Vector Basics:

1. **Vector Declaration:**
    - vector<int> v1 = {1, 2}; — Vector with values initialized.
    - vector<int> v(5); — Vector of size 5, elements default-initialized (0 for int).
    - vector<int> v3(5, 2); — Vector of size 5, with all elements set to 2.
    - vector<int> v4(v3); — A copy constructor that creates a vector with the same elements as v3.

Iteration Methods:

2. **Basic Iteration:**
    - v.begin() — Points to the first element.
    - v.end() — Points just past the last element.
    - v.rbegin() — Points to the last element (reverse iterator).
    - v.rend() — Points just before the first element (reverse iterator).
3. **Iterator Operations:**
    - **Advance by 1:**
        - it++; — Moves the iterator to the next element.
        - it += 1; — Another way to move the iterator by one position.
    - **Dereferencing iterator:**
        - *(it) — Accesses the value at the iterator's position.

For Each Element in Vector:

4. **Using auto:**
    - for (auto x : v3) { cout << x << " "; } — Iterates through elements directly.
5. **For Loop with Iterator:**
    - Forward direction:
        - for (auto it = v.begin(); it != v.end(); ++it) — Iterates from start to end.
    - Reverse direction:
        - for (auto it = v.rbegin(); it != v.rend(); ++it) — Iterates from end to start.

Output Using Iterators:

5. **Accessing via Iterators:**
    - vector<int>::iterator it = v5.begin(); it++; cout << *(it) << " ";
        - Prints the value at the second position of the vector v5.
    - it += 1; cout << *(it) << " ";
        - Another method of advancing and printing the next value.

Looping Through Vectors:

6. **Forward Iteration (Using Iterators):**
    - for (auto it = v.begin(); it != v.end(); ++it) { std::cout << *it << " "; }
        - Loops through the vector and prints each element from start to end.
7. **Reverse Iteration (Using Reverse Iterators):**
    - for (auto it = v.rbegin(); it != v.rend(); ++it) { std::cout << *it << " "; }
        - Loops through the vector and prints each element from end to start.

Vector Modifications:

6. **Erase Elements:**
    - Erase one element:
        - v.erase(v.begin() + 1); — Removes element at position 1.
    - Erase a range of elements:
        - v.erase(v.begin() + 2, v.begin() + 4); — Removes elements between positions 2 and 4 (excluding 4).

7. **Insert Elements:**

- vector<int> v(2, 100);
    - Creates a vector v with two elements, both set to 100.
- v.insert(v.begin(), 300);
    - Inserts 300 at the beginning of the vector. After this, v will be {300, 100, 100}.
- v.insert(v.begin() + 1, 2, 10);
    - Inserts two 10s at position 1. After this, v will be {300, 10, 10, 100, 100}.
- v.insert(v.begin(), copy.begin(), copy.end());
    - Inserts all elements from another vector copy at the beginning of v.

8. **Pop and Swap:**
    - v.pop_back(); — Removes the last element.
    - v1.swap(v2); — Swaps the contents of two vectors, v1 and v2.
9. **Clear Vector:**
    - v.clear(); — Clears all elements from the vector.
10. **Check if Vector is Empty:**
    - v.empty(); — Returns true if the vector is empty.

Additional Notes:

- **Memory Management:**
    - *(v.begin()) gives the value at the first element in the vector.
    - *(v.end()) is past-the-end and doesn't hold any value.
    - Reverse iteration starts from the last element and moves towards the first.