std::vector

09 June 2021 07:07

1. Dynamic array in contiguous memory – creation

```
EMPTY        - std::vector<int> first;                              // empty vector of ints
#ELEMENTS     - std::vector<int> second (4,100);                    // four ints with value 100
PARTIAL COPY - std::vector<int> third (second.begin(),second.end());// iterating through second
FULL COPY    - std::vector<int> fourth (third);                     // a copy of third
std::vector<int> arr {1, 2, 3};
```

2. Iterators – const, non-const random-access and reverse iterator

3. Members – front, back, begin, end, rbegin, rend, c(r)begin, c(r)end
   [], at

4. Size and capacity
   - Size - number of elements currently used
   - Capacity – max possible element storage
   - Empty?

   MAX_SIZE - ???

5. Capacity changing
   - Reserve – increase capacity .. if lesser capacity than current then no effect
   - shrink_to_fit () - make capacity = size

6. Vector size related
   - resize(#, value) - if #>size insert at end with value value or val init.. if #<size destroy elements at end? making size = n
   - erase(it) or erase(itr1, itr2) - and reduce size ---> capacity may not change
   - clear() - make size 0 ---> capacity may not change
   - Pop_front, pop_back

   - assign(#, value) or assign(itr1, itr2) where itr1, itr2 is from different container
   - Insert(itr, #, value) or insert(itr, itr1, itr2), push_back -> copies or moves existing objects into the container.
   - emplace(itr, constructor args), emplace_back ---> inplace construction of vector elements

7. =, [] operator works
   I
8. F
   I
9. F
10. F
    I
11. F
    I
12. F
    I
13. F
    I
14. F
    I
15. F
    I
16. D
    I

17. F
    I
18. F
    I
19. F
    I
20. F
21. F
    I
22. F
    I
23. F
    I
24. F
    I
25. F
    I
26. F
    I
27. D
    I
28. F
    I
29. F
    I
30. F
    I
31. F
32. F
    I
33. F
    I
34. F
    I
35. F
    I
36. F
    I
37. F
    I
38. D
    I
39. F
    I
40. F
    I
41. F
    I
42. F
43. F
    I
44. F

I

45. F

    I

46. F

    I

47. F

    I