University of Technology Sydney
Faculty of Engineering and Information Technology

Subject: 32547 UNIX Systems Programming, Autumn 2020

Assignment

Description

This assignment is an individual programming assignment using Python. It addresses objectives 2 and 3 as listed in the Subject Outline document.

No limits apply to the number of lines of code of the program.

Assignments are **to be completed individually** (this might be checked with the use of anti-plagiarism tools such as Turnitin). **You should not receive help in the preparation of this assignment, nor ask anyone else to prepare it on your behalf in any way**.

Marks

The assignment corresponds to 30% of the total marks.

Submission

The completed assignment is due by **5:00pm of Friday 12 June 2020**.

PLEASE PAY ATTENTION TO THE FOLLOWING SUBMISSION INSTRUCTIONS:

1. Your Python program has to be submitted on Canvas, following the "Assignments" menu and then the "Assignment" link by the due date. You can prepare your Python program anywhere you like, but probably the easiest option is to develop it in a workspace that you can create in Ed STEM.
2. Please submit only your Python program, and nothing else (no data files).
3. Late submissions will be deducted one mark per day late; more than seven days late, the assignment will receive zero. Special considerations for a late submission must be arranged in advance with the Subject Coordinator.

Academic Standards

The assignments in this Subject should be your own original work. Any code taken from a textbook, journal, the Internet or any other source should be acknowledged. For referencing, please use the standard referencing conventions (http://www.lib.uts.edu.au/help/referencing).

Marking Scheme

Mark Range	
30	All requirements of the assignment are met. The submitted program can be executed by the lecturer "as is" and produces the requested output.
24-29	The program works correctly with any valid file and for all options, but its execution experiences some minor problems.
18-23	The program does not work as expected with one of the options.
0-17	This range goes from no submission at all (0 marks) to a submission which does not meet major requirements of the assignment. Examples: • the program does not work as expected with two or more options; • the program generates unhandled errors; • the program does not solve the assignment problem.

The assignments will be marked within two weeks from the submission deadline or as soon as possible.

Important notes:

- **Submission of this assignment is not compulsory to pass the subject; do not feel that you have to submit it "at all costs" and perhaps be tempted to seek unauthorised assistance**.
- There are no minimum requirements on the marks on this assignment to pass the subject.

Title: Printing usage report with Python

In this assignment, you will write a Python program which parses a file containing information about the printing usage of a number of users and outputs summary information.

These are the specifications for your Python program:

1. It must be named *printing_summary.py*

2. It should be invoked as:

   `python printing_summary.py option printing_usage_file`

   The program must check that argument *printing_usage_file* exists, is a file and is readable. If not, it must print an error message to **the standard output** and exit. The specifications for the *printing_usage_file* and *option* arguments are given below.

3. File *printing_usage_file* can have any arbitrary name. It must be a file of text with the following format:

   a. The file consists of an arbitrary number of lines (including, possibly, zero lines).

   b. Each line must contain three fields separated by commas.

   c. The three fields are: *filename*, *size in bytes*, *username*.

   d. The *filename* field is a string of characters of arbitrary (yet reasonably limited) length. Acceptable characters include: lower and upper case letters, digits, underscore, dot, space.

   e. The *size in bytes* field is an integer limited between 1 and 67,108,864.

   f. The *username* field is a string of characters of arbitrary (yet reasonably limited) length. Acceptable characters include: lower and upper case letters, digits, underscore, dot.

   The following example should be regarded as the reference specification for the format of file *printing_usage_file*:

```
article.pdf,1550008,massimo.piccardi
USP 32547 Assignment S2012.docx,36024,MarkThomas
article.ps,10000000,massimo.piccardi
article.pdf,205000,massimo.piccardi
Recipes.html,239250,schmidt1974
```

Important note: your program **is not expected to verify** that file *printing_usage_file* complies with the above specifications. It will only be tested with compliant files.

4. Your program can be invoked with option: -a. In this case, it must print each unique username in the order in which it first appears in the file:

```
Printing users:
<first user in appearance order>
<second user in appearance order>
…
<last user in appearance order>
```

Example with the example *printing_usage_file* given above:

Command line:

```
printing_summary.py -a printing_usage_file
```

Output:

```
Printing users:
massimo.piccardi
MarkThomas
schmidt1974
```

In the case in which file *printing_usage_file* is empty, your program must instead only print:

```
No printing users
```

5. Your program can be invoked with option: -f. In this case, it must only print the following string:

```
Total number of files printed: <number of files printed>
```

Example with the example *printing_usage_file* given above:

Command line:

```
printing_summary.py -f printing_usage_file
```

Output:

```
Total number of files printed: 5
```

In the case in which file *printing_usage_file* is empty, your program must print:

```
Total number of files printed: 0
```

6. Your program can be invoked with option: -s. In this case, it must only print the following string:

```
Total number of bytes printed: <number of bytes printed>
```

Example with the example *printing_usage_file* given above:

Command line:

```
printing_summary.py -s printing_usage_file
```

Output:

```
Total number of bytes printed: 12030282
```

In the case in which file *printing_usage_file* is empty, your program must print:

```
Total number of bytes printed: 0
```

7. Your program can be invoked with option: -u *username*. In this case, it must print:

```
User username:
Total number of files printed: <number of files printed for user username>
Total number of bytes printed: <number of bytes printed for user username>
Largest file printed: <number of bytes of the largest file printed for user username>
```

Example with the example *printing_usage_file* given above:

Command line:

```
printing_summary.py -u massimo.piccardi printing_usage_file
```

Output:

```
User massimo.piccardi:
Total number of files printed: 3
Total number of bytes printed: 11755008
Largest file printed: 10000000
```

In the case in which file *username* is not present in *printing_usage_file*, your program must print:

```
User <username> not found
```

8. Your program can be invoked with option: -v. In this case, it must only print your name, surname, student ID and date of completion of your assignment, in a format of your choice. Please note that argument *argument_file* is still required.

9. No options can be used simultaneously. This means that your program can only be invoked with one of the options at a time.

10. If your program is invoked with any other syntax than those specified above, it must print a message of your choice **to the standard output** and exit.

Examples of incorrect syntax:

```
printing_summary.pl -Z argument_file
```

```
printing_summary.pl -u
```

Please be reminded that:

- **This assignment must be your own work and you should not be helped by anyone to prepare it in any way; your assignment may be tested by anti-plagiarism software that detects superficial changes such as changing variable names, swapping lines of code and the like.**
- **Understanding the assignment specifications is part of the assignment itself and no further instructions will be provided; on the other hand, whatever is not constrained you can implement it according to your own best judgment.**