EEL3834 - Programming for Electrical Engineers
Fall 2025
Programming Assignment 6: High Score Lab
Due: 11/09/2025 @ 11:59PM
To be done individually

Objective:

Create a small, menu-driven High Score Lab program that manages entries in a manually managed dynamic array. You will practice raw pointers, dynamic allocation and resizing, ownership and RAII (Rule of Three in C++), and contrasting pointer- vs reference-style helpers in C++. In Python, you will emulate capacity and keep identical prompts, validations, and outputs.

Problem Description:

Create a console app named High Score Lab managing entries:

- ID: positive, unique integer (player ID)
- Score: integer within [0 … 1,000,000]

All entries are stored in one dynamic array that you manage (size + capacity). The UI is a menu loop that repeats until Exit.

Menu:

High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit

Enter your choice:

Menu Features:

1. Add Entry:
    - Prompts the user for ID and to enter score
    - If ID <= 0 print “Error: ID must be greater than 0.”
    - If ID already exists print “Error: ID already exists.”
    - If score not in range of [0 – 1,000,000] print “Error: Score out of range.”
    - Once a successful entry is inputted print “Entry added!”

2. View Entry
    - Prompts the user for ID
    - If ID is found print “ID: <id> | Score: <score>”

- If it is not found print “Error: ID not found.”

3. Update Score
   - Prompts user to enter in ID, new score and to update it with pointers or reference
   - If ID is missing print “Error: ID not found.”
   - If score not in the range print “Error: Score out of range”
   - If user enters a method that is not 1 or 2 print “Invalid choice.”
   - For C++ for this method must have two distinct helpers
      i. Pointer-based updater
      ii. Reference-based updater

4. Remove Entry
   - Prompts for ID
   - If no ID is input or not one that exists print “Error: ID not found”
   - Once successful print “Removed.”

5. Leaderboard (Top K)
   - K is how many results to show
   - Prompts user to Enter K
   - If K <=0 print “Error: K must be greater than 0”
   - Print top K entries scored by score descending, tie breaker by smaller IDs first print the results in this format “<rank>. ID: <id> | Score: <score>”
   - If there are no entries, print nothing after asking for K and return to menu.
   - Implementation note: sort a temporary copy, do not reorder the stored array

6. Stats
   - Prints results:

Entries: <count>
Total score: <sum>
Average score: <float with 2 decimals>
Capacity: <current capacity>

7. Pointer Tools
   - Submenu
   - a) Boost by pointer
      i. Prompt user to enter ID and then enter delta
      ii. Apply delta with the C++ pointer helper, and in python you can call a function with the same intent.
   - b) Boost by reference
      i. Prompt user to enter ID and then enter delta

ii. Apply delta with the C++ reference helper, and in python you can call a helper with identical effects(just note in your code that python doesn't distinguish pointer vs reference just keep the UI the same).

iii. If a negative delta would cause a score to be below 0, have the score stop to 0.

iv. Apply delta with reference helper

- c) Back
- For any input besides a,b,or c print "Invalid choice"

8. Exit
   - Print "Goodbye!" and end the program

For any invalid main menu choice print "Invalid choice"
Any non-numeric where a number is required print "Invalid input" then reprompt

Code Requirements

C++

- Maintain Entry* data, int size, int capacity.
- Begin with a small array (capacity 4), and whenever the array becomes full (size = capacity), allocate a new array with a larger capacity (typically double), copy the existing elements over, delete the old array, and continue using the new one.
- Implement Rule of Three to guarantee deep copies and proper cleanup:
  - Destructor to delete[] data
  - Copy constructor to deep copy the buffer
  - Copy assignment to deep copy with self-assignment guard
- Distinct helper functions to update/boost through a pointer and a reference
- All prompts and messages must match the spec exactly.

