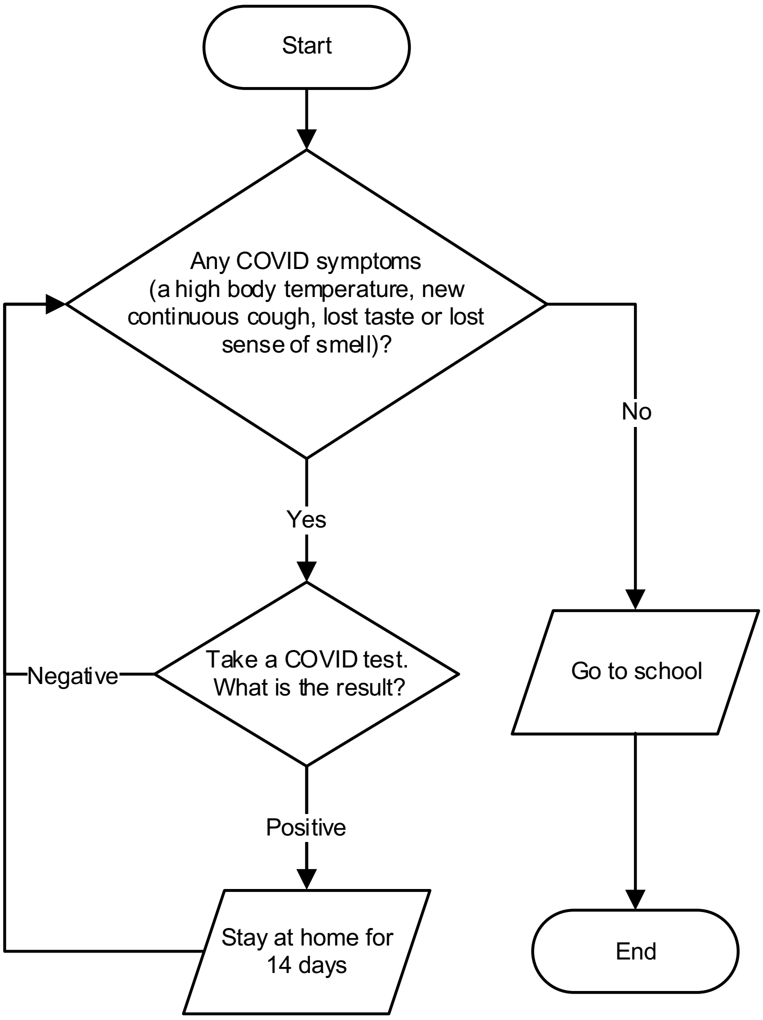
INS 107E
Int. to Prog. Lang. (Python)
2020-21 FALL SEMESTER
MIDTERM EXAM

December 17, 2020

EXAMINATION RULES AND REGULATIONS

- You have to be present at the announced ZOOM meeting and stay until you upload your answers or the exam ends.
- The exam will end at 19:45. Before 19:45, the answers must be uploaded to NINOVA Homework section for the midterm exam.
- The duration of the exam includes both solving the problems and uploading all the solutions to the NINOVA system. It is your responsibility to manage your time in order to submit your answers within the given duration.
- If a student is late to the exam or if the answers are not uploaded on time, the answers will not be accepted.
- The microphones will be muted during the exam. However, we prefer to have the cameras on.
- Students are not allowed to communicate in between each other during the exam by phone, email, messaging, etc. Students are recommended to avoid from suspicious behavior that can be evaluated as cheating. Instructor of the class has the right to not accept the answer of a student with suspicious behavior.
- The answers will be compared with others and if a cheating attempt is identified the students involved in it will get zero points from that question.
- During the exam you may use your lecture notes.
- The answers of all questions are supposed to be uploaded as separate files in one of the following file formats: Python Script (.py), Word (.docx), PDF, ZIP, JPEG or RAR. The size limit for uploaded files is 50 MB.
- You may solve the questions by using your computer or simply write on white, flat A4 papers with your own handwriting. You should write your name and number on the top of EVERY page and sign. Papers without this information will be disqualified.

1) Write a script for the COVID Symptoms flow chart given below.


Start
Any COVID symptoms (a high body temperature, new continuous cough, lost taste or lost sense of smell)?
No
Yes
Take a COVID test. What is the result?
Negative
Positive
Go to school
Stay at home for 14 days
End

The answers must be uploaded until **19:45** Points: **15+15+15+15+15+15+15**

2) Write a script that will perform the following:

a) Create a dictionary named `faculty_codes` to represent the information given in the table below.

