Hands_on_12

1. Implement a dynamic array (that is a C++ vector). You are only allowed to use C style arrays. Assume the datatype is an int

Explanation:

- ⇨ To implement a dynamic array in C++ using C-style arrays, we can create a class that manages the memory allocation, resizing, and element access. Below is a simple implementation of a dynamic array (similar to std::vector<int>) that uses C-style arrays.

- ⇨ DynamicArray Class: This class encapsulates the dynamic array functionality.
  Private Members:
- ⇨ data: A pointer to the dynamically allocated array.
- ⇨ capacity: The total capacity of the array.
- ⇨ size: The current number of elements in the array.
- ⇨ resize() Method: This method is responsible for resizing the array when needed. It allocates a new array, copies the existing elements, and frees the old array.
- ⇨ push_back() Method: This method adds a new element to the end of the array. If the array is full, it resizes the array.
- ⇨ at() Method: This method retrieves an element at a specific index, throwing an exception if the index is out of bounds.
- ⇨ pop_back() Method: This method removes the last element from the array.
- ⇨ print() Method: This method prints all the elements in the array.
- ⇨ Main Function: Demonstrates how to use the DynamicArray class.

- ⇨ This implementation provides a basic dynamic array functionality similar to std::vector<int>, but it is simplified and does not include all the features of the standard library vector.
