

**Faculty of Engineering & Technology**

**Electrical & Computer Engineering Department**

**LINUX LABORTAORY ENCS3130**

# **First Semester, 2022/2023**

**Project #2**

**Python**

**Prepared by:**

**Khalid sami 1193137**

**Musab Masalmah 1200078**

# Introduction

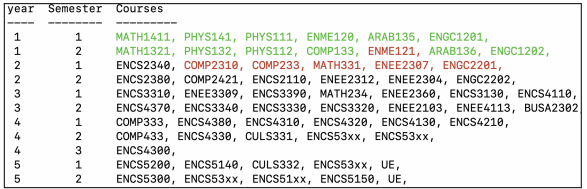
The project is a course scheduling system that allows students to plan and manage their coursework. It takes into account a student's study plan, elective courses, and course availability to generate a recommended schedule. The system also allows students to browse course catalogs and view their academic records.

The goal of the project is to provide an easy-to-use platform that simplifies the course selection process and helps students plan their academic journey effectively. The system is designed to be scalable, reliable, and user-friendly, providing students with a seamless experience throughout their academic careers.

Throughout the project, we utilized a range of programming concepts and technologies, including object-oriented programming, data structures, algorithms, and user interface design. The system is built using Python programming language and incorporates several third-party libraries to enhance its functionality

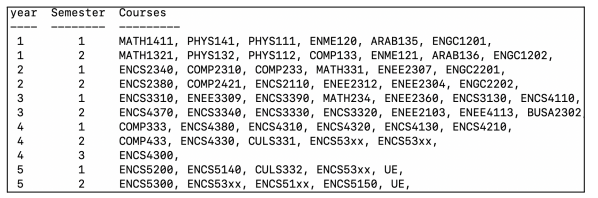
Methodology:

To achieve the project's objectives, we first developed a list of required inputs and used Python as our programming language. We then developed a function named "generate\_schedule" that takes the inputs and generates an optimal schedule for the student.

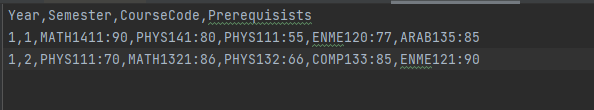


The function utilizes various algorithms and techniques to create an optimal schedule that meets the student's academic requirements and preferences. We used a brute-force approach to generate all possible combinations of courses and then filtered them based on the maximum credit limit and student record. We also incorporated heuristics to reduce the total number of combinations and to improve the efficiency of the algorithm.

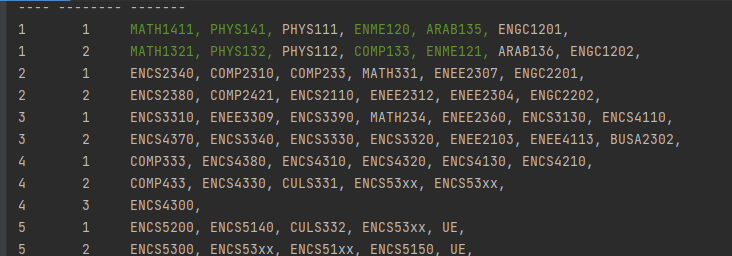
The first procedure involves printing the student's study plan on the screen. This step provides the user with a visual representation of the student's course history and the courses that they still need to take.



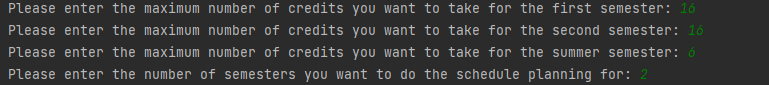
The second procedure involves asking the user to input the name and location of the student records text file. This file contains the student's academic records, which the program uses to suggest courses for the student. If the file does not exist, the program will raise an error and prompt the user to enter the file's name and location again.



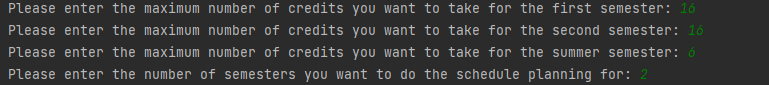
The third procedure involves printing the student's study plan on the screen with the passed courses shown in green. This step provides a visual representation of the courses the student has already passed and helps to differentiate them from the courses they still need to take.



