C++ REVIEW

STL CONTAINERS
• **Containers** are ADTs to store collection of related data.
• Example: vector, list, map
• Member functions to store, add, delete, update, access and related operations.
• Example: push_back, at

• **Iterators** are objects to iterate over the elements in the containers and access them.
• **Algorithms** are set of well known commonly used algorithms working with iterators.
• Example: sort, swap
BASIC CONTAINERS

• Maintains the order of the inserted elements as specified.

• Examples:
  • **Vector**: Growable array (doubling strategy) with flexible length. Allows random access and contiguous allocation of memory.
  • **List**: Doubly linked list. Bidirectional access with fast insert and delete. No random access.
  • **Deque**: Double ended queue. Fast insert and delete at the ends of the queue. Allows random access and flexible length. No contiguous allocation of memory.
ADAPTIVE CONTAINERS

• Adapted version of other containers with restricted operations for simplicity.

• Examples:
  • **Stack**: Last In First Out (LIFO). push(), pop(), top(), size(), empty()
  • **Queue**: First In First Out (FIFO). push(), pop(), front(), back(), size(), empty()
  • **Priority_queue**: Like queue but element with highest priority is first in the queue. push(), pop(), top(), size(), empty()
ASSOCIATIVE CONTAINERS

• Elements are inserted according to a predefined order.
• **Maps**: Dictionary with (key,value) pair. Key states the order of the sequence.
• **Sets**: Container with sorted elements. Can be viewed as a map where value and key are same element.
DISTINCT VS. NON-DISTINCT

• Non-multi: Cannot handle non-unique keys or elements
  • Map: Keys should be unique
  • Set: Elements should be unique

• Multiple: Can handle multiple keys or elements
  • Multimap: Multiple non-unique keys
  • Multiset: Duplicate elements allowed.

• Eg: \( S = \{1,2,3,4,5\} \) is a set but \( S1 = \{1,2,2,4,5\} \) is a multiset.
ORDERED VS UNORDERED

• **Ordered**: Order preserved in storing
• Faster range iteration than unordered counterparts but slower element access.
  • Map
  • Multimap
  • Set
  • Multiset

• **Unordered**: No order in storing. Uses hashing.
• Faster access of elements. Range iteration is slower than ordered counterpart.
  • Unordered_map
  • Unordered_multimap
  • Unordered_set
  • Unordered_multimap
EXERCISE

• Design a barcode scanner for a super market: Given the barcodes of the items in your shopping list, look up in the inventory database for its availability and if available print the quantity.

• What will be the underlying container?

• Read the inventory data (item barcode, quantity, per item cost) from the file given in the web.

• Bonus: Prepare the receipt (total cost) for the shopping list.