

Python Assignment 1

Min Wu <minwu.training@gmail.com>

1. There are 3 course list files: `course-list1.txt`, `course-list2.txt`, `course-list3.txt`. Each of them contains a list of courses that I copied from the old UCSC Extension website.

Implement `course-analysis.py` that displays all the common courses listed in all 3 files (download them into the same folder as `course-analysis.py`).

Hints:

- In each file, the line that includes a course name always ends with 'units' or 'unit'. You can use Python 3 string `endswith()` method.
- In the line that contains a course name, the course name starts from the beginning and ends at the position of the last comma in that line. You can use Python 3 string `rindex()` method.
- If a course name ends with a '*', it is a core course. Please remove the '*' from the course name.
- If you find the following error when you try to open a course list file,

```
UnicodeDecodeError: 'ascii' codec can't decode byte 0xe2...
```

Please add the encoding option, for example,

```
open('course-list1.txt', encoding='utf-8')
```

The following 4 courses should be displayed as common courses:

Perl Programming, Comprehensive

Python for Programmers

Python Programming for Beginners

Java Programming for Beginners

2. In information theory, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. In another way, it measures the minimum number of substitutions required to change one string into the other.

For example, the Hamming distance between

- **'karolin'** and **'kathrin'** is 3

- **'karolin'** and **'kerstin'** is 3

Implement the `hamming_distance()` function.

```
def hamming_distance(value1, value2):
```

```
    # write your code here
```

`value1` and `value2` can be a pair of strings.

Here is the result when executing the function:

```
>>> hamming_distance('karolin', 'kathrin')
```

```
3
```

```
>>> hamming_distance('karolin', 'kerstin')
```

```
3
```