University of the Western Cape
Department of Computer Science
Private Bag X17, Belville, 7535
South Africa
Tel: +27 (0) 21 959 2406
E-mail: ahenney@uwc.ac.za

Practical 1 [20 marks] Due not a second later than: 14th August 2019 at 11:54:59 PM

This practical aims to help you understand the nature of file systems, as used in decades of old. You've been provided with the file "crime.csv" on iKamva, along with this practical sheet. The file is quite a small excerpt of a file system database of crimes recorded in the city of Boston in the USA. The full file can be found on Kaggle: https://www.kaggle.com/AnalyzeBoston/crimes-in-boston in case you're interested.

Back in the day, companies stored their records in a file very similar to this one. In fact, even up to today, some companies still store their records in a file like this. In order to store information into, or retrieve information from, a pre-historic file like this, you need to know two things:

1. The structure of the file i.e. you need to know the ordering of the columns. So if you want to filter out records by date, you would need to know that the date is column number 4 in the file, and then use that info to do your filtering of records. This means that there is structural dependence in the file.

2. The data format of each field i.e. is it a number, string, date etc. So if you want to filter out records by date using column 4, you **also** need to know exactly how the date has been formatted. This means that there is data dependence in the file.

Ideally, however, you want some means of being able to store and retrieve information without caring how/where the data is stored in the file, or in what format the data is stored. What you want is a structurally independent and data independent data management system. This course will move towards this.

For now, this practical will demonstrate to you, practically, exactly what structural and data dependence are, and why they are undesirable. Your job is to take the role of a "data processing specialist/manager" back in the day, whose job was to write software to get answers to questions in a file.

The Task [20 marks]

Your job is to write a Python 3 program that uses the provided file and answers the following questions.

1. How many crimes were reported in the year 2015?
2. How many incidents involving shootings were reported in the year 2018?
3. How many "larceny" incidents occurred in 2017?
4. How many "drug violation" incidents occurred in all Districts "A" and "B" combined?
5. Which were the three most common offense codes in 2016 (sorted in descending order of the number of offenses)?
6. Which two years had the most robberies (sorted in descending order of the number of robberies)?

7. List the top three non-null street names in terms of the number of incidents reported (in descending order of the number of incidents reported, and then alphabetically if need be).
8. List the offense code groups of the three most common offense codes that occur on Fridays, Saturdays and Sundays combined in districts A, C and E combined (sorted in descending order of the number of offenses).

Format and Submission

Your main Python file must be called **mainfile.py**. If your submission does not contain a file with this name, the auto-marker will not be able to mark it and you will obtain a zero.

Your program should be able to take in exactly one command-line parameter (described below) and print out output depending on that parameter. The parameter specifies either the question number or student number (see the examples below). Below, I've provided you with exactly the output that your program should produce given the data set you've been given and given the input parameter specified. You can use this to cross-check your work (although if you think I've gotten one or more answers wrong, let me know; that would be unlikely though). Note that the first test case below prints out a student number of 1234567, but in your case it should print out your student number. It is critical that this works; this will be used to record your mark.

```
> python mainfile.py studentnumber
1234567

> python mainfile.py 1
12219

> python mainfile.py 2
39

> python mainfile.py 3
2442

> python mainfile.py 4
1554

> python mainfile.py 5
3115
3006
3831

> python mainfile.py 6
2016
2017

> python mainfile.py 7
WASHINGTON ST
BLUE HILL AVE
BOYLSTON ST

> python mainfile.py 8
Medical Assistance
Motor Vehicle Accident Response
Investigate Person
```

Please note that your work will be marked by an auto-marking script, and most likely using a different set of records than the ones you've been given. First of all, if your program output does not match the format specification above **exactly** (perhaps with the exception of case sensitivity of characters), your answer(s) will be marked wrong. No extra spaces, new lines or anything else. No exceptions will be made. You are 3rd years and I expect you to be able to handle something like this just before you go out to job market with the UWC brand on your back.

Second, while I've provided you with answers above, hard-coding won't help you since I'll test your code on a different set of records. Please focus on trying to make it work.

Also, again, please make sure to use only Python 3.

Please don't re-submit the data file along with your submission, but please **do** make sure you provide all other files and resources that you think will be necessary for your source code to run successfully on our machines e.g. if you've got multiple Python files etc.

Also, make absolutely sure that you leave more than enough time for your file(s) to be uploaded, transmitted across the network etc. Aim to submit your work at least 12 hours before the deadline. Absolutely no late submissions will be accepted and absolutely no excuses involving networks in computer labs/residences etc. not working will be accepted.