there is more than one of the shortest length, any will do.
Hint: Don't overthink this. A good way to find the shortest is just to sort them.
Code-

Week 5 > exercise 3.py > ...

```python
from sys import argv
list = argv[:]
list.sort(key=len)
print(list)
try:
    print(f"The shortest word is: {list[0]}")
except:
    print("word is not passed")
```

Output-

```
PS C:\Users\diksh\OneDrive\Desktop\FOCP>
python "C:\Users\diksh\OneDrive\Desktop\FOCP\Week 5\exercise 3.py" Dikshya Diki Diks D
['D', 'Diki', 'Diks', 'Dikshya', 'C:\\Users\\diksh\\OneDrive\\Desktop\\FOCP\\Week 5\\exercise 3.py']
The shortest word is: D
PS C:\Users\diksh\OneDrive\Desktop\FOCP>
```

4. Write a program that takes a URL as a command-line argument and reports whether or not there is a working website at that address.
Hint: You need to get the HTTP response code.
Another Hint: StackOverflow is your friend.
Code-

Week 5 > qustion 4.py > ...

```python
from sys import argv
from urllib.request import urlopen
from urllib.error import *
try :
    user_link = argv [1]
    link = urlopen(user_link)

except IndexError as i:
    print("No url provided and", i)

except HTTPError as h:
    print("HTTP erroe", h)

except URLError as u:
    print("Opps ! page not found !", u)

else:
    print("Yeah ! page found")
```

Output-

```
PS C:\Users\diksh\OneDrive\Desktop\FOCP>
python "C:\Users\dikshpython "C:\Users\diksh\OneDrive\Desktop\FOCP\Week 5\qustion 4.py" https://www.google.com
Yeah ! page found
PS C:\Users\diksh\OneDrive\Desktop\FOCP>
```

5. Last week you wrote a program that processed a collection of temperature readings entered by the user and displayed the maximum, minimum, and mean. Create a

version of that program that takes the values from the command-line instead. Be sure to handle the case where no arguments are provided!

Code-

Week 5 > question 5.py > ...

```python
from sys import argv
from statistics import mean

try:
    list = argv[1:]
    new_list = [(int(i)) for i in list]

    max_value = max(new_list)
    min_value = min(new_list)
    mean_value = mean(new_list)


except IndexError as i:
    print("No string provided and", i)

except ValueError as v:
    print("No Argument Provided", v)

except NameError as n:
    print("No Argument Provided", n)

else:
    print("The max Value is", max_value)
    print("The min Value is", min_value)
    print("The mean Value is", mean_value)
```

Output-

```
PS C:\Users\diksh\OneDrive\Desktop\FOCP> python "C:\Users\diksh\OneDrive\Desktop\FOCP\Week 5\question 5.py"5 7 8 23 10
The max Value is 23
The min Value is 5
The mean Value is 10.6
PS C:\Users\diksh\OneDrive\Desktop\FOCP>
```

6. Write a program that takes the name of a file as a command-line argument, and creates a backup copy of that file. The backup should contain an exact copy of the contents of the original and should, obviously, have a different name.

Hint: By now, you should be getting the idea that there is a built-in way to do the heavy lifting here! Take a look at the "Brief Tour of the Standard Library" in the docs.

Code-

Week 5 > question 6.py > ...

```python
from shutil import copyfile
from sys import argv
origin_filename=None
try :
    origin_filename = argv[1]
    filename, extension = origin_filename.split('.')

    target_filename = (f"{filename}_backup.{extension}")

    copyfile(origin_filename, target_filename)
    print ("File copied")

except IndexError as i:
    print('file name is not given', i)
except ValueError as v:
    print("cannot find file extension.", v)
except FileNotFoundError as f:
    filename=origin_filename
    print(f"Cannot open {filename}.", f)
```

Output-

```
PS C:\Users\diksh\OneDrive\Desktop\FOCP\Week 5>  python "question 6.py" document.txt
File copied
PS C:\Users\diksh\OneDrive\Desktop\FOCP\Week 5>
```