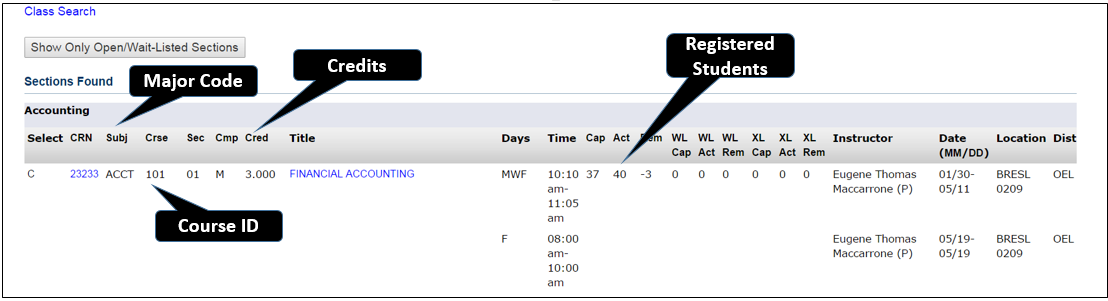
**CSC015 Assignment 10 (Regular Expression and Data Analysis)  
Total: 100pts.**

**What to submit:**(1) this WORD document with all screenshots required  
(2) as10.tar with all source files.

**Name: Abaan Khalid**

In this project, we will perform a financial analysis on Hofstra University. The analysis will be based on the public course catalog information available on the university website. By collecting the class size information of each listed course section (see Figure 1 below), and the tuition rate, we will able to estimate the total tuition income of the university. It is required that you address all the challenges in this assignment using a Python program.



**Figure 1.** Hofstra Course Catalog

**Challenge 1. Extract Course Sections. [30pts]**

The Fall 2018 course catalog is available for download in Assignment 10. Please download **hofstra.html** to your working directory. Your first job is to define a function named “**extractCourse**”, which returns **an array of strings**. **Each string should contain the entire record for one course.** (See a sample below in Figure 2).



**Figure 2.** One Record of a Course Section

It is required that you **use regular expression** to tackle the problem. The following are some ideas that you could use:

(1) Each record starts with a **<tr>** tag, which contains “**section\_row**” and it ends with a **</tr>** tag. A regular expression can easily capture the pattern. Pay special attention that **the shortest match operator \*?** Is better than \* in most cases.

(2) Once you’ve retrieved the matches of the above search pattern, you still need to do an additional round of filtering, because some records are fake records. They are actually the final exam date entries. You can inspect the content of the 3rd <td> tag of each record. If it is not **&nbsp;** it is a real course record; otherwise it is a final exam date entry. For accomplishing this check, you can use **another regular expression pattern to extract an array of <td> tag contents**, and check the 3rd element of the array.

(3) When construct the regular expression, using special form of compile function below:

**P1 = re.compile(“…your pattern here”, re.MULTILINE | re.DOTALL);**

Please call your function by **print(len(extractCourse())** and let us know **how many sections are offerred by Hofstra University?**

Please paste the **screenshot** below.

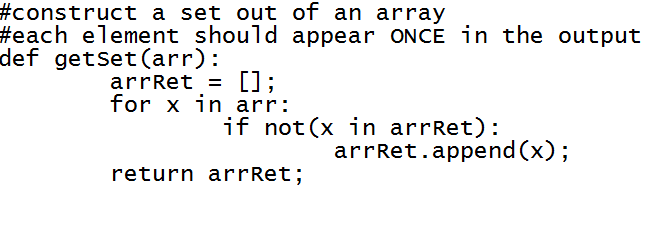
A black screen with white text

Description automatically generated

**Challenge 2. Majors. [30pts]**

Our next challenge is to figure out how many majors are offered by Hofstra. This is accomplished by looking at the major code (e.g., “ACCT” and “CSC”) in the course catalog. Since extractCourse() returns an array of strings, you can define an additonal function named “**extractMajors(arr)**”, which given the output of extractCourse(), returns an array of strings and each element is a **distinct major code**. (That is: each major code should appear exact once in the array).

Recall that to construct a set from an array, the following for loop does the job:



**Figure 3.** A function to extract the set of elements from an array

To accomplish this task, you can **define a regular expression to capture the patterns of <td>…</td> tags** and then the major code is the third element. Then adapt the code in Figure 3 to make sure that each major code appears exactly once in the array to be returned.

To demonstrate the success of your function please run your code and show us **all the major codes** and **how many majors are offered by Hofstra**.

Please paste the **screenshot** below.

A black screen with white text

Description automatically generated

**Challenge 3. Tuition Income. [20pts]**

Your final challenge is to figure out Hofstra’s tuition income in Fall 2018. The assumption is that: **each credit costs $1380** (ignore the difference between grad and undergrad sections). What is the total tuition income of the university?

Please paste the **screenshot** below.

A black screen with white text

Description automatically generated