Python

- Use a list to emulate a dynamic array; track size and capacity explicitly.
- On growth, expand capacity and ensure only first size entries are considered valid.
- Store entries in a list and track size/capacity.
- For Menu 3 and 7, accept 1 or 2; both call helpers that update the selected entry and clamp the score.
- Use only built ins; no imports.
- Python has no raw pointers; your Python version should simulate capacity and pointer/reference behavior but will not use actual pointers.

Sample runs *(please note you want your assignment to have the exact wording and layout or you will lose points)*

Run 1: (red is user input, blue is where you should have a new line (an empty line))

```
High Score Lab
1. Add Entry
2. View Entry
```

```
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 9
Invalid choice.
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 2
Enter ID: 10
Error: ID not found.
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 1
Enter ID: 0
Error: ID must be greater than 0.
Enter ID: -5
Error: ID must be greater than 0.
Enter ID: 101
Enter Score: -1
```

```
Error: Score out of range.
Enter Score: abc
Invalid input.
Enter Score: 1000001
Error: Score out of range.
Enter Score: 9000
Entry added!
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 1
Enter ID: 101
Error: ID already exists.
Enter ID: 202
Enter Score: 45000
Entry added!
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 2
Enter ID: 101
ID: 101 | Score: 9000
(new line here)
High Score Lab
```

```
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 3
Enter ID: 999
Enter new Score: 12000
Method (1=pointer, 2=reference): 2
Error: ID not found.
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 3
Enter ID: 101
Enter new Score: 1500000
Method (1=pointer, 2=reference): 2
Error: Score out of range.
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
```

```
Enter your choice: 3
Enter ID: 101
Enter new Score: 15000
Method (1=pointer, 2=reference): 7
Invalid choice.
Method (1=pointer, 2=reference): 1
Updated.
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 7
a) Boost by pointer
b) Boost by reference
c) Back
Enter choice: z
Invalid choice.
Enter choice: a
Enter ID: 404
Enter delta: 250
Error: ID not found.
Enter choice: a
Enter ID: 101
Enter delta: 2000000
Boosted.
Enter choice: b
Enter ID: 202
Enter delta: -999999
Boosted.
Enter choice: c
(new line here)
High Score Lab
```

```
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 2
Enter ID: 101
ID: 101 | Score: 1000000
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 2
Enter ID: 202
ID: 202 | Score: 0
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 5
Enter K: 0
Error: K must be greater than 0.
(new line here)
```

```
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 5
Enter K: x
Invalid input.
Enter K: 5
1. ID: 101 | Score: 1000000
2. ID: 202 | Score: 0
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 6
Entries: 2
Total score: 1000000
Average score: 500000.00
Capacity: 4
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
```

```
7. Pointer Tools
8. Exit
Enter your choice: 4
Enter ID: 999
Error: ID not found.
Enter ID: 202
Removed.
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 5
Enter K: 3
1. ID: 101 | Score: 1000000
(new line here)
High Score Lab
1. Add Entry
2. View Entry
3. Update Score
4. Remove Entry
5. Leaderboard (Top K)
6. Stats
7. Pointer Tools
8. Exit
Enter your choice: 8
Goodbye!
```

Expectations

Your grade will be subject to the following condition(s):

- Submission:
  The submission deadline is **11:59PM** on **11/09/25**. You will be penalized in increments of 25% per day late (regardless of the time). Submit your code on Canvas. You just need

to **upload** your .cpp file and .py file, not copy and paste your code. Please name your .py and .cpp file as firstname_lastname_assignment6.py /.cpp

- Correct Commenting
    - There should be comments on key areas in your code explaining why you chose certain aspects and what they are doing, a good rule of thumb is you should have a couple of comments every 10 lines of code
- Correct Variable Naming
    - Naming variables in relation to their purpose.
- Code working correctly/compiling
    - Please note: Although your can use whatever compiler you want, your assignment will be graded in the class compiler (VS code and Pycharm) so it must compile with these compilers, or you may lose points
- Code print format is as shown in sample runs

*See rubric on canvas assignment for more details.