

Data Structures & Algorithms

Python Programming Assignment 1 Set A (40 marks)

SUBMISSION REQUIREMENTS:

- 1) Name the **ZIP files** containing your solutions according to the following requirements:
ZIP all Python files to be submitted into a zip file and named it as "**ADMINNO_ASSN.zip**". For example:
"**123456F_ASSN.zip**".
- 2) At the beginning of every Python files to be submitted, include your "**Name, Student Admin no. and Tutorial Group**" as comments.
- 3) Submission Due Date: **Sun, 15 June 2025, 2359hrs (Week 8)**
- 4) The deliverable is to be submitted to your module tutor via Bright Space
- 5) Late Submission: Marks will be deducted for late submission as per the NYP RGO guidelines for late submission.
- 6) Code review will be scheduled in week 8 during LAB and TUT sessions. There will be code walkthrough and technical question asked about the assignment completed by student.

Student Course Registration System

Assignment 1 Question (40 marks)

You are required to develop a Student Course Registration System that captures and manages student data and course data efficiently. Each student record must include:

<u>Field Name</u>	<u>Data Type</u>	<u>Example 1</u>	<u>Example 2</u>
Name	String	Alice Boo	Johan Power
Student ID (Primary Key)	Integer	10001	10045
Email	String	alice.boo@poly.edu	johan.pow@poly.edu
Course List	List[String]	["IT101", "CS112"]	["IT111", "SF102"]
Year of Study	Integer	1	2
Full-time/Part-time Status	Boolean	True (Full-time)	True (Full-time)

Main Requirements

1. Create a Student class with the above listed fields and include methods to add courses, remove courses, and display student details (5 marks).
2. Use a data structure (e.g., Python List or Dictionary) to manage all student records, allowing efficient access and updates (2 marks).
3. Develop a menu driven application that allows users to perform the following:
 - a. Display all student records (5 marks).
 - List all students with their full details.
 - b. Add a new student record (5 marks).
 - Prompt the user to enter all required student data.
 - Validate all input (i.e. unique ID, valid email format, and others).
 - Insert the new student record into the data structure.
 - c. Enrol student for a new course (5 marks).
 - Allow the user to select a student by ID.
 - Prompt for a course code.
 - Add the course to the student's course list (if not already registered).
 - d. Sort students by Year of Study [Bubble Sort, Ascending] (5 marks).
 - Implement Bubble Sort to order students by Year of Study.
 - Display the sorted list.
 - e. Sort students by Num of Registered Course [Selection Sort, Descending] (5 marks).
 - Implement Selection Sort to order students by number of Registered Courses.
 - Display the sorted list.
 - f. Search for a student by ID or Name (2 marks).
 - Allow the user to enter an ID or name and display the matching student's details.
 - g. Exit the program (1 mark).
 - Cleanly exit the application, saving data if needed.

Optional Requirements

4. Additional features that enhance the application's usability (5 marks).
 - a. Access control for specific application features or student data.
 - b. Logging functionality to track actions performed on student records.
 - c. Export student or course data to formats such as PDF or Excel.
 - d. Other features may include layout improvements or aesthetic enhancements.

Assessment Rubrics (40 marks)

Technical Walk Thru (Interview) and Submission.

Individual (100% - 40 marks)					
Category	Excellent (A)	Very Good (B)	Good (C)	Satisfactory (D)	Marks
[I] Main Requirements	<p>Completed all (or more) menu functions listed in the user requirements for record processing with no errors.</p> <p>Extensive coverage of validation rules ensures data submitted by users is in the correct format for all fields and with excellent validation messages to meet user needs.</p>	<p>Completed 3 menu functions listed in the user requirements for record processing with no errors.</p> <p>Good coverage of validation rules ensures data submitted by users is in the correct format for all fields and with excellent validation messages to meet user needs.</p>	<p>Completed 2 menu functions listed in the user requirements for record processing with no errors.</p> <p>Reasonable coverage of validation rules ensures data submitted by users is in the correct format for all fields and with excellent validation messages to meet user needs.</p>	<p>Completed 1 menu function listed in the user requirements for record processing with no errors.</p> <p>Implement validation rules with minimum validation messages in meeting user needs.</p>	[I]
[II] Optional Requirements	Completed 4 (or more) additional features that enhance the usability of the overall system.	Completed 3 (or more) additional features that enhance the usability of the overall system.	Completed 2 (or more) additional features that enhance the usability of the overall system.	Completed 1 additional feature that enhances the usability of the overall system.	[II]
[I + II]			Total Marks (40 marks)		

-- end of page --