

# CS 115 - Introduction to Programming in Python

## Lab 04

---

### Lab Objectives: Strings, Files, Modules

---

#### Notes:

- You should only use functionality covered in CS115 in your solution.
  - Your solution for this lab should not use lists, tuples, dictionaries.
  - Do NOT use the string `split()` function.
  - Include a docstring for your functions.
1. Download the files `doctor_data.txt`, `hospital_data.txt`, `medicana_docs.txt`. The first two files you can use as sample input files, and `medicana_docs.txt` is a sample output file for doctors at Medica (see below).
  2. Create a module, `doc_module.py` that contains the following functions:
    - a. `get_field()`: takes a string line, a string delimiter and a boolean, `is_begin`. The **default** delimiter is the tab character (`\t`) and the **default** `is_begin` is `True`. The line is a set of tokens, delimited by the delimiter character. If `is_begin` is `True`, the function will return the first token in the line, if `is_begin` is `False`, it will return the last token. (note each token has delimiter between).  
For example if the line contains: `'abc,def,ghi,jkl'` the delimiter is the comma, the first token is `'abc'` and the last token is `'jkl'`.
    - b. `filter_hospital()`: takes the name of a hospital, and 3 file names as input. You may assume the first file contains doctor information, the second file contains the id and hospitals of each doctor, and the third parameter is the name of an output file. You should find the hospital of each doctor in the first file (using data from the second file). For each doctor at the hospital with the given name (parameter) write the following fields to the output file, separated by a tab character: `doctor_id` `patients_per_day`. The number of patients given in the file is per month, you can calculate the patients per day assuming each month has 30 days. **Note: you should use functionality defined in the `get_field()` function where appropriate.**
    - c. `patient_averages()`: takes the name of a file. You may assume the file is tab-delimited and contains the id numbers of doctors and their daily number of patients. The function should return the daily patient average for all doctors in the file. **Note: you should use functionality defined in the `get_field()` function where appropriate.**
  3. Write a script, `yourname_doc.py` that inputs the name of a hospital from the user. Using functionality defined in the module, your program should write the information about all doctors at the hospital to the file `<hospitalname>_docs.txt`. Your program should also display the average patients per day at the input hospital.
  4. Save both your files in a zip file with the name: `Lab04_yourname.zip`.