

Assignment 2 – Exploring Vectors

Objectives

- Practice using C++ vectors (dynamic arrays).
- Learn basic vector operations: **push_back**, **insert**, **erase**, **size**, **capacity**.
- Write simple functions that operate on vectors.
- Begin exploring Big-O notation: what it means and why it matters.

Submission Guideline

- Submit three **.cpp** files with your code.
- Add a comment block with your reflections at the end of the .cpp file.
- Make sure your code compiles and runs on an IDE.
- Use clear comments in your code explaining your understanding.

Grading Rubric (20 points total)

1. Scenario 1 (Cafeteria Menu): **5 points**
2. Scenario 2 (Club Attendance + functions): **7 points**
3. Vector Growth Experiment: **4 points**
4. Reflection (Big-O basics + vector behavior): **4 points**

Accessibility Services

If you are a student with a disability, either temporary or permanent, please reach out to the Office of Accessibility Services (OAS) at accessibilityservices@jjay.cuny.edu. We are here to assist you in achieving your academic goals.

OAS works in partnership with the entire John Jay community to ensure access to all areas of campus life and arrange for appropriate academic adjustments, programs and services. We provide referrals to both on-campus offices and outside agencies to support students' success.

Reminder: Copy-pasting from AI or online sources without understanding is plagiarism. You should be able to explain your code in person if asked.

Part A – Coding Tasks

Scenario 1: John Jay Cafeteria Menu

The John Jay cafeteria updates its menu throughout the day. Dishes can be added, inserted, or removed.

- Create a **vector<string>** called **menu**.
- Add 5 dishes using **push_back**.
- Insert a new dish at the **2nd position**.
- Remove the 4th dish using **.erase()**.
- Print the final menu using a **range-based for loop**.

Scenario 2: Student Club Attendance

You're managing sign-ups for a John Jay student club event.

- Use a **vector<int>** to store student IDs.
- Add 10 student IDs (you may hardcode values).
 - Or, you can ask for user input using **cin**
- Write a function **double getAverage(const vector<int>&)** that returns the average student ID (treat **IDs as numbers** for practice).
- Write another function **int getHighest(const vector<int>&)** that returns the highest student ID.
- In **main()**, call both functions with your vector and print the results (the average and the highest student ID).

Part B – Short Reflection

Answer briefly in your own words (2–4 sentences each):

- What is **Big-O Notation**? Explain it simply (no math needed).
- Why is Big-O important for programmers? Give one real-life example (e.g., searching names in a class roster).
- From your scenario (Part A):
 - Why was vector a better choice than array for this problem?

Reminder: Copy-pasting from AI or online sources without understanding is plagiarism. You should be able to explain your code in person if asked.