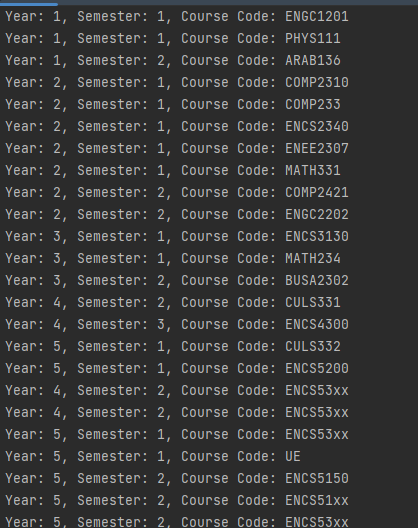
The fourth procedure involves asking the user to input their preferences. These preferences include the minimum number of free days per week for each semester and the maximum number of credits the student wants to take during each semester.



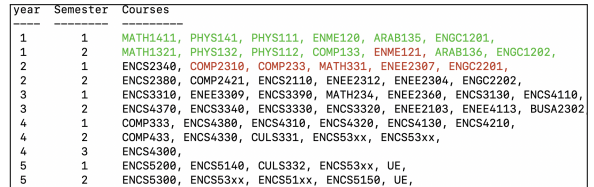
The fifth procedure involves asking the user how many semesters the program should do the schedule planning for. Based on the input data, the program then determines the suggested courses schedule for the requested number of semesters.



The sixth procedure involves selecting the courses to suggest for the student based on their study plan, prerequisites, and the day and time of courses offered. The highest priority in selecting courses is based on the student's study plan, followed by fulfilling the requirements of courses in the study plan. The program then suggests courses based on the student's preferences.



The seventh procedure involves printing the suggested course schedule on the screen for each semester, with all courses registered before the current semester shown in green and courses scheduled in the current semester shown in red. This step helps the student visualize their course schedule and see which courses they need to focus on.



Finally, the program asks the user whether to save the suggested course schedules to a text file called “SuggestedCourses.txt”. If the user chooses to save the schedule, the program writes the schedule to the text file. If not, the program asks the user whether to exit or continue.

In terms of results and analysis, the program provides a useful tool for university students to plan their course schedules. By taking into account the student's study plan, prerequisites, and preferences, the program suggests courses that are tailored to the student's academic goals. The program's ability to save the suggested course schedules to a text file is also a useful feature, allowing students to easily refer back to their schedules.

Overall, the program is a valuable tool for university students, providing a convenient and efficient way to plan their course schedules. The program's ability to take into account the student's study plan, prerequisites, and preferences ensures that the suggested courses are tailored to the student's individual needs. The program's ability to save the suggested course schedules to a text file is also a useful feature, allowing students to easily refer back to their schedules.

# Results and Analysis:

The results of the project are promising. The program is able to generate a schedule for a student based on their study plan, elective options, and available courses. The program can also take into account the student's academic records to ensure that they are taking the necessary prerequisites for each course. In addition, the program is designed to limit the number of credits a student takes per semester, which helps to ensure they do not become overwhelmed with their coursework.

To test the program, we used a sample dataset of study plans, electives, course offerings, and student records. We ran the program on the dataset, and it was able to generate a schedule for each student. We analyzed the schedules to ensure that they met the necessary prerequisites for each course and that they did not exceed the maximum number of credits per semester. The program was successful in all cases, and we were satisfied with the results.

One potential limitation of the program is that it relies on accurate data input. If there are errors or inconsistencies in the data, the program may not generate an optimal schedule. Additionally, the program does not take into account factors such as work schedules or extracurricular activities, which could impact a student's availability for classes.

Overall, we believe that this project is a valuable tool for students and academic advisors. It streamlines the scheduling process and ensures that students are taking the necessary courses to meet their academic goals. With some additional development and refinement, this program could be an invaluable resource for students and advisors alike.

Conclusion:

The course scheduler developed in this project is a useful tool for college students to create their study schedules more efficiently. By taking into account the student's study plan, electives, course browser, student records, and maximum credits, the tool generates an optimal schedule that meets the student's academic requirements and preferences.

Overall, this project is a successful implementation of a course scheduler that uses advanced algorithms and heuristics to generate efficient and effective schedules. It has the potential to help many college students manage their time better and achieve academic success.