

# std::list

09 June 2021 07:07

## 1. std::list – creation

```
std::list<int> first; // empty list of ints
std::list<int> second (4,100); // four ints with value 100

std::list<int> third (second.begin(),second.end()); // iterating through second
std::list<int> fourth (third); // a copy of third
// the iterator constructor can also be used to construct from arrays
int myints[] = {16,2,77,29};
```

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

## 3. Members – front, back, begin, end, rbegin, rend, c(r)begin, c(r)end,

## 4. Size

- Size - number of elements currently used
- Empty?

MAX\_SIZE - ???

## 5. List 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
- `clear()` - make size 0
- `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_front`, `push_back`,
- `emplace(itr, constructor args)`, `emplace_front`, `emplace_back`,

## 6. Operator =

## 7. Special - merge

```
first.merge(second);
```

## 8. Special – remove any value from list

```
remove (const value_type& val)
```

## 9. Special – sort(<), sort(binary-predicate or bool func)

## 10. Special – splice

- The first version (1) transfers all the elements of x into the container.
- The second version (2) transfers only the element pointed by i from x into the container.
- The third version (3) transfers the range [first,last) from x into the container.

## 11. Special - reverse

## 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