

## INFS 740 – HA2- Python comprehensions

- Download and unpack archive HA2.zip. It has a number of files. If you use ATOM studio (which I recommend), under "File", choose "Add Project Folder" and select folder "HA2"
- 
- Assume a JSON database of the form as given in the collection "sampleUnivDB.json" (see file in the folder HA2). The meaning of the stored info is self-explanatory. For the purpose of queries below, assume that the possible grades are A, B, C and F; and that to satisfy a prerequisite for a class/course means to have taken the prerequisite courses (in transcript) with the grade of B or better.
- 
- Implement the queries in questions 1 and 2, by filling out the template in the file ha2lib.py (see file in the folder HA2). To run using Python:
  - 1. make HA2 your current folder in command line, and then:
  - 2. Run in command line: `>> python3 ha2_main.py > out.json`
- Your result will be in the output file out.json. Note that initially, ha2lib.py is a template with queries returning "tbd"; you need to replace "tbd"s with correct code.
- 
- Note that the file "queryAnswers.json" contains the correct answer to queries for the input in "sampleUnivDB.json". You can use it for debugging your queries.
- 
- Under the folder "testDBs", there are test JSON databases "db1", "db2", etc. as well as the JSON file "correct\_answers.json", to help you test and debug your queries. When you're ready to test "ha1lib.jq", you can generate a report on correctness of your queries as follows, run the following in command line from the current folder "HA2":

```
>> python3 ha2_produce_answers_main.py > ./testDBs/answers.json
>> zorba report_main.jq -o ./testDBs/report.json
```

Open report.json in Atom: you can see how many correct queries do you have out of how many, and gives you a per query report, including for which test databases it produced correct vs. incorrect answer. It is convenient to prettify report.json, and collapse it before you open the relevant parts.

- 1. Write Python queries returning True or False for each of the following logical sentences.
    - a. The student with ssn = 82 has taken the course "CS 530" (must be in Transcripts)

- b. A student named "John Smith" has taken the course "CS 530" (must be in Transcripts)
  - c. All students named "John Smith" has taken the course "CS 530" (must be in Transcripts)
  - d. The student with ssn = 82 has satisfied all prerequisites for each class she is enrolled in.
  - e. Every student has satisfied all prerequisites each class she is enrolled in.
  - f. Every student who majors in "CS" has satisfied all prerequisites for each class she is enrolled in.
  - g. A student named "John Smith" is enrolled in a class for which he did satisfied all prerequisites.
  - h. Some courses do not have prerequisites
  - i. All classes offered this semester have prerequisites.
  - j. Some students received only grades "A" or "B" in every course they have taken (must appear in Transcripts)
  - k. All students currently enrolled in classes taught by professor Brodsky (i.e., the name is "Brodsky" in faculty), major in "CS"
  - l. Some students who are currently enrolled in classes taught by professor Brodsky major in "CS"
- Write Python queries to express/compute each of the following sets (sequences) of:
    - a. All students { ssn: ..., name: ..., major: ..., status: ...} who have taken the course "cs530" (must be in transcripts)
    - b. All students { ssn: ..., name: ..., major: ..., status: ...} named "John" (i.e., name = "John" in student) who have taken the course "CS 530" (must be in transcripts)
    - c. All students { ssn: ..., name: ..., major: ..., status: ...} who satisfied all prerequisites each class they are enrolled in.
    - d. All students { ssn: ..., name: ..., major: ..., status: ...} who are enrolled in a class for which they have not satisfied all its prerequisites.
    - e. All students { ssn: ..., name: ..., major: ..., status: ...} named "John" who are enrolled in a class for which they have not satisfied all its prerequisites.
    - f. All courses {dcode: ..., cno: ....} that do not have prerequisites
    - g. All courses {dcode: ..., cno: ....} that do have some prerequisites
    - h. All classes {class: ..., dcode: ..., cno: ..., instr: ...} offered this semester that have prerequisites.
    - i. All students { ssn: ..., name: ..., major: ..., status: ...} who received only grades "A" or "B" in every course they have taken (must appear in Transcripts)
    - j. All CS students { ssn: ..., name: ..., major: ..., status: ...} who are currently enrolled in a class taught by professor Brodsky (name = "Brodsky" in faculty).