Data Structures & Algorithms
Python Programming Assignment 2 Set A (60 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 file to be submitted, include your "**Name, Student Admin no. and Tutorial Group**" as comments.

3) Submission Due Date: **Sun, 17 August 2025, 2359hrs (Week 17)**

4) The deliverable is to be submitted to your module tutor via Bright Space

5) Code review will be scheduled in week 16/17 during LAB and TUT sessions. There will be code walkthrough and technical question asked about the assignment completed by student.

6) Late Submission and Penalty:

For late submissions of 5 calendar days or less,
- If you passed the assignment/project deliverable, your score will be capped at 50% of the base score of the assignment/project deliverable.
- If you failed the assignment/project deliverable, you will be awarded a failed score.

For late submission or more than 5 calendar days,
- You will be awarded zero marks for the assignment/project deliverable.

Please note:
- It is your responsibility to ensure that your submission is complete and correct. You may wish to download a copy after submission to verify that all files are included and accurately submitted.
- If you have multiple submissions, the most recent submission will be considered as your final version and will be subjected to penalty if it is submitted late.

7) Academic Integrity

The following are considered acts of academic dishonesty :

- Attempting to get or provide unauthorised assistance in an assessment
- Falsifying data, information or citations
- Asking another person to complete the work that the candidate is supposed to do
- Taking and using the whole or any part of ideas, words or works of other people and passing it off as one's own work without acknowledgement of the original source

If a learner is caught cheating for the:

- First time in any assessment, he/she will fail the entire learning unit.
- Second time in any assessment, he/she will fail all learning units in that semester.
- Third time in any assessment, he/she will be removed from the Polytechnic.

Student Course Registration System (Phase II)

Assignment 2 Question (60 marks)

Phase II of the Student Course Registration System enhances sorting functions, improves student request handling, and boosts overall system usability. These requirements include:

Main Requirements		
1.	Quick Sort on Year of Study with Secondary Sort (on Name) ▪ Create a function to sort student records by Year of Study (ascending) using Quick Sort. ▪ If multiple students have the same Year of Study, sort them further by Name (ascending). ▪ Display results in a clear tabular format.	7 marks
2.	Merge Sort by Num of Registered Course and Student ID ▪ Use Merge Sort to sort students by Num of Registered Course and then by Student ID, both in ascending order. ▪ Add a filter to display students from a selected Year of Study.	8 marks
3.	Manage Student Requests (Enhanced Queue System)	-
3a.	Priority Queue with Timestamp ▪ Create a StudentRequest class with: Student_ID, Request_Type, Priority_Level, Request_Details and timestamp. ▪ Implement a Priority Queue: lower Priority_Level = higher priority. ▪ If equal priority, process by timestamp (FIFO). ▪ Allow duplicate Student_IDs. ▪ Other features deemed relevant.	10 marks
3b.	Student ID Validation with Search ▪ Before adding a student request, validate if the Student_ID already exists using a search algorithm (binary or sequential). ▪ If found, prompt confirmation before adding duplicate student request. ▪ Other features deemed relevant.	6 marks
3c.	View Student Requests Queue Statistics ▪ Provide a function to display total requests in Queue. ▪ Filter by Request_Type or Priority_Level. ▪ Other features deemed relevant.	4 marks
3d.	Process Next Student Requests and Logging Create a function to process next requests in priority queue: ▪ Show request details. ▪ Remove it from queue. ▪ Show updated request count. ▪ Log processed request to a file.	10 marks

	▪ Other features deemed relevant.	
4.	<u>Dashboard Summary View</u> Implement a Dashboard View accessible from the main menu. It should display: ▪ Total number of students. ▪ Number of full-time and part-time students. ▪ Most common course enrolled. ▪ Total number of pending student requests. ▪ Other features deemed relevant.	5 marks
<u>Optional Requirements</u>		
5.	Implement additional validation, algorithmic functions, enhancements across data structures, and aesthetic features to improve the usability of the application system. *The level of complexity will be a significant factor in the assessment.* ▪ Undo and Redo for Student Requests Queue using Stacks. ▪ Represent the Student Course Registration System using a Tree data structure. ▪ Maintaining a linked list to track each student's course registration history. ▪ Data Encryption for Sensitive Fields using simple encryption algorithm. ▪ Real-Time Analytics Dashboard to track and display live statistics. ▪ Generative AI-powered interactive command-line chatbot assistant (Generative API Integration) for course advisory. ▪ Biometric Login Simulation using image processing libraries for login. ▪ Other challenging and advanced features as deemed relevant.	10 marks

Assessment Rubrics (60 marks)

Technical Walk Thru (Interview) and Submission.

Individual (100% - 60 marks)					
Category	**Excellent** (A)	**Very Good** (B)	**Good** (C)	**Satisfactory** (D)	**Marks**
[I] Main Requirements	Completed all (or more) menu functions listed in the user requirements for student record processing with no errors. 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.	Completed 3 menu functions listed in the user requirements for student record processing with no errors. 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.	Completed 2 menu functions listed in the user requirements for student record processing with no errors. 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.	Completed 1 menu function listed in the user requirements for student record processing with no errors. Implement validation rules with minimum validation messages in meeting user needs.	[I]
[II] Optional Requirements	Completed 4 (or more) additional features that enhance the usability of the overall system. Marks based on difficulty of additional features implemented.	Completed 3 (or more) additional features that enhance the usability of the overall system. Marks based on difficulty of additional features implemented.	Completed 2 (or more) additional features that enhance the usability of the overall system. Marks based on difficulty of additional features implemented.	Completed 1 additional feature that enhances the usability of the overall system. Marks based on difficulty of additional features implemented.	[II]
			[I + II]	**Total Marks** (60 marks)	

END OF PAPER