Code	Faculty
01	Civil Engineering
02	Architecture
03	Mechanical Engineering
04	Electrical and Electronics Engineering
05	Mines
06	Chemical and Metallurgical Engineering
07	Management
08	Naval Architecture and Ocean Engineering
09	Science and Letters
10	Aeronautics and Astronautics
11	Turkish Music State Conservatory
12	Maritime
13	Textile Technologies and Design
14	Computer and Informatics Engineering

b) Request the identification number of an ITU student to create the variables below:

Example: 010970255

`faculty_code`	First two digits from left to right. (**01**)
`acceptance_year`	Use the fourth and fifth digits from left to right (**97**) to generate the full year format, i.e., **97** would be **1997**, and **12** would be **2012**.
`ranking`	Last three digits from left to right. (**255**)

c) Compose the sentence below regarding the student ID number given by the user and print it out.

Sample printout format:

You were admitted to the **Faculty of Civil Engineering** in the year of **1997** with the ranking of **255**.

3) Student name, student number, midterm points, final points and grade are stored in a list as shown below.

Write a script which will sort the list according to any random criterion determined by the user by using the `input` command (such as Name, Number, Midterm, Final, Grade). The script should ask the user whether sorting will be in the ascending or descending order. Please write your own sorting script, **do not** use the `sort` method or `sorted` function.

Name	Number	Midterm	Final	Grade
Ahmet	0100130075	50	70	62
Merve	0100140176	60	75	69
Aslı	0100130082	80	70	74
Burak	0100150101	40	80	64
….				
….				
….				
….				

⇨

```
my_class = [
      ["Ahmet", "0100130075", 50, 70, 62],
      ["Merve", "0100140176", 60, 75, 69],
      ["Asli", "0100130082", 80, 70, 74],
      ["Burak", "0100150101", 40, 80, 64],
       ...
       ...
       ...
       ... ]
```

Exam duration is **90 minutes.** Points: **15+15+15+15+15+15+15**

4) Write a script which will divide the list `A` from the middle and create a new list `B` by adding the first half to the end of it. If the list has odd number of items, keep the additional item in the first half when dividing the list.

Examples:

```
A = [1, 2, 3, 4, 5, 6]   ⟹   B = [4, 5, 6, 1, 2, 3]
A = [1, 2, 3, 4, 5]      ⟹   B = [4, 5, 1, 2, 3]
```

5) Write a script which will check whether the number series in the list `L` is monotone increasing or not. If the series is not monotonic, print out the number which disrupts the monotonicity.

Note: Monotone increasing series is an always increasing number series, i.e., the next number in the series is always greater than the previous.

Examples:

```
L = [2, 6, 8, 11, 14, 17]  ⟹  The series is monotone increasing.
L = [1, 5, 8, 6, 9, 10]    ⟹  The series is not monotone increasing due to 6.
```

6) Write a script that will perform the following:

a) Create a dictionary named **`laptops`** to represent the information given in the table below.

Brand	CPU	RAM	Storage
Lenovo	Intel Core i7	16 GB	512 GB
HP	Intel Core i7	8 GB	512 GB
Apple	Apple M1	16 GB	1 TB
Dell	Intel Core i7	16 GB	512 GB
Acer	Intel Core i5	8 GB	128 GB
Asus	AMD Ryzen	16 GB	1 TB

b) Request the laptop brand from the user as an input. From your dictionary dataset, compose the sentence below regarding the given laptop brand and print it out.

Sample printout format:

Lenovo laptop has the following properties: CPU: **Intel Core i7**, RAM: **16 GB**, Storage: **512 GB**.

7) According to the Turkish Seismic Design Code (2019), the concrete compressive strength of an existing building (f_{cm}) is determined by considering uniaxial compression tests performed on concrete core specimens. The concrete compressive strength is defined as:

$$f_{cm} = Maximum\ (0.85 f_{c,mean};\quad f_{c,mean} - StDev)$$

where, $f_{c,mean}$ is the mean compressive strength and $StDev$ is the standard deviation of the test results.

A series of tests are carried out and the following compressive strength values (in MPa unit) are obtained:
R = [22, 26, 20, 24, 28, 18, 30, 15, 32, 26]

Prepare a script that stores the given test results in a list and then calculates and returns the $f_{c,mean}$, $StDev$ and f_{cm} values by printing them on the screen.

$$f_{c,mean} = \frac{\sum_{i=1}^{n} f_{c,i}}{n}, \qquad StDev = \sqrt{\frac{\sum_{i=1}^{n}(f_{c,i} - f_{c,mean})^2}{n-1}} \qquad (n \text{ is the number of tests})$$

Exam duration is **90 minutes.** Points: **15+15+15+15+15+15+15**