**std::vector<T> (для прикладу – std::vector<int>)**

1) конструктор

explicit vector(size\_type Count);  
vector(size\_type Count, const Type& Val);

2) ініціалізація вектора

2.a) push\_back  
2.b) resize(), size() + at(), operator[]  
2.c) resize() + vector<T>::iterator, begin(), end()

=> проініціалізувати вектор n-елементів в циклі (заповнити довільними значеннями)  
=> прочитати вектор n-елементів з текстового файлу

3) друк вектора

3.a) size() + at(), operator[]  
3.b) vector<T>::const\_iterator, cbegin(), cend()

4) зворотній порядок

4.a) size() + at(), operator[]  
4.b) vector<T>::reverse\_iterator, vector<T>::const\_reverse\_iterator, rbegin(), rend(), crbegin(), crend()

=> створити копію вектора у зворотньому порядку  
=> надрукувати вектор у зворотньому порядку

5) вставка елементів у вектор

iterator insert(const\_iterator \_Where, const Type& val);  
void insert(const\_iterator \_Where, size\_type count, const Type& val);  
void insert(const\_iterator \_Where, InputIterator first, InputIterator last);

v1.insert( v1.begin( ) + 1, 40 );  
v1.insert( v1.begin( ) + 2, 4, 50 );  
v1.insert( v1.begin( )+1, v1.begin( )+2, v1.begin( )+4 );

=> insert in a certain position a certain value

=> insert in a certain position in one vector a certain range from the other vector

6) work with user-defined class

std::vector<Student>

std::vector<Student\*>

=> read from file

=> print

=> find a Student with maximum mark: function that returns iterator to the element, print this Student

=> copy to a second vector all Students that belong to a certain group, print this vector