

Description about the program,

This Python script is designed to facilitate the assessment of student progression based on their credit accumulation within a university course. The program allows both students and staff members to input and process data regarding the credits attained in different modules, categorizing the progression outcomes into four categories: "Progress," "Progress (module trailer)," "Module retriever," and "Exclude."

Key Features:

- **Student Mode:** In this mode, students input the number of credits they have achieved in three categories: Pass, Defer, and Fail. Based on these inputs, the program determines the student's progression outcome and displays it to the user. The outcomes are categorized based on predefined criteria.
- **Staff Mode:** Staff members can input multiple sets of data, allowing them to assess the progression of multiple students. After each input, the staff member can choose to continue entering data or quit the program. Once the staff member decides to quit, the program displays the overall outcomes in a graphical histogram and generates a text file containing all the data.
- **Histogram Visualization:** The program generates a graphical histogram representing the distribution of progression outcomes, with bars indicating the number of students in each category: Progress, Progress (module trailer), Module retriever, and Exclude.
- **Output File Generation:** After processing all the data, the program generates a text file containing detailed information about each student's progression outcome. This file serves as a record for further analysis or documentation purposes.

Tools and Techniques Used:

- **Python Programming:** The entire script is written in Python, utilizing various programming constructs such as loops, conditional statements, functions, and exception handling to achieve the desired functionality.
- **Graphics Module:** The graphics module is employed to create a graphical representation of the histogram, visualizing the distribution of progression outcomes in a user-friendly manner.
- **File Handling:** The script utilizes file handling techniques to create and write data to a text file, enabling the generation of a record containing detailed progression information.
- **User Input Validation:** Input validation techniques are implemented to ensure that the user provides valid integer inputs within the specified range, enhancing the robustness of the program.